

EXHIBIT 8

U.S. Patent No. 5,412,717
“Computer System Security Method And Apparatus Having Program Authorization Information Data Structures”
Inventors: Addison M. Fischer
Filing Date: May 15, 1992
Priority Date: May 15, 1992
Date of Issue: May 2, 1995
 (“*Fischer*”)

U.S. Patent No. 6,125,447 – Claim 1	<i>Fischer</i>
1. A method for providing security, the method comprising the steps of:	<p><i>Fischer</i> discloses a method for providing security.</p> <p>“More particularly, the invention relates to a method and apparatus for providing enhanced computer system security while processing computer programs, particularly those of unknown origin, which are transmitted among users.”</p> <p><i>Fischer</i>, 1:20-25.</p> <p>“Thus, the present invention advantageously protects a user from any program to be executed. The present invention is particularly advantageous in light of current data processing practices where programs are obtained from a wide range of diverse, untrustworthy places such as computer bulletin boards or other users of unknown trustworthiness.”</p> <p><i>Fischer</i>, 2:49-55.</p>
establishing one or more protection domains, wherein a protection domain is associated with zero or more permissions;	<p><i>Fischer</i> discloses establishing one or more protection domains, wherein a protection domain is associated with zero or more permissions.</p> <p>For example, <i>Fischer</i> discloses that the system monitor builds (i.e., establishes) a Program Authorization Information (“PAI”) data structure as a protection domain:</p>

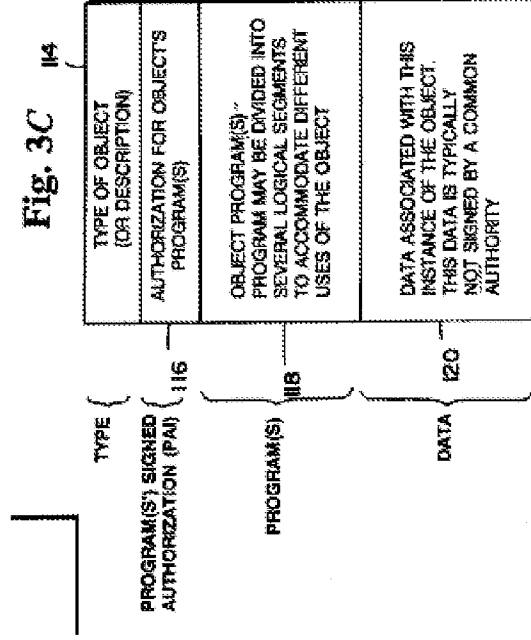
U.S. Patent No. 6,125,447 – Claim 1	Fischer
	<p>“The present method an apparatus utilizes a unique operation system design that includes a system monitor which limits the ability of a program about to be executed to the use of predefined resources (e.g., data files, disk writing capabilities etc.). The system monitor builds a data structure including a set of authorities defining that which a program is permitted to do and/or that which the program is precluded from doing.</p> <p>The set of authorities and/or restrictions assigned to a program to be executed are referred to herein as ‘program authorization information’ (or ‘PAI’). Once defined, the program authorization information is thereafter associated with each program to be executed to thereby delineate the types of resources and functions that the program is allowed to utilize.”</p> <p><i>Fischer</i>, 2:16-30.</p> <p><i>Fischer</i> further discloses that PAI information for a program may be combined, as appropriate, with the PAI associated with a calling program. :</p> <p>“Thereafter, the program X’s program authorizing information is combined, as appropriate, with the PAI associated with the PCB of the calling program, if any. This combined PAI, which may include multiple PAI’s, is then stored in an area of storage which cannot generally be modified by the program and the address of the PAI is stored in the process control block (PCB) as indicated in field 156 of FIG. 5. Thus, if program X is called by a calling program, it is subject to all its own constraints as well as being combined in some way with the constraints of the calling program, which aggregate constraints are embodied into program X’s PAI. In this fashion, a calling program may not be permitted to exceed its assigned bounds by merely calling another program.”</p> <p><i>Fischer</i>, 19:40-54.</p> <p><i>Fischer</i> further discloses that the PAI is associated with zero or more permissions, such as a range of operations that a program may execute or may be precluded from executing:</p>

U.S. Patent No. 6,125,447 – Claim 1	Fischer
	<p>“The PAI defines the range of operations that a program may execute and/or defines those operations that a program cannot perform. The program is permitted to access what has been authorized and nothing else. In this fashion, the program may be regarded as being placed in a program capability limiting ‘safety box.’ This ‘safety box’ is thereafter associated with the program such that whenever the system monitor runs the program, the PAI for that program is likewise loaded and monitored. When the program is to perform a function or access a resource, the associated PAI is monitored to confirm that the operation is within the defined program limits. If the program attempts to do anything outside the authorized limits, then the program execution is halted.”</p> <p><i>Fischer</i>, 2:34-48. Indeed, <i>Fischer</i> discloses a PAI associated with zero permissions (e.g., “no known trustworthiness” that leads to “a wide range of restrictions”), and a PAI associated with more permissions (e.g., “an unlimited number of different resources and functions to be controlled”):</p> <p>“Even programs with no known trustworthiness can be used after program authorization information associates a wide range of restrictions to thereby allow potentially beneficial programs to be safely used—even if they do not have an official certification of trust.</p> <p>The present invention also allows an unlimited number of different resources and functions to be controlled. For example, some useful resources/functions which may be controlled include: the ability to limit a program to certain files or data sets; the ability to transmit data via electronic mail to someone outside the user’s domain; the ability of a program to create or solicit digital signatures; the ability to limit access to a program of certain security classes, etc.”</p> <p><i>Fischer</i>, 3:48-61.</p>
establishing an association between said one or more protection domains and one or more classes of one or more objects;	<i>Fischer</i> discloses establishing an association between said one or more protection domains and one or more classes of one or more objects.

U.S. Patent No. 6,125,447 – Claim 1	Fischer
and	<p>For example, <i>Fischer</i> discloses that a PAI (i.e., a protection domain) is associated with a program:</p> <p>“Once defined, the program authorization information [(PAI)] is thereafter associated with each program to be executed to thereby delineate the types of resources and functions that the program is allowed to utilize. The PAI associated with a particular program may be assigned by a computer system owner/user or by someone who the computer system owner/user implicitly trusts.”</p> <p><i>Fischer</i>, 2:26-33.</p> <p><i>Fischer</i> further discloses that a program with which a PAI is associated may be part of an object:</p> <p>“The present invention is directed to providing reliable security, even when operating with complex data structures, e.g., objects, containing their own program instructions, which are transmitted among users.”</p> <p><i>Fischer</i>, 2:6-9.</p> <p>“Through the use of the present invention, general object oriented data may be transferred from user to user without exposing users to the potential dangers of viruses or mischievous users.”</p> <p><i>Fischer</i>, 4:10-13.</p> <p>“In one contemplated embodiment of the present invention, programs may be part of data objects, which are written in a high-level control language and are executed by a standardized interpreter program which executes this high-level language. In this case, part of the interpreter’s task is to verify that the functions encountered in the high level logic are, in fact, permissible. If such tasks are not permissible, the interpreter then suppresses the execution of the program not authorized to perform such tasks.”</p>

