(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Ex Parte Reexamination of:
Li GONG

Control No.: 90/011,491 Confirmation No.: 8208

Reexamination Filing Date: February 15, 2011 Art Unit: 3992

Patent No.: 6,125,447 Examiner: M. Steelman

Issue Date: September 26, 2000

For: PROTECTION DOMAINS TO PROVIDE SECURITY IN A COMPUTER SYSTEM

RESPONSE TO FIRST OFFICE ACTION

MS Ex Parte Reexam
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Patent Owner provides the following remarks in response to the first Office Action issued on June 29, 2011 ("the Action"), in this reexamination. Patent Owner requests reconsideration.

Amendments to the Claims begins on page 2.
Status of the Claims begins on page 8.
Listing of the Exhibits begins on page 9.
Remarks/Arguments begin on page 10.
Interview Summary begins at page 15.

AMENDMENTS TO THE CLAIMS

Claim 1 (issued): A method for providing security, the method comprising the steps of:

establishing one or more protection domains, wherein a protection domain is associated with zero or more permissions;

establishing an association between said one or more protection domains and one or more classes of one or more objects; and

determining whether an action requested by a particular object is permitted based on said association between said one or more protection domains and said one or more classes.

Claim 2 (issued): The method of claim 1, wherein:

at least one protection domain of said one or more protection domains is associated with a code identifier;

at least one class of said one or more classes is associated with said code identifier; and

the step of establishing an association between said one or more protection domains and said one or more classes of one or more objects further includes the step of associating said one or more protection domains and said one or more classes based on said code identifier.

Claim 3 (issued): The method of claim 2, wherein said code identifier indicates a source of code used to define each class of said one or more classes.

Claim 4 (issued): The method of claim 2, wherein said code identifier indicates a key associated with each class of said one or more classes.

Claim 5 (issued): The method of claim 2, wherein said code identifier indicates a source of code used to define each class of said one or more classes and indicates a key associated with each class of said one or more classes.

Claim 6 (issued): The method of claim 2, wherein the step of associating said one or more protection domains and said one or more classes based on said code identifier further includes associating said one or more protection domains and said one or more classes based on data persistently stored, wherein said data associates code identifiers with a set of one or more permissions.

Claim 7 (currently amended): A method of providing security, the method comprising the steps of:

establishing one or more protection domains, wherein a protection domain is associated with zero or more permissions;

establishing an association between said one or more protection domains and one or more sources of code, wherein the one or more sources of code is at least one of a file, a persistent object, a FLASH_EPROM reader, or a set of system libraries; and

in response to executing code making a request to perform an action, determining whether said request is permitted based on a source of said code making said request and said association between said one or more protection domains and said one or more sources of code.

Claim 8 (issued): The method of claim 7, wherein the step of establishing an association between said one or more protection domains and said one or more sources of code further includes establishing an association between said one or more protection domains and said one or more sources of code and one or more keys associated with said one or more sources of code.

Claim 9 (issued): The method of claim 8, wherein the step of establishing an association between said one or more protection domains and said one or more sources of code and said one or more keys associated with said one or more sources of code further includes establishing said association between said one or more protection domains and said one or more sources of code and said one or more keys associated with said one or more sources of code based

on data persistently stored, wherein said data associates particular sources of code and particular keys with a set of one or more permissions.

Claim 10 (issued): A computer-readable medium carrying one or more sequences of one or more instructions, the one or more sequences of the one or more instructions including instructions which, when executed by one or more processors, causes the one or more processors to perform the steps of:

establishing one or more protection domains, wherein a protection domain is associated with zero or more permissions;

establishing an association between said one or more protection domains and one or more classes of one or more objects; and

determining whether an action requested by a particular object is permitted based on said association between said one or more protection domains and said one or more classes.

Claim 11 (issued): The computer readable medium of claim 10, wherein:

at least one protection domain of said one or more protection domains is associated with a code identifier;

at least one class of said one or more classes is associated with said code identifier; and

the step of establishing an association between said one or more protection domains and said one or more classes of one or more objects further includes the step of associating said one or more protection domains and said one or more classes based on said code identifier.

Claim 12 (issued): The computer readable medium of claim 11, wherein said code identifier indicates a source of code used to define each class of said one or more classes.

Claim 13 (issued): The computer readable medium of claim 11, wherein said code identifier indicates a key associated with each class of said one or more classes.

Claim 14 (issued): The computer readable medium of claim 11, wherein said code identifier indicates a source of code used to define each class of said one or more classes and indicates a key associated with each class of said one or more classes.

Claim 15 (issued): The computer readable medium of claim 14, wherein the step of associating said one or more protection domains and said one or more classes based on said code identifier further includes associating said one or more protection domains and said one or more classes based on data persistently stored, wherein said data associates code identifiers with a set of one or more permissions.

Claim 16 (currently amended): A computer-readable medium carrying one or more sequences of one or more instructions, wherein the execution of the one or more sequences of the one or more instructions causes the one or more processors to perform the steps of:

establishing one or more protection domains, wherein a protection domain is associated with zero or more permissions;

establishing an association between said one or more protection domains and one or more sources of code, wherein the one or more sources of code is at least one of a file, a persistent object, a FLASH_EPROM reader, or a set of system libraries; and

in response to executing code making a request to perform an action, determining whether said request is permitted based on a source of said code making said request and said association between said one or more protection domains and said one or more sources of code.

Claim 17 (issued): The computer readable medium of claim 16, wherein the step of establishing an association between said one or more protection domains and said one or more sources of code further includes establishing an association between said one or more protection domains and said one or more sources of code and one or more keys associated with said one or more sources of code.

Claim 18 (issued): The computer readable medium of claim 17, wherein the step of establishing an association between said one or more protection domains and said one or more sources of code and said one or more keys associated with said one or more sources of code further includes establishing said association between said one or more protection domains and said one or more sources of code and said one or more keys associated with said one or more sources of code based on data persistently stored, wherein said data associates particular sources of code and particular keys with a set of one or more permissions.