U.S. Patent No. 6,125,447 – Claim 1	Fischer
	<p><i>Fischer</i>, 3:11-20.</p> <p>“In accordance with the present invention, a PAI is associated with programs to be executed. FIGS. 3A through 3D depict four exemplary approaches for associating program authorization information with a program. . . .</p> <p>...</p> <p>FIG. 3C shows an important application in which a PAI data structure is associated with a program according to an embodiment of the present invention. FIG. 3C shows an illustrative data structure for a secure exchangeable ‘object’. The data structure may be signed by a trusted authority. The signing of such a data structure allows the object to be securely transmitted from user to user. Although the data structure shown in FIG. 3 is set forth in a general format, it may be structured as set forth in the inventor’s copending application filed on Apr. 6, 1992 and entitled ‘Method and Apparatus for Creating, Supporting and Processing a Travelling Program’ (U.S. Ser. No. 07/863,552.), which application is hereby expressly incorporated herein by reference.</p> <p>... The program authorization information is embedded in a segment 116 which specifies the authorization for the object’s program or programs in a manner to be described more fully hereinafter.”</p> <p><i>Fischer</i>, 7:14-18, 7:49-8:2.</p> <p>“Thereafter, the PAI is stored using, for example, one of the approaches set forth in FIGS. 3A through 3D so that it is associated with its program 272”</p> <p><i>Fischer</i>, 15:24-26. As just described, FIG. 3C discloses the program as part of an object/class. See <i>Fischer</i>, 7:49-8:2.</p> <p>To the extent <i>Fischer</i> does not expressly disclose “classes” of the objects, one of ordinary</p>

skill in the art would understand that a class, as that term is used in the '447 Patent, is a necessarily present feature of the objects disclosed in *Fischer*. To be sure, Figure 3C in *Fischer* shows an object-oriented data structure including a type segment (114), program (e.g., method) segment (118), and a data segment (120), necessarily implicating an object-oriented program architecture.



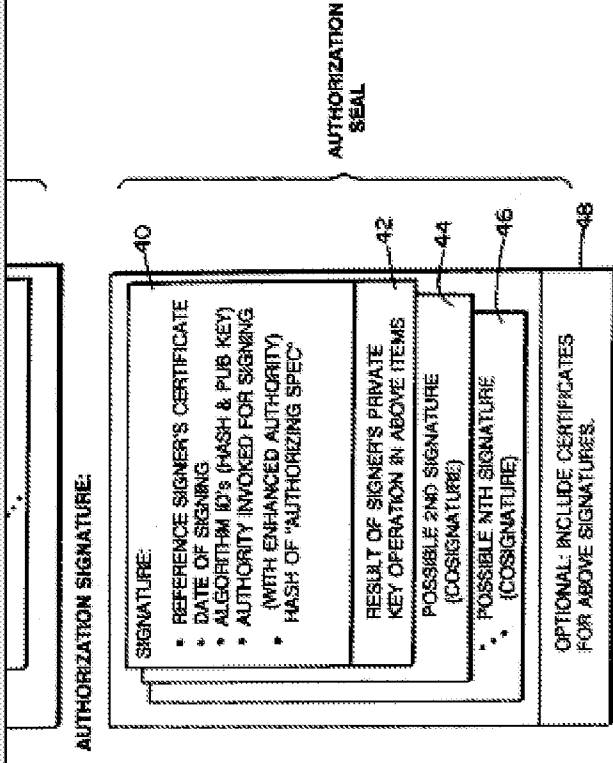
Fischer, FIG. 3C.

The '447 Patent discloses that a class is a high-level abstraction or definition of an object, such that "[e]ach object belonging to a class has the same fields ('attributes') and the same methods." '447 Patent, 7:4-5; *see also*, '447 Patent, 6:63-7:25.

Thus, *Fischer*'s disclosure of a data object in Figure 3C that includes fields/attributes (labeled "DATA ASSOCIATED WITH THIS INSTANCE OF THE OBJECT") as well as methods (labeled "OBJECT PROGRAM(S)") is consistent with the the '447 Patent's description of a class. In addition, Figure 3C of *Fischer* mentions that the depicted object is an "instance," which further shows that *Fischer*'s disclosure includes object-oriented data structures, which

U.S. Patent No. 6,125,447 – Claim 1	Fischer
	<p>are necessarily part of a class, as that term is used in the '447 Patent. <i>See</i> '447 Patent, 7:7-8 ("An object is said to be an 'instance' of the class to which the object belongs.").</p> <p>Additionally, the '447 Patent admits that the ideas of classes and object instances were well known to those skilled in the art: "[C]lass definitions are generated from source code written by a programmer. For example, a programmer using a Java Development Kit enters source code that conforms to the Java programming language into a source file. The source code embodies class definitions and other instructions which are used to generate byte code which controls the execution of a code executor (i.e. a Java virtual machine). <i>Techniques for defining classes and generating code executed by a code executor, such as a Java virtual machine, are well known to those skilled in the art.</i>" '447 Patent, 7:15-24.</p> <p><i>Fischer's</i> disclosure is entirely consistent with this admission, as <i>Fischer</i> discloses that "programs may be part of data objects, which are written in a high-level control language and are executed by a standardized interpreter program which executes the high-level language." <i>Fischer</i>, 3:12-15. In light of these disclosures in the '447 Patent and <i>Fischer</i>, there can be no doubt that one of ordinary skill in the art would view the object disclosed in <i>Fischer</i> as an instance of a class, such that the class, if not expressly disclosed, is necessarily present in the <i>Fischer</i> disclosure.</p>
<p>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.</p>	<p><i>Fischer</i> discloses 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.</p> <p>For example,</p> <p>“FIGS. 10 and 11 illustrate the sequence of operations of a supervisor program for controlling the processing of a program being executed in accordance with its program authorization information.”</p> <p><i>Fischer</i>, 15:56-59.</p> <p>“Depending on the processing in block 316 [of FIG. 10], a decision is made in block 322</p>

U.S. Patent No. 6,125,447 – Claim 1	<i>Fischer</i>
	<p>whether the signatures are valid, authorized and trusted. If the signatures are not determined to be valid, then the routing branches to block 324 where the execution in program X is suppressed.”</p> <p><i>Fischer</i>, 16:66-17:3.</p> <p>“If the processing in blocks 322 and 316 reveal that the signatures are valid, then the processing in block 326 is performed.”</p> <p><i>Fischer</i>, 17:31-33.</p>
U.S. Patent No. 6,125,447 – Claim 2	<i>Fischer</i>
<p>2. The method of claim 1, wherein:</p> <p>at least one protection domain of said one or more protection domains is associated with a code identifier;</p>	<p><i>Fischer</i> discloses the method of claim 1. See claim chart above for further details.</p> <p><i>Fischer</i> discloses at least one protection domain of said one or more protection domains is associated with a code identifier.</p> <p>For example, the ‘447 Patent discloses a code identifier as “describing the source of code that defines a class, a set of public cryptographic keys associated with the source of code, or other information which describes the source of code, or any combination thereof. A ‘source of code’ is an entity from which computer instructions are received. Examples 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 Patent, 3:13-21. Figure 3 of the ‘447 Patent discloses a policy file that includes a URL (i.e., file://somesource) and a key name (i.e., “somekey”), and describes both as code identifiers. ‘447 Patent, 9:26-37 & Fig. 3.</p> <p>Similar to this disclosure of the “code identifier” in the ‘447 Patent, Figure 2 in <i>Fischer</i> discloses the PAI data structure (i.e., protection domain data structure), which explicitly associates the protection domain with a “source of code” such as the signer of a digital certificate:</p>



“The authorization signature includes a signature segment 40. The signature segment 40 may include a reference to the signer's certificate, i.e., an identifier for identifying the signer's certificate. In accordance with a preferred embodiment of the present invention, such a digital certificate is a digital message created by a trusted entity which contains the user's public key and the name of the user (which is accurate to the entity's satisfaction) and possibly a representation of the authority which has been granted to the user by the party who signs the digital message.”

Fischer, 6:25-35 (emphasis added) & Fig. 2 (excerpted).

Fischer expressly discloses that the digital signature (i.e., code identifier) may be associated with a manufacturer of the program (i.e., a source of code or “an entity from which computer instructions are received”):

“The present invention allows PAI information to be associated in any appropriate

U.S. Patent No. 6,125,447 – Claim 2	Fischer
	<p><i>manner, so that in principle a user could define one or more levels of PAI which are then combined together with perhaps a more universal PAI, or with a PAI which was signed and supplied by the or [sic] manufacturer of this program.”</i></p> <p><i>Fischer, 9:3-8 (emphasis added).</i></p> <p>“FIGS. 6 through 9 is a flowchart illustrating an exemplary sequence of operations of a utility program for establishing program authorization information. Such a utility program prompts a user, i.e., the end user, the user’s agent, or even the manufacturer, to define a range of authorities which are associated with a program to be executed by the user’s system.”</p> <p><i>Fischer, 11:7-13 (emphasis added).</i></p> <p>“If no PAI has yet been associated with the program, then a check is made to determine whether the program has an associated signed ‘pedigree’ from the manufacturer (306). Thus, if a well known manufacturer of programs has signed the program with a public key or digital certificate, then, if desired, such a program may be assigned whatever level of authority desired depending upon how much the manufacturer is trusted and the system may permit execution of such program. Such a digital signature from the manufacturer can be used to verify that the associated program had not been infected with a virus since it can be determined whether or not the program is exactly the same as it was when it was generated by the manufacturer.”</p> <p><i>Fischer, 16:12-25 (emphasis added).</i></p>
at least one class of said one or more classes is associated with said code identifier; and	<p><i>Fischer</i> discloses at least one class of said one or more classes is associated with said code identifier. For example, as discussed above, Figures 2 and 3C of <i>Fischer</i> disclose that the PAI data structure may contain a manufacturer’s signature (i.e., code identifier) (see Fig. 2), 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.</p>

U.S. Patent No. 6,125,447 – Claim 2	<div data-bbox="191 184 849 1383" data-label="Diagram"> </div> <p data-bbox="776 285 849 1383"><i>Fischer</i>, Fig. 3C (“PROGRAM(S)’ SIGNED AUTHORIZATION (PAI)” included as element 116 of the disclosed object/class data structure).</p>
<p data-bbox="849 1383 1255 1919">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.</p>	<p data-bbox="849 184 1255 1383"><i>Fischer</i> discloses associating said one or more protection domains and said one or more classes based on said code identifier. For example, <i>Fischer</i> discloses that authorization (e.g., protection domains) may be associated with a program based on the digital signature (i.e., code identifier) included in an object/class:</p> <p data-bbox="1036 184 1182 1383">“Thus, if a well known manufacturer of programs has signed the program with a public key or digital certificate, then, if desired, <i>such a program may be assigned whatever level of authority desired depending upon how much the manufacturer is trusted</i> and the system may permit execution of such program.”</p> <p data-bbox="1214 909 1255 1383"><i>Fischer</i>, 16:15-20 (emphasis added).</p>
U.S. Patent No. 6,125,447 – Claim 3	<p data-bbox="1287 184 1408 1383"><i>Fischer</i> discloses the code identifier indicates a source of code used to define each class of said one or more classes. For example, <i>Fischer</i> indicates that each object/class may</p>

<p>U.S. Patent No. 6,125,447 – Claim 3</p> <p>code used to define each class of said one or more classes.</p>	<p><i>Fischer</i></p> <p>include the digital signature (i.e., a code identifier):</p> <p>“FIG. 3C shows an illustrative data structure for a secure exchangeable ‘object’. The data structure may be signed by a trusted authority. The signing of such a data structure allows the object to be securely transmitted from user to user.”</p> <p><i>Fischer</i>, 7:51-56.</p>
<p>U.S. Patent No. 6,125,447 – Claim 4</p> <p>4. The method of claim 2, wherein said code identifier indicates a key associated with each class of said one or more classes.</p>	<p><i>Fischer</i></p> <p><i>Fischer</i> discloses the code identifier indicates a key associated with each class of said one or more classes. For example, Figure 2 in <i>Fischer</i> discloses the digital certificate (i.e., code identifier) includes a public key that may be associated with the object/class:</p> <p>“The authorization signature includes a signature segment 40. The signature segment 40 may include a reference to the signer’s certificate, i.e., an identifier for identifying the</p>

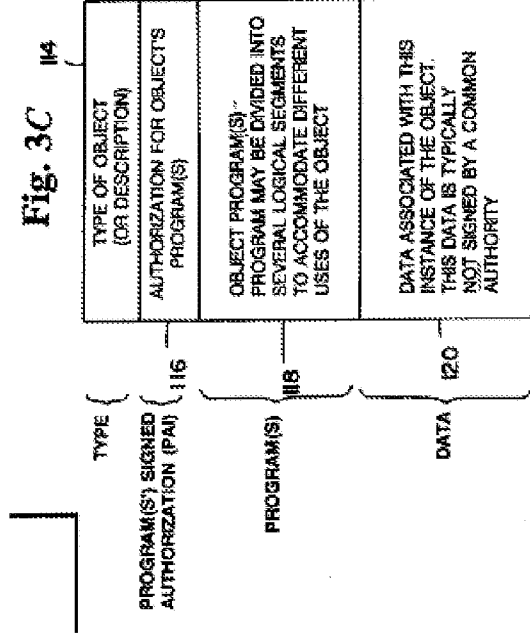
U.S. Patent No. 6,125,447 – Claim 4		<p style="text-align: center;"><i>Fischer</i></p> <p>signer's certificate. In accordance with a preferred embodiment of the present invention, such a digital certificate is a digital message created by a trusted entity <i>which contains the user's public key</i> and the name of the user (which is accurate to the entity's satisfaction) and possibly a representation of the authority which has been granted to the user by the party who signs the digital message.”</p> <p><i>Fischer</i>, 6:25-35 (emphasis added) & Fig. 2 (excerpted).</p>
U.S. Patent No. 6,125,447 – Claim 5	<p>5. 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 . . .</p>	<p style="text-align: center;"><i>Fischer</i></p> <p><i>Fischer</i> discloses the code identifier indicates a source of code used to define each class of said one or more classes.</p> <p>For example, the ‘447 Patent discloses the “source of code” of a code identifier as “an entity from which computer instructions are received. Examples 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 Patent, 3:15-21.</p> <p><i>Fischer</i> expressly discloses that the digital signature (i.e., code identifier) may be associated with a manufacturer of the program (i.e., a source of code or “an entity from which computer instructions are received”):</p> <p>“The present invention <i>allows PAI information to be associated in any appropriate manner</i>, so that in principle a user could define one or more levels of PAI which are then combined together with perhaps a more universal PAI, <i>or with a PAI which was signed and supplied by the or [sic] manufacturer of this program.</i>”</p> <p><i>Fischer</i>, 9:3-8 (emphasis added).</p> <p>“FIGS. 6 through 9 is a flowchart illustrating an exemplary sequence of operations of a utility program for establishing program authorization information. Such a utility program prompts a user, i.e., the end user, the user's agent, <i>or even the manufacturer</i>, to</p>