Claim 19 (issued): A computer system comprising:

a processor;

a memory coupled to said processor;

one or more protection domains stored as objects in said memory, wherein each protection domain is associated with zero or more permissions;

a domain mapping object stored in said memory, said domain mapping object establishing an association between said one or more protection domains and one or more classes of one or more objects; and

said processor being configured to determine whether an action requested by a particular object is permitted based on said association between said one or more protection domains and said one or more classes.

Claim 20 (issued): The computer system of claim 19, wherein:

at least one protection domain of said one or more protection domains is associated with a code identifier;

at least one class of said one or more classes is associated with said code identifier; and

said computer system further comprises said processor configured to establish an association between said one or more protection domains and said one or more classes of one or more objects by associating said one or more protection domains and said one or more classes based on said code identifier.

Claim 21 (issued): The computer system of claim 20, wherein said code identifier indicates a source of code used to define each class of said one or more classes.

Claim 22 (issued): The computer system of claim 20, wherein said code identifier indicates a key associated with each class of said one or more classes.

Claim 23 (issued): The computer system of claim 20, wherein said code identifier indicates a source of code used to define each class of said one or more classes and indicates a key associated with each class of said one or more classes.

Claim 24 (issued): The computer system of claim 20, further comprising said processor configured to associate said one or more protection domains and said one or more classes based on said code identifier by associating said one or more protection domains and said one or more classes based on data persistently stored in said computer system, wherein said data associates code identifiers with a set of one or more permissions.

STATUS OF THE CLAIMS

Pursuant to 37 C.F.R. § 1.530(e), and with this Amendment, claims 1-24 are pending. Claims 7 and 16 have been amended. Support for this amendment is found, for example, at '447, 3:15-21.

LISTING OF EXHIBITS

The following Exhibits are submitted herewith.

Exhibit	Description
A	Declaration of Prof. Benjamin Goldberg ("Goldberg Declaration")
B	Curriculum Vitae of Prof. Benjamin Goldberg

REMARKS

I. Introduction

Claims 1-24 are pending in the present reexamination of U.S. Patent No. 6,125,447 ("the '447 Patent"). The '447 Patent is directed to a system, apparatus, and method for providing security. This is achieved through the following unique combination of features as set forth in exemplary claim 1:

- "**establishing one or more protection domains, wherein a protection domain is associated with zero or more permissions;**[1]
- **establishing an association between said one or more protection domains and one or more classes of one or more objects; and**
- **determining whether an action requested by a particular object is permitted based on said association between said one or more protection domains and said one or more classes.**"

The Office, during the original examination, found this combination of features—specifically the association between the classes and the protection domain—to be patentable over the prior art. (Interview Summary, dated May 5, 2000.)

The art of the present reexamination does not compel a different conclusion. None of the cited references—Fischer, Goldstein, or Shah—teaches or suggests the above claimed combination of features of the '447 Patent. Fischer does not disclose "**establishing an association between … protection domains and … classes**." Instead, Fischer is directed to providing security by associating program authorization information ("PAI") with a *program*. Fischer is silent as to *classes* or any security based on *classes*. Indeed, the lack of any discussion as to classes in Fischer is dispositive not only with respect to the above "establishing" feature, but also to the other features such as "**determining whether an action requested by a particular object is permitted based on**

[1] Throughout this Response, recited claim language is in quotes and boldface.

said association between ... protection domains and ... classes."

Goldstein and Shah have similar deficiencies. Goldstein and Shah discuss the Gateway Security Model for the Java Electronic Commerce Framework ("JECF"). Through the use of "roles," "tickets," "gates," and "capabilities," the JECF provides security to allow a "cassette" to access a resource. However, Goldstein and Shah do not disclose "**establishing an association between ... protection domains and ... classes**" because both Goldstein and Shah, similarly to Fischer, are silent as to any security based on *classes*.

Accordingly, Patent Owner submits that all of the rejections should be withdrawn.

II. Brief Summary of the Invention

The '447 Patent is directed to a new approach to designing security policies within a computer system. The focus of the '447 Patent is to maintain and enforce security rules in an object-oriented system using protection domains. Protection domains, which are associated with zero or more permissions, are further associated with one or more classes of one or more objects, as set forth in exemplary claim 1. This association between protection domains and classes may then be used to determine whether an action requested by a particular object in an object-oriented programming environment, instantiated from a particular class, is permitted. (Goldberg Declaration, ¶8.) Claim 1, as noted above, provides for "**determining whether an action requested by a particular object is permitted based on said association between said one or more protection domains and said one or more classes**."

The claimed invention provides various advantages over the prior art. For example, associating protection domains with classes makes it possible to establish and modify permissions in an object-oriented programming language easily. Further, each class may be associated with a different protection domain, thereby providing finer granularity compared to the prior art where all code (e.g., an entire program) is restricted to the same limited set of resources. Additionally, by grouping permissions within protection domains and assigning those protection domains to classes,

different security constraints for different portions of code within the same program can be succinctly expressed. (Goldberg Declaration, ¶9.)

An exemplary method and exemplary associations of the claimed invention are described at least at column 11, line 33 – column 13, line 22 and illustrated in FIGS. 5 and 6 (reproduced below).

Figure 6 depicts a call stack representing objects associated with protection domains and permissions. "Call stack 610 represents the calling hierarchy of the methods invoked by the thread but have not yet been completed by the thread." (11:42-44.) In this example, a.x invokes b.y, which invokes c.z. Each of the class objects is associated with a protection domain. Each of these protection domains is further associated with permissions. Protection Domain I includes the permissions to write and read "/tmp/". Protection Domain J includes the permissions to write and read "/share/".