U.S. Patent No. 6,125,447 – Claim 5	Fischer
	<p>define a range of authorities which are associated with a program to be executed by the user's system.”</p> <p><i>Fischer</i>, 11:7-13 (emphasis added).</p> <p>“If no PAI has yet been associated with the program, then a check is made to determine whether the program has an associated signed ‘pedigree’ from the manufacturer (306). Thus, if a well known manufacturer of programs has signed the program with a public key or digital certificate, then, if desired, such a program may be assigned whatever level of authority desired depending upon how much the manufacturer is trusted and the system may permit execution of such program. Such a digital signature <i>from the manufacturer can be used to verify that the associated program had not been infected with a virus since it can be determined whether or not the program is exactly the same as it was when it was generated by the manufacturer.</i>”</p> <p><i>Fischer</i>, 16:12-25 (emphasis added).</p>
<p>... [wherein said code identifier] indicates a key associated with each class of said one or more classes.</p>	<p><i>Fischer</i> discloses the code identifier indicates a key associated with each class of said one or more classes. For example Figure 2 in <i>Fischer</i> discloses the digital certificate (i.e., code identifier) includes a public key may be associated with the object/class:</p>

U.S. Patent No. 6,125,447 – Claim 5	<div data-bbox="196 730 228 840" data-label="Text">Fischer</div> <div data-bbox="293 890 318 1167" data-label="Text">AUTHORIZATION SIGNATURE:</div> <div data-bbox="331 420 834 1142" data-label="Diagram"> </div>
-------------------------------------	--

U.S. Patent No. 6,125,447 – Claim 6	<div data-bbox="1230 730 1263 840" data-label="Text">Fischer</div> <div data-bbox="1268 201 1377 1377" data-label="Text"> <p>Fischer discloses 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.</p> </div>
-------------------------------------	--

U.S. Patent No. 6,125,447 – Claim 6	Fischer
<p>identifier further includes associating said one or more protection domains and said one or more classes based on data persistently stored,</p>	<p>For example, the ‘447 Patent discloses persistently stored data such as instructions stored in a file, mappings stored in a database system, and mapping attributes of a persistent object:</p> <p>“Storing instructions in a file is just one method of representing the policy of the system with persistently stored data. Other methods are possible for representing the policy with persistent data. For example, data in a database system can be used to map code identifiers to authorized permissions, or attributes of a persistent object can be used to map code identifiers to authorized permissions.”</p> <p>‘447 Patent, 9:19-25.</p> <p>Similar to this disclosure of the “persistently stored” data in the ‘447 Patent, <i>Fischer</i> discloses that the association between the protection domains and the one or more classes is based on data that is persistently stored as an attribute of a persistent object:</p> <p>“FIG. 3C shows an important application in which a PAI data structure is associated with a program according to an embodiment of the present invention. FIG. 3C shows an illustrative data structure for a secure exchangeable ‘object’. The data structure may be signed by a trusted authority. The signing of such a data structure allows the object to be securely transmitted from user to user. Although the data structure shown in FIG. 3 is set forth in a general format, it may be structured as set forth in the inventor’s copending application filed on Apr. 6, 1992 and entitled ‘Method and Apparatus for Creating, Supporting and Processing a Travelling Program’ (U.S. Ser. No. 07/863,552.), which application is hereby expressly incorporated herein by reference.</p> <p>... The program authorization information is embedded in a segment 116 which specifies the authorization for the object’s program or programs in a manner to be described more fully hereinafter.”</p> <p><i>Fischer</i>, 7:49-8:2.</p> <p>Figure 3C in <i>Fischer</i> shows these attributes of a persistent object:</p>

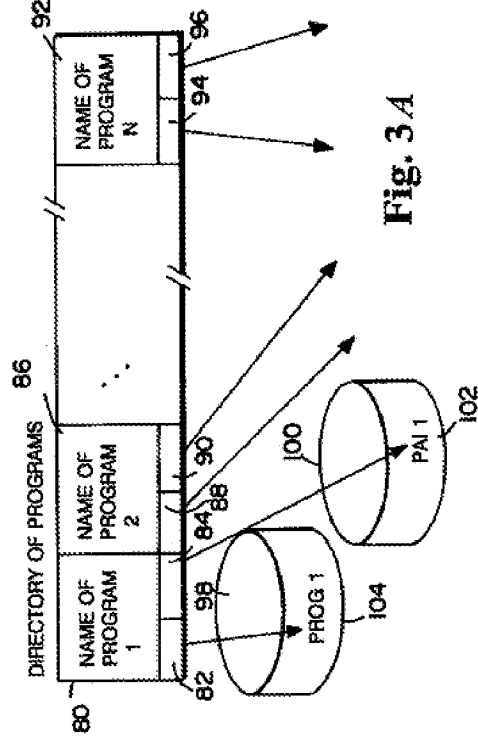


Fischer, FIG. 3C (see, e.g., element 116).

In addition to its disclosure of persistent data stored as an attribute of a persistent object, *Fischer* discloses more generally associating the PAI information (i.e., protection domains) based on other types of persistently stored data. For example, *Fischer* discloses storing PAI information on a separate/remote storage device or in the same memory as the program:

“FIG. 3A shows an exemplary schematic representation of a system’s directory of programs. . . .

. . . Additionally, associated with each of the program related identifiers is an indicator 84, 90, . . . 96, respectively, which identifies the location of its associated program authorization information, e.g., PAI 1. Although the program authorization information, PAI 1, is depicted as being stored in a separate memory device 100, it may, if desired, be stored in the same memory media as its associated program.”



Fischer, 7:20-35 & Fig. 3A.

wherein said data associates code identifiers with a set of one or more permissions.

Fischer discloses that the persistently stored data associates code identifiers with a set of one or more permissions. As discussed above, *Fischer* discloses that the persistently stored PAI segment of the Figure 3C object/class associates code identifiers (e.g., a digital signature) with a set of one or more permissions:

“The present method an apparatus utilizes a unique operation system design that includes a system monitor which limits the ability of a program about to be executed to the use of predefined resources (e.g., data files, disk writing capabilities etc.). The system monitor builds a data structure including a set of authorities defining that which a program is permitted to do and/or that which the program is precluded from doing.

The set of authorities and/or restrictions assigned to a program to be executed are referred to herein as ‘program authorization information’ (or ‘PAI’). Once defined, the program authorization information is thereafter associated with each program to be executed to thereby delineate the types of resources and functions that the program is allowed to utilize.”

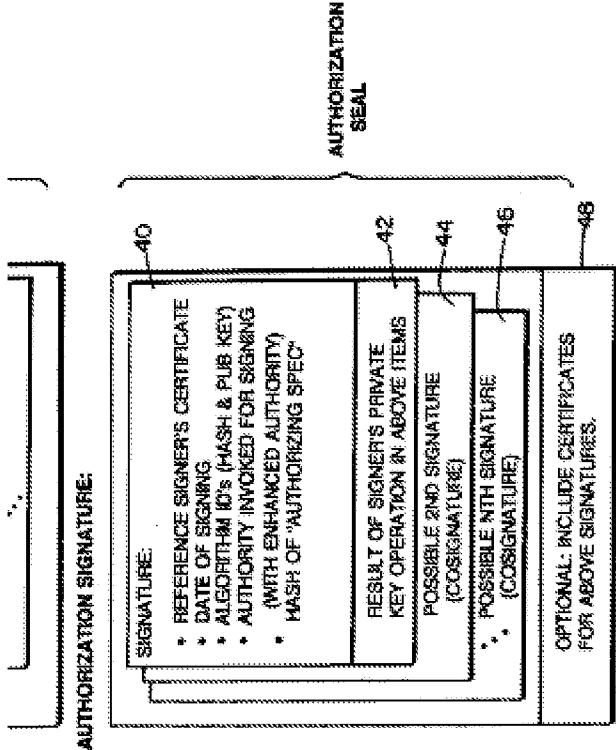
U.S. Patent No. 6,125,447 – Claim 6	Fischer
	<p><i>Fischer</i>, 2:16-30.</p> <p>“The PAI defines the range of operations that a program may execute and/or defines those operations that a program cannot perform. The program is permitted to access what has been authorized and nothing else. In this fashion, the program may be regarded as being placed in a program capability limiting ‘safety box.’ This ‘safety box’ is thereafter associated with the program such that whenever the system monitor runs the program, the PAI for that program is likewise loaded and monitored. When the program is to perform a function or access a resource, the associated PAI is monitored to confirm that the operation is within the defined program limits. If the program attempts to do anything outside the authorized limits, then the program execution is halted.”</p> <p><i>Fischer</i>, 2:34-48.</p> <p>“Even programs with no known trustworthiness can be used after program authorization information associates a wide range of restrictions to thereby allow potentially beneficial programs to be safely used—even if they do not have an official certification of trust.</p> <p>The present invention also allows an unlimited number of different resources and functions to be controlled. For example, some useful resources/functions which may be controlled include: the ability to limit a program to certain files or data sets; the ability to transmit data via electronic mail to someone outside the user’s domain; the ability of a program to create or solicit digital signatures; the ability to limit access to a program of certain security classes, etc.”</p> <p><i>Fischer</i>, 3:48-61.</p> <p>“Additionally, in block 340, an examination is made of the PAI information stored in the process control block. As a follow up to, or associated with, the processing in block 340, a check is made in block 342 to determine whether the examined PAI is allowed access to the required resources or allowed to perform the required functions. For example, if an</p>

U.S. Patent No. 6,125,447 – Claim 6	<i>Fischer</i>
	<p>attempt is made to use electronic mail, a check is made of the PAI to determine whether the program is authorized to perform electronic mail functions and if so whether the mailing is limited to a set of mail identifiers.</p> <p>If the check at 342 reveals that the PAI does not allow the attempted function or resource access, then a error message is generated in block 344 to indicated that the program is attempting to exceed its limits, access to the resource or function is denied and an appropriate error code or message is generated. . . .</p> <p>If the check in block 342 reveals that the PAI does allow access to the function or resource, then a check is made in block 346 to apply conventional access controls to ensure that the user of the program is still within the bounds of his authority.”</p> <p><i>Fischer</i>, 19:16-33, 19:51-55.</p>

U.S. Patent No. 6,125,447 – Claim 7	<i>Fischer</i>
<p>7. A method of providing security, the method comprising the steps of:</p>	<p><i>Fischer</i> discloses a method of providing security.</p> <p>“More particularly, the invention relates to a method and apparatus for providing enhanced computer system security while processing computer programs, particularly those of unknown origin, which are transmitted among users.”</p> <p><i>Fischer</i>, 1:20-25.</p> <p>“Thus, the present invention advantageously protects a user from any program to be executed. The present invention is particularly advantageous in light of current data processing practices where programs are obtained from a wide range of diverse, untrustworthy places such as computer bulletin boards or other users of unknown trustworthiness.”</p> <p><i>Fischer</i>, 2:49-55.</p>

U.S. Patent No. 6,125,447 – Claim 7	<i>Fischer</i>
<p>establishing one or more protection domains, wherein a protection domain is associated with zero or more permissions;</p>	<p><i>Fischer</i> discloses establishing one or more protection domains, wherein a protection domain is associated with zero or more permissions.</p> <p>For example, <i>Fischer</i> discloses that the system monitor builds (i.e., establishes) a Program Authorization Information (“PAI”) data structure as a protection domain:</p> <p>“The present method an apparatus utilizes a unique operation system design that includes a system monitor which limits the ability of a program about to be executed to the use of predefined resources (e.g., data files, disk writing capabilities etc.). The system monitor builds a data structure including a set of authorities defining that which a program is permitted to do and/or that which the program is precluded from doing.</p> <p>The set of authorities and/or restrictions assigned to a program to be executed are referred to herein as ‘program authorization information’ (or ‘PAI’). Once defined, the program authorization information is thereafter associated with each program to be executed to thereby delineate the types of resources and functions that the program is allowed to utilize.”</p> <p><i>Fischer</i>, 2:16-30.</p> <p><i>Fischer</i> further discloses that PAI information for a program may be combined, as appropriate, with the PAI associated with a calling program. :</p> <p>“Thereafter, the program X’s program authorizing information is combined, as appropriate, with the PAI associated with the PCB of the calling program, if any. This combined PAI, which may include multiple PAI’s, is then stored in an area of storage which cannot generally be modified by the program and the address of the PAI is stored in the process control block (PCB) as indicated in field 156 of FIG. 5. Thus, if program X is called by a calling program, it is subject to all its own constraints as well as being combined in some way with the constraints of the calling program, which aggregate constraints are embodied into program X’s PAI. In this fashion, a calling program may not be permitted to exceed its assigned bounds by merely calling another program.”</p>