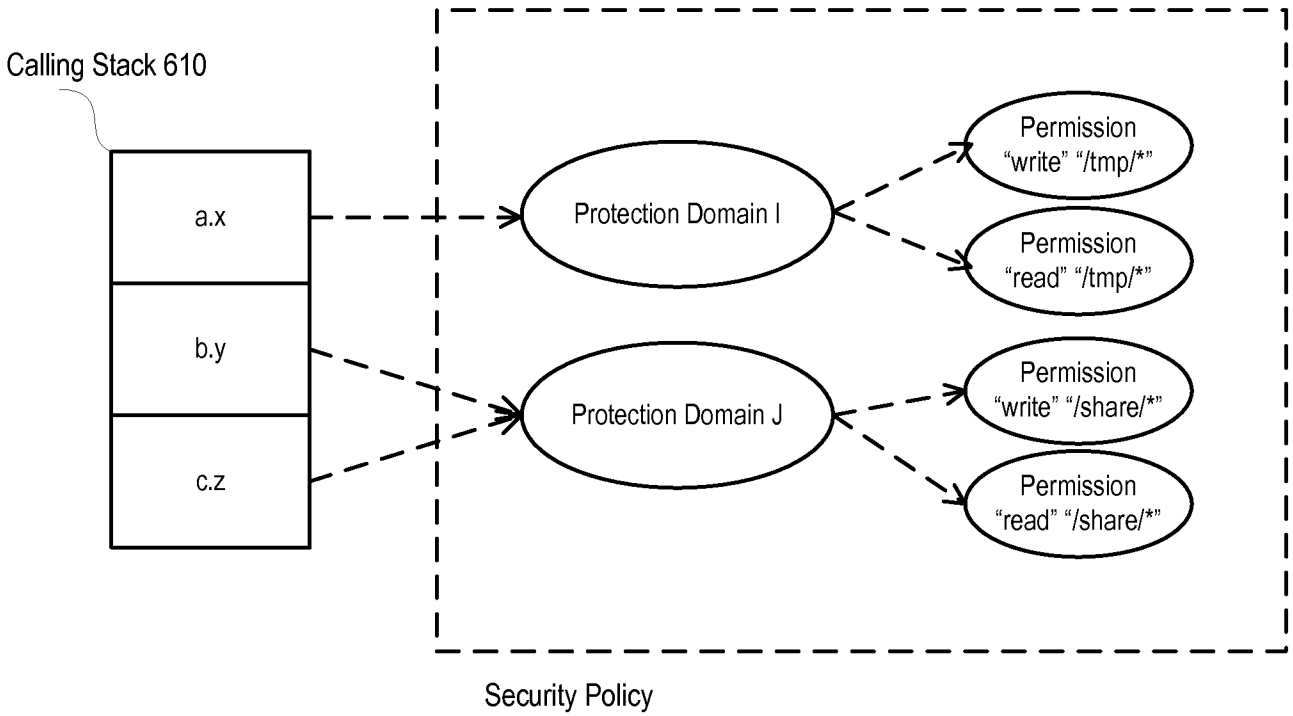

Fig. 6

In Figure 5, the exemplary method for providing security begins at 550 with a request to perform a particular action being received from an object. For example, object "a" may request to read a file in the /tmp/ directory, such as illustrated in FIG. 6. At 564, "a determination is made as to whether the required permission for the requested action is included in the protection domains associated with the request to perform the action." (12:25-28.) This is accomplished by "[e]xamining the permission of a particular protection domain associated with [the requesting] object . . . by determining an object's class." (12:40-42.) Therefore, the class of object "a" will be determined as, for example, class "CA." Next, the protection domain associated with the class is invoked. In the example above, class "CA" is associated with Protection Domain I as illustrated in FIG. 6. Next, "[e]ach permission in the [class's] protection domain is examined until it is determined whether any permission in the protection domain authorizes the requested permission." (12:50-54.) If, at 564, the permission is included in the class's protection domain, the object is allowed to perform the requested action at 568. If the permission is not included, execution ceases. In the example above, Protection Domain I includes permission "read /tmp/" as shown in FIG. 6. Therefore, object "a" will be allowed to read a file in the /tmp/ directory.

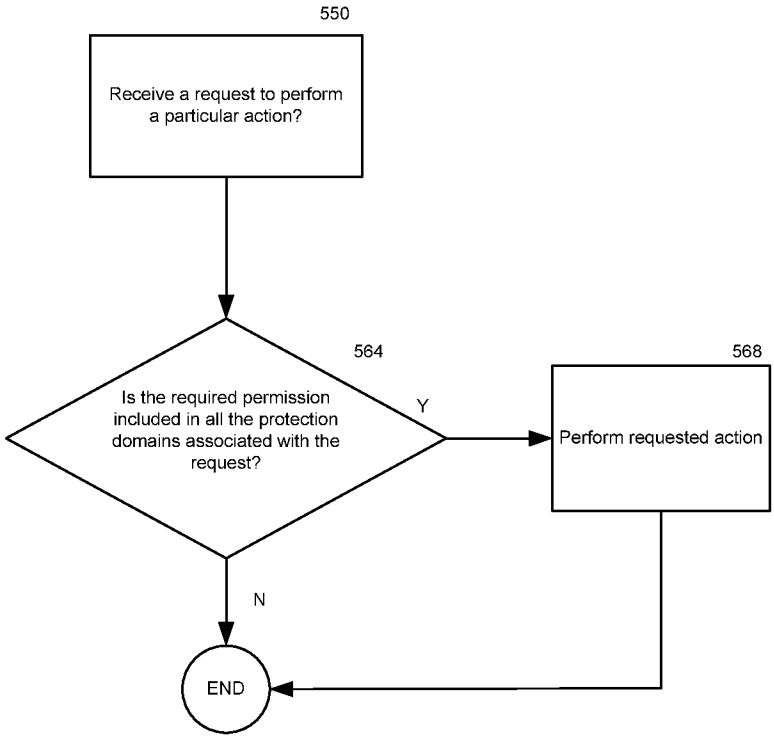

Fig. 5

During the original examination, the Examiner found the above quoted combination of features of exemplary claim 1 to be patentable over the art of record. In particular, the Examiner stated that it "was determined that the [cited] reference … does not teach an association between the classes and the protection domain." Further, "the [cited] reference … does not teach 'sources of code' associated with protection domains." (Interview Summary dated May 5, 2000.)

As discussed in detail below, the cited references in the pending reexamination also fail to teach or suggest the novel combination of features.

III. Interview Summary Pursuant to 37 C.F.R. 1.560(b)

Patent Owner thanks the Examiner for the courtesy of an in-person interview to discuss the Office Action on August 4, 2011. In attendance for the interview were Examiners Mary Steelman, Eric Kiss, and Fred Ferris, and for the Patent Owners, Christopher Eide (48,375), Julie Akhter (59,570), George Simon (47,089), Benjamin Goldberg (technical expert), Tracy Druce (35,493), and Lissi Mojica (63,421). Claim 1 of the '447 Patent and the Fischer, Goldstein, and Shah references were discussed during the interview.

The following features of claim 1 were primarily discussed: "**establishing an association between said one or more protection domains and one or more classes of one or more objects**," and "**determining whether an action requested by a particular object is permitted based on said association between said one or more protection domains and said one or more classes**."

Regarding Fischer, Patent Owner presented that Fischer discloses an association between the program authorization information ("PAI") and a program, but not to "one or more classes of one or more objects." Patent Owner further presented that Fischer does not determine "whether an action requested by a particular object is permitted based on said association between ... protection domains and ... classes." Rather, Fischer discloses determining whether an action requested by a program is permitted based on an association between the program and the PAI. (See also, the Examiner Interview Agenda accompanying the Examiner's Interview summary mailed August 12, 2011.)

Regarding Goldstein and Shah, Patent Owner presented that Goldstein/Shah do not teach that "roles" are associated with the class of the "cassette." (See also, the Examiner Interview Agenda accompanying the Examiner's Interview summary mailed August 12, 2011.)

In response to Patent Owner's presentation, the Examiners inquired as to the proper scope of the claims, particularly with respect to the term "association" and requested that Patent Owner address the construction of "association" in its response. The proper scope of the claims is discussed in detail below and the claims are distinguishable over the cited art as discussed in more detail below.

IV. The Anticipation Rejection Based on Fischer Should Be Withdrawn (Claims 1-24)

A. Fischer Does Not Disclose the Claimed Association between Protection Domains and Classes

Claim 1[2] recites, *inter alia*:

> "**establishing an association between said one or more protection domains and one or more classes of one or more objects**."

The Office Action (p. 12) asserts that Fischer discloses this claim feature in a "set of authorities and/or restrictions assigned to a program to be executed are referred to herein as 'program authorization information' (or 'PAI')." (Fischer, 2:24-26.) Patent Owner respectfully disagrees with this assertion.

Fischer teaches a security monitor that limits the ability of a program about to be executed to the use of predefined resources through the use of the PAI. Fischer envisions the program, to which the PAI is assigned, as a virus program (Fischer, 1:30-31), a game (Fischer, 9:29-30), or the like. A program, however, is most certainly *not* a class. According to Prof. Goldberg's declaration, "a program in an object-oriented language (e.g. Java and C++) may *include* classes, whereas a program in a non object-oriented language (e.g. C, Lisp, assembly language, etc.) would not be considered to include classes." (Emphasis added; Goldberg Declaration, ¶11.) Thus, Fischer's programs do not disclose "**one or more classes of one or more objects**" as expressly required by the claims.

Further, Fischer's provision of security at a program-level is also insufficient. "In Fischer, every portion of the program – including the classes if the program is written in an object-oriented language – would have the same permissions. Thus, even in an object-oriented language, Fischer does not provide a mechanism for associating different permissions with different portions of code within the same program." (Goldberg Declaration, ¶11.) Because Fischer does not teach anything related to *classes*, it cannot disclose that the PAI is associated with "**one or more classes of one or more objects**." Therefore, claim 1 is not anticipated by Fischer.

In the Office Action and during the interview, the Examiner viewed the program in Fischer as having classes and inquired that, if Fischer disclosed a program with classes, whether the association of the PAI with a program having classes would meet the above feature if construed broadly. It is believed that the Examiner was trying to abstract an object-oriented implementation from Fischer based on Fischer's Fig. 3C. Even if Fischer disclosed a program with classes, which it does not (Goldberg Declaration, ¶11), the feature would still not be met, as the Patent Office's proposed construction is not supported by the claim language nor is it consistent with the '447 specification, as noted by Prof. Goldberg during the interview. The claim language—"**establishing an association between … protection domains and … classes**"—expressly links the protection domain to classes, *not* programs. The link between protection domains and classes was emphasized in the '447 specification as providing a finer level of granularity for establishing protections within a computing system than at the entire program level (i.e., at the level that Fischer discusses). (Goldberg Declaration, ¶12.) For example, the specification states that the "sandbox approach is not very granular because all remote code is restricted to the same limited set of resources." ('447, 2:11-13.)

It is undesirable to choose a coarser level of granularity, such as "all remote code" or an entire program such as the one described in Fischer, because this limits the flexibility of the code to be assigned varying permissions. According to Prof. Goldberg's declaration, "[c]hoosing a coarser level of granularity than associating permissions with classes, such as associating the same permissions with 'all remote code' (i.e. all code loaded from remote sources) as in the prior art Java 'sandbox' model, or associating permissions with an entire program as in Fischer, limits the flexibility of the code to be assigned varying permissions. For example, since Java programs are often constructed from classes retrieved from multiple sources – some of the sources more trusted than others – it is desirable to be able to assign different permissions to the classes retrieved from different sources." (Goldberg Declaration, ¶12.) All remote code, or all code within a program, is assigned the same permissions, regardless of trustworthiness of the various components. However, "[p]roviding security measures that allow more granularity than the sand box method involves

[2] Throughout this Response, Patent Owner refers to exemplary independent claim 1. The analysis as to claim 1 applies

establishing a complex set of relationships between principals and permissions." ('447, 2:24-26.) "In Fischer, however, all code within a program is assigned the same permissions, regardless of trustworthiness of the various components. The claimed invention of the '447 Patent solves these problems by providing a more granular level of protection (i.e., associating protection with classes) and simplifying the relationships between principals and permissions via the protection domain mechanism. In fact, Fischer teaches away from using class-level protection by teaching program-level protection." (Goldberg Declaration, ¶13.) "'Teaching away' does not require that the prior art foresaw the specific invention that was later made, and warned against taking that path." *Spectralytics, Inc., v. Cordis Corp.* (Fed. Cir. June 13, 2011). Rather, the design of the prior art device itself can teach away from the invention. (*Id.*) Here, the design of Fischer (i.e., protection associated at the program level) teaches away from the claimed "**establishing an association between … protection domains and … classes**" of the '447 Patent.