U.S. Patent No. 6,125,447 – Claim 7	<i>Fischer</i>
	<p><i>Fischer</i>, 19:40-54.</p> <p><i>Fischer</i> further discloses that the PAI is associated with zero or more permissions, such as a range of operations that a program may execute or may be precluded from executing:</p> <p>“The PAI defines the range of operations that a program may execute and/or defines those operations that a program cannot perform. The program is permitted to access what has been authorized and nothing else. In this fashion, the program may be regarded as being placed in a program capability limiting ‘safety box.’ This ‘safety box’ is thereafter associated with the program such that whenever the system monitor runs the program, the PAI for that program is likewise loaded and monitored. When the program is to perform a function or access a resource, the associated PAI is monitored to confirm that the operation is within the defined program limits. If the program attempts to do anything outside the authorized limits, then the program execution is halted.”</p> <p><i>Fischer</i>, 2:34-48. Indeed, <i>Fischer</i> discloses a PAI associated with zero permissions (e.g., “no known trustworthiness” that leads to “a wide range of restrictions”), and a PAI associated with more permissions (e.g., “an unlimited number of different resources and functions to be controlled”):</p> <p>“Even programs with no known trustworthiness can be used after program authorization information associates a wide range of restrictions to thereby allow potentially beneficial programs to be safely used--even if they do not have an official certification of trust.</p> <p>The present invention also allows an unlimited number of different resources and functions to be controlled. For example, some useful resources/functions which may be controlled include: the ability to limit a program to certain files or data sets; the ability to transmit data via electronic mail to someone outside the user’s domain; the ability of a program to create or solicit digital signatures; the ability to limit access to a program of certain security classes, etc.”</p>

U.S. Patent No. 6,125,447 – Claim 7	Fischer
<p>establishing an association between said one or more protection domains and one or more sources of code; and</p>	<p><i>Fischer</i>, 3:48-61.</p> <p><i>Fischer</i> discloses establishing an association between said one or more protection domains and one or more sources of code.</p> <p>For example, the ‘447 Patent discloses a “source of code” as “an entity from which computer instructions are received. Examples 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 Patent, 3:15-21.</p> <p>Similar to this disclosure of the “source of code” in the ‘447 Patent, <i>Fischer</i> discloses the PAI data structure (i.e., protection domain data structure), which explicitly associates the protection domain with a “source of code” such as the signer of a digital certificate:</p> 

U.S. Patent No. 6,125,447 – Claim 7	Fischer
	<p>“The authorization signature includes a signature segment 40. <i>The signature segment 40 may include a reference to the signer’s certificate, i.e., an identifier for identifying the signer’s certificate.</i> In accordance with a preferred embodiment of the present invention, such a digital certificate is a digital message created by a trusted entity which contains the user’s public key and the name of the user (which is accurate to the entity’s satisfaction) and possibly a representation of the authority which has been granted to the user by the party who signs the digital message.”</p> <p><i>Fischer, 6:25-35 (emphasis added) & Fig. 2 (excerpted).</i></p> <p><i>Fischer</i> expressly discloses that the digital signature may be associated with a manufacturer of the program (i.e., a source of code or “an entity from which computer instructions are received”):</p> <p>“The present invention <i>allows PAI information to be associated in any appropriate manner</i>, so that in principle a user could define one or more levels of PAI which are then combined together with perhaps a more universal PAI, <i>or with a PAI which was signed and supplied by the or [sic] manufacturer of this program.</i>”</p> <p><i>Fischer, 9:3-8 (emphasis added).</i></p> <p>“FIGS. 6 through 9 is a flowchart illustrating an exemplary sequence of operations of a utility program for establishing program authorization information. Such a utility program prompts a user, i.e., the end user, the user’s agent, <i>or even the manufacturer</i>, to define a range of authorities which are associated with a program to be executed by the user’s system.”</p> <p><i>Fischer, 11:7-13 (emphasis added).</i></p> <p>“If no PAI has yet been associated with the program, then a check is made to determine whether the program has an associated signed ‘pedigree’ from the manufacturer (306). Thus, if a well known manufacturer of programs has signed the program with a public key</p>

U.S. Patent No. 6,125,447 – Claim 7	Fischer
	<p>or digital certificate, then, if desired, such a program may be assigned whatever level of authority desired depending upon how much the manufacturer is trusted and the system may permit execution of such program. Such a digital signature <i>from the manufacturer can be used to verify that the associated program had not been infected with a virus since it can be determined whether or not the program is exactly the same as it was when it was generated by the manufacturer.</i>”</p> <p><i>Fischer</i>, 16:12-25 (emphasis added).</p>
<p>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.</p>	<p><i>Fischer</i> discloses 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.</p> <p>For example, based on the digital signature (i.e., a source of code identifying the manufacturer of the code), <i>Fischer</i> discloses determining whether a program that is executing is permitted to perform an action:</p> <p>“FIGS. 10 and 11 illustrate the sequence of operations of a supervisor program for controlling the processing of a program being executed in accordance with its program authorization information.”</p> <p><i>Fischer</i>, 15:56-59.</p> <p>“Depending on the processing in block 316 [of FIG. 10], a decision is made in block 322 whether the signatures are valid, authorized and trusted. If the signatures are not determined to be valid, then the routing branches to block 324 where the execution in program X is suppressed.”</p> <p><i>Fischer</i>, 16:66-17:3.</p> <p>“If the processing in blocks 322 and 316 reveal that the signatures are valid, then the</p>

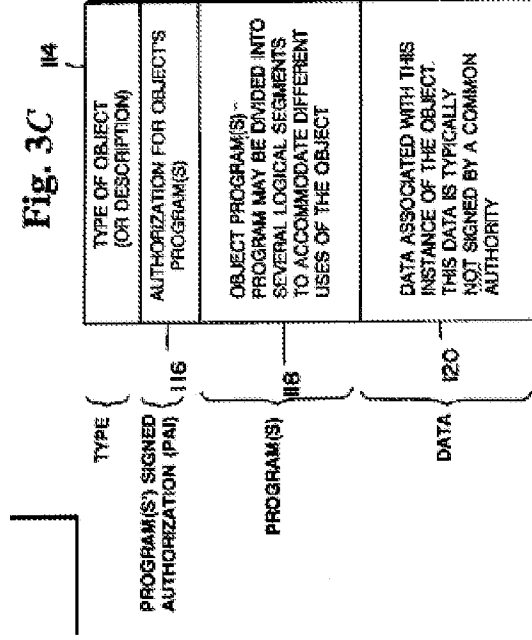
U.S. Patent No. 6,125,447 – Claim 7	<div data-bbox="191 184 228 1383"> <p><i>Fischer</i></p> </div> <div data-bbox="228 184 524 1383"> <p>processing in block 326 is performed.”</p> <p><i>Fischer</i>, 17:31-33.</p> <p>Accordingly, <i>Fischer</i> is clear that execution of the code is conditioned on verifying the source of code (i.e., digital signature), and that the requested action will not be permitted if the signature is not valid.</p> </div>
<div data-bbox="565 1383 602 1917"> <p>U.S. Patent No. 6,125,447 – Claim 8</p> </div> <div data-bbox="602 1383 993 1917"> <p>8. 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.</p> </div>	<div data-bbox="565 184 602 1383"> <p><i>Fischer</i></p> </div> <div data-bbox="602 184 993 1383"> <p><i>Fischer</i> discloses 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.</p> <p>For example, Figure 2 in <i>Fischer</i> discloses the digital signature (i.e., source of code) includes an associated public key:</p> </div>