Moreover, upon reading the specification, one of skill in the art would recognize that the "**association**" in "**establishing an association between … protection domains and … classes**" requires the protection domains to be associated with classes, not with programs. (Goldberg Declaration, ¶14.) See, e.g., '447, 2:10-22 (where in the background art, sets of files are "associated" with particular banks), and contrast with examples described at '447, 2:50-3:50 (where protection domains are "associated" classes and permissions), 6:65-66 (where methods are "associated" with objects), etc. In each of the examples from the '447 specification, "associated" means that the two pieces are associated with each other, not associated through another construct, such as a program. One of skill in the art could not interpret the claimed "**association**" differently because none of the examples provided in the specification suggest otherwise. (Goldberg Declaration, ¶14.) Indeed, to adopt such a construction would directly contravene the requirement of giving claims their "broadest reasonable construction *consistent with the specification*" in a reexamination. (MPEP 2258(I)(G) (emphasis added); *In re Suitco Surface*, 603 F.3d 1255, 1259 (Fed. Cir. 2010). In contrast, construing the feature as associating protection domains with classes would be consistent with the specification, as the entire patent teaches such an association.

---

to the other independent claims (claims 10 and 19) unless otherwise noted.

(Goldberg Declaration, ¶14.) According to Prof. Goldberg's declaration, "[o]ne of ordinary skill in the art reading the '447 patent would understand the "association" between classes and protection domains to be any mechanism that allows the code from different classes within the same program to operate with different protection domains. Fischer's disclosure of associating a PAI with a program does not provide such a mechanism." (Goldberg Declaration, ¶14.) Therefore, one of skill in the art would recognize that associating a PAI with a program does not disclose "**establishing an association between … protection domains and … classes**," as recited in claims 1, 10, and 19.

When discussing Fischer, the Office Action cites to the disclosure at ('447, 7:4-6), that each object belonging to a class has the same fields and the same methods, to conclude that protection domain attributes may be stored in the fields of an object instantiated from that class. (Office Action, page 12.) However, at the time of invention, while programs as a whole may generally have had protection attributes as shown by Fischer, protection was not provided on a class by class basis. (Goldberg Declaration, ¶14.) Accordingly, although the fields of objects instantiated from the same class were the same, the fields of a *class* did not have protection domain attributes.

For at least this reason and the reasons discussed above, Fischer does not teach or suggest "**establishing an association between one or more protection domains and one or more classes of one or more objects**," as claimed in independent claims 1, 10, and 19.

B. Fischer Does Not Disclose the Claimed Determination of Whether an Action Requested by a Particular Object is Permitted based on said Association

Claim 1 recites, *inter alia*:

> "**determining whether an action requested by a particular object is permitted based on said association between said one or more protection domains and said one or more classes**."

As claimed, "**an action [is] requested by a particular object**." Whether this action is permitted is "**based on said association between said one or more protection domains and said one or more classes**." Unlike the claimed invention, in Fischer, the program requests the action. (Fischer, FIG. 10.) As discussed above, the PAI is associated with the program. (Fischer, 2:24-26.) Therefore, Fischer determines whether an action requested by a *program* is permitted based on an

association between the PAI and the *program*. (Goldberg Declaration, ¶15.) This disclosure in Fischer is deficient as to the above feature in independent claims 1, 10, and 19 for a number of reasons.

First, the feature in claims 1, 10, and 19 requires an association between protection domains and classes. As discussed in Section A above, such an association is not disclosed in Fischer. Fischer determines whether an action requested is permitted based on an association between the PAI and the program, not whether an action requested by a particular object is permitted "**based on an association between … protection domains and … classes**."

Second, the claim language requires that the action be requested by a particular "**object**" based on an association between protection domains and "**classes**." That is, "**object**" and "**classes**" are distinct entities. According to Prof. Goldberg's declaration, "[a] class is a portion of code defining a data type and specifies the data fields (aka 'instance variables' or 'data members') and procedures (aka 'methods' or 'member functions') that are encapsulated together within elements of that type. An object results from 'instantiating' a class – that is, an object is a data structure of the type defined by the class and contains a value for each data field specified by the class definition and utilizing the procedures defined by the class definition. Fundamentally, a class is code and an object is data." (Goldberg Declaration, ¶16.) This construction is supported by the specification. As described therein, "[w]hen an object requests an action, a determination is made as to whether the action is permitted based on the class to which the object belongs and the association between classes and protection domains." ('447, Abstract.) Therefore, Fischer's program cannot be interpreted as both an "**object**" and the "**class**" of the particular object associated with the protection domain.

Third, the Office Action (p. 13) cites to checking the signatures of the PAIs in Fischer to determine "**whether an action … is permitted**." In Fischer, the PAIs are signed and the system checks that the signatures are valid, authorized, and trusted before proceeding. (Fischer, FIG. 10, 16:49-17:3.) However, in the '447 Patent, determining whether an action is permitted is "**based on said association between said one or more protection domains and said one or more classes**."

Checking (digital) signatures in Fischer is not the same as "**determining whether an action … is permitted based on an association between … protection domains and … classes**," as recited in the '447 claims. According to Prof. Goldberg's declaration, "a digital signature, like a handwritten signature, is a mechanism for verifying the identity of the author of something – such as document, a message or, in Fischer's case, a program and its PAI. Digital signatures generally rely on cryptographic methods, such as providing a public decryption key to decrypt a digital string (the signature) that could have been only encrypted by someone, such as the author to be verified, holding a private encryption key. In Fischer, although a signature attached to a program or PAI is used to authenticate the program or PAI, the signature does not specify an association between a class and permissions nor is verifying (checking) a signature the same as determining whether an action is permitted or not." (Goldberg Declaration, ¶17.)