U.S. Patent No. 6,125,447 – Claim 8	<div data-bbox="196 730 228 842" data-label="Text">Fischer</div> <div data-bbox="293 890 318 1171" data-label="Text">AUTHORIZATION SIGNATURE:</div> <div data-bbox="331 422 837 1146" data-label="Diagram"> </div>
	<p>“The authorization signature includes a signature segment 40. The signature segment 40 may include a reference to the signer’s certificate, i.e., an identifier for identifying the signer’s certificate. In accordance with a preferred embodiment of the present invention, such a digital certificate is a digital message created by a trusted entity <i>which contains the user’s public key</i> and the name of the user (which is accurate to the entity’s satisfaction) and possibly a representation of the authority which has been granted to the user by the party who signs the digital message.”</p> <p><i>Fischer</i>, 6:25-35 (emphasis added) & Fig. 2 (excerpted). Thus, <i>Fischer</i> discloses that a user’s public key may be associated with the digital signature (i.e., a source of code).</p>

U.S. Patent No. 6,125,447 – Claim 9	<div data-bbox="1266 730 1299 842" data-label="Text">Fischer</div> <p><i>Fischer</i> discloses establishing an association between one or more protection domains and one or more sources of code and one or more keys associated with one or more sources of code based on data persistently stored.</p>
-------------------------------------	---

U.S. Patent No. 6,125,447 – Claim 9	Fischer
<p>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,</p>	<p>For example, the ‘447 Patent discloses persistently stored data such as instructions stored in a file, mappings stored in a database system, and mapping attributes of a persistent object:</p> <p>“Storing instructions in a file is just one method of representing the policy of the system with persistently stored data. Other methods are possible for representing the policy with persistent data. For example, data in a database system can be used to map code identifiers to authorized permissions, or attributes of a persistent object can be used to map code identifiers to authorized permissions.”</p> <p>‘447 Patent, 9:19-25.</p> <p>Similar to this disclosure of the “persistently stored” data in the ‘447 Patent, <i>Fischer</i> discloses that the association between the protection domains and the digital signatures (i.e., sources of code) and their associated keys are based on data that is persistently stored as an attribute of a persistent object:</p> <p>“FIG. 3C shows an important application in which a PAI data structure is associated with a program according to an embodiment of the present invention. FIG. 3C shows an illustrative data structure for a secure exchangeable ‘object’. The data structure may be signed by a trusted authority. The signing of such a data structure allows the object to be securely transmitted from user to user. Although the data structure shown in FIG. 3 is set forth in a general format, it may be structured as set forth in the inventor’s copending application filed on Apr. 6, 1992 and entitled ‘Method and Apparatus for Creating, Supporting and Processing a Travelling Program’ (U.S. Ser. No. 07/863,552.), which application is hereby expressly incorporated herein by reference.</p> <p>... The program authorization information is embedded in a segment 116 which specifies the authorization for the object’s program or programs in a manner to be described more fully hereinafter.”</p> <p><i>Fischer</i>, 7:49-8:2.</p>

Figure 3C in *Fischer* shows these attributes of a persistent object:



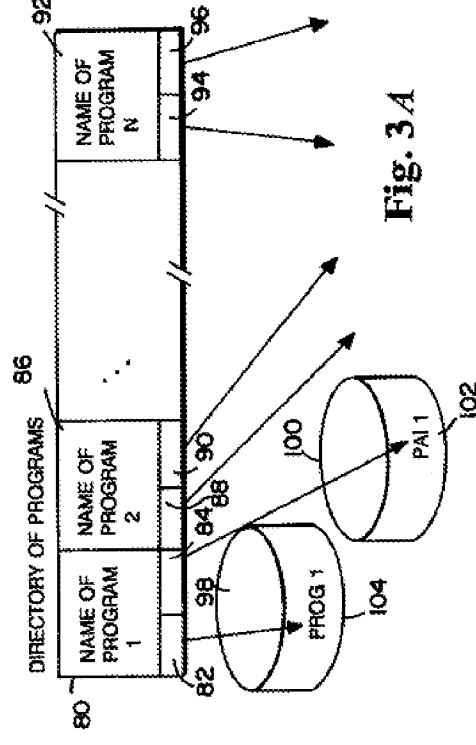
Fischer, FIG. 3C (see, e.g., element 116).

In addition to its disclosure of persistent data stored as an attribute of a persistent object, *Fischer* discloses more generally associating the PAI information (i.e., protection domains) based on other types of persistently stored data. For example, *Fischer* discloses storing PAI information (which includes the digital signature/keys) on a separate/remote storage device or in the same memory as the program:

“FIG. 3A shows an exemplary schematic representation of a system’s directory of programs. . . .

. . . Additionally, associated with each of the program related identifiers is an indicator 84, 90, . . . 96, respectively, which identifies the location of its associated program authorization information, e.g., PAI 1. Although the program authorization information,

PAI 1, is depicted as being stored in a separate memory device 100, it may, if desired, be stored in the same memory media as its associated program.”



Fischer, 7:20-35 & Fig. 3A.

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

As discussed above, *Fischer* discloses that the persistently stored PAI segment of the Figure 3C object/class associates a digital signature (i.e., a source of code) and the associated public keys with a set of one or more permissions:

“The present method an apparatus utilizes a unique operation system design that includes a system monitor which limits the ability of a program about to be executed to the use of predefined resources (e.g., data files, disk writing capabilities etc.). The system monitor builds a data structure including a set of authorities defining that which a program is permitted to do and/or that which the program is precluded from doing.

The set of authorities and/or restrictions assigned to a program to be executed are referred to herein as ‘program authorization information’ (or ‘PAI’). Once defined, the program authorization information is thereafter associated with each program to be executed to thereby delineate the types of resources and functions that the program is allowed to

U.S. Patent No. 6,125,447 – Claim 9	Fischer
	<p>utilize.”</p> <p><i>Fischer</i>, 2:16-30.</p> <p>“The PAI defines the range of operations that a program may execute and/or defines those operations that a program cannot perform. The program is permitted to access what has been authorized and nothing else. In this fashion, the program may be regarded as being placed in a program capability limiting ‘safety box.’ This ‘safety box’ is thereafter associated with the program such that whenever the system monitor runs the program, the PAI for that program is likewise loaded and monitored. When the program is to perform a function or access a resource, the associated PAI is monitored to confirm that the operation is within the defined program limits. If the program attempts to do anything outside the authorized limits, then the program execution is halted.”</p> <p><i>Fischer</i>, 2:34-48.</p> <p>“Even programs with no known trustworthiness can be used after program authorization information associates a wide range of restrictions to thereby allow potentially beneficial programs to be safely used—even if they do not have an official certification of trust.</p> <p>The present invention also allows an unlimited number of different resources and functions to be controlled. For example, some useful resources/functions which may be controlled include: the ability to limit a program to certain files or data sets; the ability to transmit data via electronic mail to someone outside the user’s domain; the ability of a program to create or solicit digital signatures; the ability to limit access to a program of certain security classes, etc.”</p> <p><i>Fischer</i>, 3:48-61.</p> <p>“Additionally, in block 340, an examination is made of the PAI information stored in the process control block. As a follow up to, or associated with, the processing in block 340, a check is made in block 342 to determine whether the examined PAI is allowed access to</p>

U.S. Patent No. 6,125,447 – Claim 9	<i>Fischer</i>
	<p>the required resources or allowed to perform the required functions. For example, if an attempt is made to use electronic mail, a check is made of the PAI to determine whether the program is authorized to perform electronic mail functions and if so whether the mailing is limited to a set of mail identifiers.</p> <p>If the check at 342 reveals that the PAI does not allow the attempted function or resource access, then a error message is generated in block 344 to indicated that the program is attempting to exceed its limits, access to the resource or function is denied and an appropriate error code or message is generated. . . .</p> <p>If the check in block 342 reveals that the PAI does allow access to the function or resource, then a check is made in block 346 to apply conventional access controls to ensure that the user of the program is still within the bounds of his authority.”</p> <p><i>Fischer</i>, 19:16-33, 19:51-55.</p>

NOTE: Claims 10-18 are exact replicas of claims 1-9, except that claims 10-18 are written as apparatus claims (instructions embodied in a computer readable medium), whereas claims 1-9 are written as method claims.