Fourth, the Office Action states that "the protection domain objects are derived from (association) a protection domain class definition." (Office Action, page 13.) However, the recited "**said one or more classes**" do not refer to the classes of the protection domains. The "**said one or more classes**" in the third feature of claim 1 refers to the "**one or more classes of one or more objects**" introduced in the second feature of claim 1. Furthermore, the specification recites that "[w]hen an object requests an action, a determination is made as to whether the action is permitted based on the class to which the object belongs and the association between classes and protection domains." ('447, Abstract.) One of skill in the art, reading the '447 specification, would recognize that the "**said one or more classes**" recited in the claims cannot be the class of the protection domain object. (Goldberg Declaration, ¶18.) "Rather, the '**said one or more classes**' must be the classes from which the '**particular object[s]**' of the claims (i.e. the objects making the requests for actions) are instantiated. These protection domains accordingly cannot be the claimed objects or classes and so the '**association between said one or more protection domains and said one or more classes**' cannot be the association of a particular protection domain object with the protection domain class from which it was instantiated. Rather, it must be the association between the protection domain and the class to which the requesting object belongs." (Goldberg Declaration, ¶18; Emphasis added.)

For at least the reasons above, Fischer does not teach "**determining whether an action requested by a particular object is permitted based on said association between said one or more protection domains and said one or more classes**," as recited in independent claims 1, 10, and 19.

C. Fischer Does Not Disclose the Claimed Association between Classes and Code Identifiers As Recited in Dependent Claims 2, 11, and 20

Claims 2, 11, and 20 recite, *inter alia*:

> "**at least one class of said one or more classes is associated with said code identifier**."

The Office Action asserts that Fischer discloses this feature because "[t]he PAI data structure may contain a manufacturer's signature (i.e., code identifier) and that the PAI with the code identifier is associated with the object/class data structure because it is expressly included as part of the object/class data structure." (Office Action at 14-15.) On the contrary, Fischer does not teach or suggest "**at least one class of said one or more classes [being] associated with said code identifier**."

As discussed above, since Fischer does not teach anything related to classes, but rather refers generally to programs, it cannot disclose that a "**class … is associated with said code identifier**." Further, as disclosed in Fischer, an association of a signature to a PAI and then an association of that PAI to a program is not the same as "**at least one class … is associated with said code identifier**," as claimed in claims 2, 11, and 20. In Fischer, the signature is associated with the PAI, which is associated with the *program*. (Goldberg Declaration, ¶19.) As discussed by Prof. Goldberg, "[o]ne of skill in the art would recognize that the claims require the code identifier to be associated with the class, not with the entire program – thus allowing different code identifiers to be associated with different classes within the same program. Fischer's association of signatures with PAI's and association of PAI's with programs at most associates signatures indirectly with entire programs, not with classes." (Goldberg Declaration, ¶19.)

For at least the reasons above, Fischer does not teach or suggest "**at least one class of said one or more classes is associated with said code identifie**r," as recited in claims 2, 11, and 20

D. The Fischer Rejection With Respect To Claims 7 and 16 Should Be Withdrawn

Claim 7[3] recites, *inter alia*:

> "**establishing an association between said one or more protection domains and one or more sources of code, wherein the one or more sources of code is at least one of a file, a persistent object, a FLASH_EPROM reader, or a set of system libraries**."

Claims 7 and 16 refer to an association between protection domains and "**sources of code**," rather than classes, as in claim 1. Claims 7 and 16 have been amended to clarify the definition of "**sources of code**" as described in the specification: "[e]xamples of sources of code include a file or persistent object stored on a data server connected over a network, a FLASH_EPROM reader that reads instructions stored on a FLASH_EPROM, or a set of system libraries." ('447, 3:16-21.) The Office Action states that claim 7 is rejected for the same reasons as claim 1. (Office Action at 11.) However, the rejection against claim 7 fails for the same reasons as the rejection against claim 1, discussed above.

The Office Action, via the Exhibit 8 claim chart, appears to identify Fischer's "signer of a digital certificate" as the claimed "**source of code**." (Exhibit 8 at 23.) As stated by Prof. Goldberg, "[t]he listed elements of claims 7 and 16 are not met by a signer of a digital certificate. According to Fischer, the signer of a digital certificate could be a user that is authorized to determine the level of trust of a program (Fischer at 11:14-31) or a manufacturer supplying a program (Fischer at 16:12-25). In my opinion, a skilled artisan would not consider a file, a persistent object, a FLASH_EPROM reader, or a set of system libraries to even be similar to a signer of a digital certificate." (Goldberg Declaration, ¶20.)

For at least this reason and the reasons discussed above, Fischer does not teach or suggest "**establishing an association between one or more protection domains and one or more sources**

of code, wherein the one or more sources of code is at least one of a file, a persistent object, a FLASH_EPROM reader, or a set of system libraries," as recited in claims 7 and 16.

E. The Fischer Rejection With Respect to the Dependent Claims Should Be Withdrawn (Claims 2-6, 11-14, 20-24)

The dependent claims are distinguishable over Fischer for at least the same reasons as their respective independent claims. For example, claims 2-6, 11-14, and 20-24 each recite features related to "**classes**." As discussed above, Fischer teaches programs, but not classes. Therefore, it cannot disclose any of these features related to "**classes**" and thus the anticipation rejection should be withdrawn.

Claims 4, 13, and 22, directed to "**a key associated with each class**" are distinguishable for at least the reasons discussed above regarding claims 2, 11, and 20 because the Office Action identifies that the key is part of the digital certificate. (Office Action at 16.)

Claims 6, 15, and 24, directed to "**the step of associating said one or more protection domains and said one or more classes,**" are distinguishable over Fischer at least by virtue of their dependence from their respective independent claims. Claims 8-9 and 17-18 are similarly distinguishable at least by virtue of their dependence from their respective independent claims.

V. The Anticipation Rejection Based on Goldstein with Shah Should Be Withdrawn (Claims 1-24)

The combination of Goldstein with supporting evidence by Shah does not anticipate the claimed invention for the reasons described below. The combination also does not disclose the dependent claims.

A. Goldstein Does Not Disclose the Claimed Protection Domains or the Association between Protection Domains and Classes

Claim 1 recites, *inter alia*:

"**establishing an association between said one or more protection**

[3] Throughout this Response, Patent Owner refers to exemplary independent claim 7. The analysis as to claim 7 applies to claim 16 unless otherwise noted.

domains and one or more classes of one or more objects."

The Office Action states that "[a]ccording to Goldstein, an application called a 'cassette' is associated with a set of permissions (protection domains) represented by 'Roles.'" (Office Action at 20.) On the contrary, as described in greater detail below, Goldstein teaches that the decision to grant a permission may take into account the role that a ticket was instantiated with, but Goldstein does not disclose "**establishing an association between said one or more protection domains and one or more classes of one or more objects**."

Goldstein discloses applications called "cassettes" which are downloadable applets, retained on the customer's system, that store information in a database provided by the JECF. (Goldstein at 8.) The JECF uses a capabilities model to determine "where rights can be transferred from one principal to another." (Goldstein at 10.) "The Capabilities model means that possession of an object confers the right to use it. Capabilities need a gatekeeper service to decide whether to grant a right to the object." (Goldstein at 10-11.) JECF refers to individual capabilities as "permits." (Goldstein at 11.) In an example provided by Goldstein, the ability to read or write to a table (i.e., the permission to read or write to a table) is encapsulated via "permits." (Goldstein at 11.) A "Gate decides whether to grant an instance of a permit to a caller." (Goldstein at 12.) These "gates" are essentially authentication methods. (Goldstein at 15.) Goldstein also introduces "tickets" (a token that represents the actual consumer of the permit) and "roles" (a representation of the business relationship for an object). (Goldstein at 12-13.) "A Ticket is instantiated with a specific role" and is passed into Gates. (Goldstein at 13-15.) "The JECF uses digital signatures to represent roles" which are "essentially a public and private key pair." (Goldstein at 13.) The public key of a role may be embedded in a cassette. (Goldstein at 13.)

Prof. Goldberg discusses that "[i]n JECF, when a principal (e.g. a program) makes a request to access a resource (such as confidential user information in a database), the requester instantiates a ticket with a specific role object. This ticket is passed into a gate, which decides whether to grant a permit (i.e., a capability), taking into account the role object. If the permit is granted, the requester receives permission to access the requested resource." (Goldberg Declaration, ¶22.) However, Goldstein does not teach that the roles are associated with the *class* of the object making a request.

Goldstein instead teaches that the decision to grant a permission may take into account the role that a ticket was instantiated with. Goldstein simply does not teach security based on classes. (Goldberg Declaration, ¶24.)

The Office Action cites to Goldstein's roles as identifying the claimed "**protection domains**." However, "[t]he role object, according to Goldstein, is essentially a public/private key pair (Goldstein at 13) related to the identity of a requester of a resource or to the function (*role*) that the requester is performing. As an analogy in the real world, one role could be 'bank president,' another role could be 'bank teller.' Neither role can be viewed as a 'set of permissions,' but if a request is made to transfer large sums of money, then whether the request comes from a bank president or a bank teller will certainly be considered before the transfer is made. Similarly, a role object is not a protection domain of the '447 patent which can 'viewed as a set of permissions' ('447 patent at 8:42)." (Goldberg Declaration, ¶23.) Since Goldstein's roles cannot be viewed as a set of permissions, they cannot anticipate the claimed "protection domains" of independent claims 1, 7, 10, 16, and 19.

Goldstein also discusses that packages may be security principals; that is, packages may use rights. (Goldstein at 10.) However, packages are not *classes*. (Goldberg Declaration, ¶25.) According to Prof. Goldberg's declaration, "packages are collections *of* classes and other Java entities (such as interfaces)." (Emphasis added; Goldberg Declaration, ¶25.) Further, Goldstein does not disclose associating packages with roles. "Even if an association between packages and roles were added to Goldstein, this would still provide security at a different level of granularity than that provided by the claimed invention, since packages are not classes. The class-level security of the claimed invention is separate and distinct from a security domain at the package level contemplated by Goldstein." (Goldberg Declaration, ¶25.) Therefore, Goldstein does not teach or suggest "**establishing an association between said one or more protection domains and one or more classes of one or more objects**."

The deficiencies of Goldstein are not corrected by Shah. Shah appears to call the Role an "Identity class" for a cassette. (Shah at 2.) Shah states that "a Ticket is generated as a

representative for the identity." Goldstein discloses that "Tickets are use-once tokens of Roles." (Goldstein at 15.) However, Shah's "Identity class" is not the same as the class from which a cassette is instantiated. (Goldberg Declaration, ¶26.) Rather, "Identity class" appears to refer to the identity of the cassette that is established by the signature associated with the cassette. (Shah at 2.) (Goldberg Declaration, ¶26.) Shah does not teach or suggest that these Roles or Identity classes are associated with the class of the cassette, as required by the claims.

For at least the above reasons, the combination of Goldstein and Shah does not teach or suggest "**establishing an association between said one or more protection domains and one or more classes of one or more objects**," in independent claims 1, 10, and 19.

B. Goldstein Does Not Enable One Skilled in the Art to Carry Out the Claimed Invention

For a prior art reference to anticipate, it must be enabling. ("A prior art reference provides an enabling disclosure and thus anticipates a claimed invention if the reference describes the claimed invention in sufficient detail to enable a person of ordinary skill in the art to carry out the claimed invention." MPEP 2121 III.) Goldstein does not teach or suggest "**establishing an association between … protection domains and … classes.**" Further, Goldstein's speculation regarding future versions of Java does not describe "**establishing an association between … protection domains and … classes**" in sufficient detail to enable a person of ordinary skill in the art to carry out the claimed invention. (Goldberg Declaration, ¶27.) "Goldstein theorizes that "[p]erhaps in a future version of Java, another security domain may emerge." (Goldstein at 10.) However, Goldstein provides no teaching for what this other "security domain" may be or how to apply it to the JECF system, and certainly provides no hint that the future security domain would relate to classes. Goldstein's hypothetical statements about "another security domain" would not enable one skilled in the art to establish "**an association between … protection domains and … classes.**" "In 1997, the time of invention, one of ordinary skill in the art would not have been able to carry out the claimed invention of the '447 Patent based on the insufficient detail provided by Goldstein." (Goldberg Declaration, ¶27.)

C. The Rejection Under 102 Based On Goldstein as Supported by Shah Is Not Proper (Claims 1-24)

The Office Action rejects the claims as being anticipated by Goldstein with supporting evidence by Shah. (Office Action at 20.) While multiple references may be proper for an anticipation rejection in certain circumstances (see MPEP 2131.01), Shah does not (a) prove that Goldstein contains an enabled disclosure, (b) explain the meaning of a term used in Goldstein, or (c) show that a characteristic not disclosed in Goldstein is inherent.

The Office Action states that "Shah provides supporting evidence for Goldstein's teachings of the JECF (Java Electronic Commerce Framework)" by showing an inherent characteristic of the concepts taught by Goldstein. (Office Action at 21.) Patent Owner respectfully disagrees with this assertion. Shah is a JavaWorld article written by Rawn Shah. As Prof. Goldstein states: "Because the writer appears to be a reporter reporting on Goldstein's JECF project, rather than a scientist working on the project, Shah is imprecise in expressing terms of art and how JECF works. For example, regarding the function of 'roles' in JECF [which the Office Action asserts identifies the claimed '**protection domains**'], Goldstein states that '[r]oles represent the signature and are used to check Tickets.' (Goldstein at 15.) Shaw's statement, '[t]hese roles dictate the available resources and security levels and control that program's interface to the JECF code' (Shah at 2), however, is not quite right. Access to a resource may be granted to the requester depending on the role object provided by the requester within a ticket – but the role object itself does not dictate which resources can be accessed." (Emphasis added; Goldberg Declaration, ¶28.) Because Shah mischaracterizes the function of roles, as demonstrated by Goldstein, Shah cannot be used to support an assertion that these features are necessarily present in Goldstein or that Goldstein's "roles" inherently disclose the claimed "**protection domains**."

D. Goldstein Does Not Disclose the Claimed Association between Classes and Code Identifiers As Recited in Dependent Claims 2, 11, and 20

Claims 2, 11, and 20 recite, *inter alia*:

> "**at least one protection domain of said one or more protection domains is associated with a code identifier**."

The Office Action asserts that Goldstein discloses this feature through the disclosure of the claim chart provided by the third party requestor. The claim chart asserts that Goldstein discloses this feature because "Goldstein discloses a Role (i.e., a set of permissions or protection domain) associated with a digital signature corresponding to a public key/private key pair (e.g., a code identifier)." (Exhibit 9 at 15.) On the contrary, Goldstein does not teach or suggest that "**at least one protection domain of said one or more protection domains is associated with a code identifie**r."

Goldstein discloses that the "JECF uses digital signature to represent roles" and that a role "is essentially a public and private key pair." (Goldstein at 13.) "Goldstein does not teach that a role is *associated* with a digital signature. Rather, Goldstein teaches that a role *is* a digital signature." (Goldberg Declaration, ¶29.) The role cannot be both the claimed "**protection domain**" and the claimed "**code identifier**" as these are distinct features within the claim.

For at least the reasons above, Goldstein does not teach or suggest "**at least one protection domain of said one or more protection domains is associated with a code identifier**," as recited in claims 2, 11, and 20.

E. The Goldstein Rejection With Respect To Regarding Claims 7 and 16 Should Be Withdrawn

Claim 7 recites, *inter alia*:

> "**establishing an association between said one or more protection domains and one or more sources of code, wherein the one or more sources of code is at least one of a file, a persistent object, a FLASH_EPROM reader, or a set of system libraries**."

Claims 7 and 16 refer to an association between protection domains and "**sources of code**," rather than classes, as in claim 1. The rejection against claim 7 fails for the same reasons as the rejection against claim 1, discussed above.

The Office Action, via the Exhibit 9 claim chart, appears to identify Goldstein's "signer of a digital certificate" as the claimed "**source of code**," quoting that "Roles reify the trust relationship between two business entities." (Exhibit 9, pg 31.) As stated by Prof. Goldberg, "[t]he listed elements of claims 7 and 16 are not met by a signer of a digital certificate or a "business entity." In my opinion, a skilled artisan would not consider a file, a persistent object, a FLASH_EPROM reader, or a set of system libraries, to be even similar to a signer of a digital certificate or a business entity." (Goldberg Declaration, ¶30.)

For at least this reason and the reasons discussed above, Goldstein does not teach or suggest "**establishing an association between one or more protection domains and one or more sources of code**," as recited in independent claims 7 and 16.

F. The Goldstein Rejection With Respect to the Dependent Claims Should Be Withdrawn (Claims 2-6, 11-14, 20-24)

The dependent claims are distinguishable over Goldstein/Shah for at least the same reasons as their respective independent claims. For example, claims 2-6, 11-14, and 20-24 each recite features related to "**classes**." As discussed above, since Goldstein does not teach security based on classes, it cannot disclose any of these features.

Claims 4, 13, and 22, directed to "**a key associated with each class**" are distinguishable for at least the reasons discussed below regarding claims 2, 11, and 20 because the Office Action identifies that the key is part of the Role. (Exhibit 9 at 15.)

Claims 6, 15, and 24, directed to "**the step of associating said one or more protection domains and said one or more classes,**" are distinguishable over Goldstein at least by virtue of their dependence from their respective independent claims. Claims 8-9 and 17-18 are similarly distinguishable at least by virtue of their dependence from their respective independent claims.

VI. Conclusion

For at least these reasons, Patent Owner requests reconsideration and withdrawal of the rejections in the Office Action and confirmation of the patentability of claims 1-24 of the '447 Patent.

In the event that the Office determines that relief or fees (such as payment of a fee under 37 C.F.R. § 1.17 (g)) are required, the Patent Owner petitions for any required relief and authorizes the Commissioner to charge the cost of such petition and/or other fees due in connection with the filing of this document to **Deposit Account No. 03-1952** referencing **154892800300**.

Dated: August 29, 2011

Respectfully submitted,

Electronic Signature: /Julia Akhter /
Julia Akhter
Registration No.: 59,570
MORRISON & FOERSTER LLP
755 Page Mill Road
Palo Alto, California 94304-1018
(650) 813-5677

By Electronic Signature / Christopher B. Eide/
Christopher B. Eide
Registration No.: 48,375
MORRISON & FOERSTER LLP
755 Page Mill Road
Palo Alto, California 94304-1018
(650) 813-5720