U.S. Patent No. 6,125,447 – Claim 10	<i>Fischer</i>
<p>10. 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:</p>	<p><i>Fischer</i> discloses a computer-readable medium carrying one or more sequences of one or more instructions . . . which . . . are executed by one or more processors.</p> <p>For example, <i>Fischer</i> discloses a system including IBM PC computers having processors, a memory (i.e., computer-readable medium), and a program (i.e., one or more sequences of instructions) stored in the memory:</p> <p>“FIG. 1 shows in block diagram form an exemplary communications system which may be used in conjunction with the present invention. . . . Terminals, A, B . . . N may, by way of example only, be IBM PC’s having a processor (with main memory) 2 which is coupled to a conventional keyboard/CRT display 4. Additionally, each processor is</p>

preferably coupled to a non-volatile program and program authorization information (PAI) storage 7 which may be a disk memory device.”

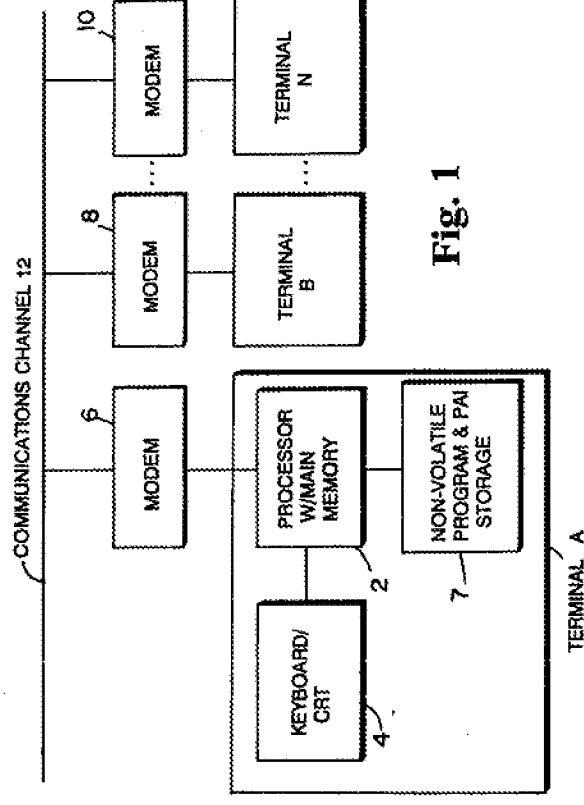


Fig. 1

Fischer, 4:45-58 & Fig. 1.

“Turning back to FIG. 1, a program of unknown trust may be injected into the system via communications channel 12 or from a floppy disk loaded into terminal A. The program may be initially stored in, for example, the user’s program disk memory 7.”

Fischer, 9:64-68.

Fischer discloses the instructions causing the program to establish one or more protection domains, wherein a protection domain is associated with zero or more permissions.

For example, *Fischer* discloses that the program implements a system monitor that builds (i.e., establishes) a Program Authorization Information (“PAI”) data structure as a protection domain:

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

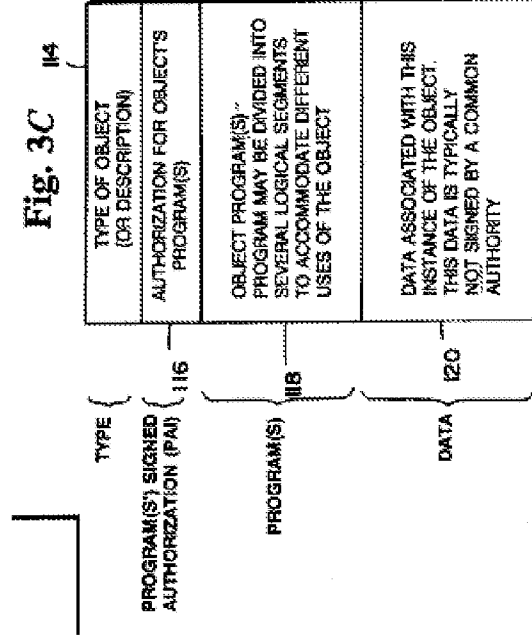
U.S. Patent No. 6,125,447 – Claim 10	Fischer
	<p>“The present method an apparatus utilizes a unique operation system design that includes a system monitor which limits the ability of a program about to be executed to the use of predefined resources (e.g., data files, disk writing capabilities etc.). The system monitor builds a data structure including a set of authorities defining that which a program is permitted to do and/or that which the program is precluded from doing.</p> <p>The set of authorities and/or restrictions assigned to a program to be executed are referred to herein as ‘program authorization information’ (or ‘PAI’). Once defined, the program authorization information is thereafter associated with each program to be executed to thereby delineate the types of resources and functions that the program is allowed to utilize.”</p> <p><i>Fischer</i>, 2:16-30.</p> <p><i>Fischer</i> further discloses that PAI information for a program may be combined, as appropriate, with the PAI associated with a calling program. :</p> <p>“Thereafter, the program X’s program authorizing information is combined, as appropriate, with the PAI associated with the PCB of the calling program, if any. This combined PAI, which may include multiple PAI’s, is then stored in an area of storage which cannot generally be modified by the program and the address of the PAI is stored in the process control block (PCB) as indicated in field 156 of FIG. 5. Thus, if program X is called by a calling program, it is subject to all its own constraints as well as being combined in some way with the constraints of the calling program, which aggregate constraints are embodied into program X’s PAI. In this fashion, a calling program may not be permitted to exceed its assigned bounds by merely calling another program.”</p> <p><i>Fischer</i>, 19:40-54.</p> <p><i>Fischer</i> further discloses that the PAI is associated with zero or more permissions, such as a range of operations that a program may execute or may be precluded from executing:</p>

U.S. Patent No. 6,125,447 – Claim 10	Fischer
	<p>“The PAI defines the range of operations that a program may execute and/or defines those operations that a program cannot perform. The program is permitted to access what has been authorized and nothing else. In this fashion, the program may be regarded as being placed in a program capability limiting ‘safety box.’ This ‘safety box’ is thereafter associated with the program such that whenever the system monitor runs the program, the PAI for that program is likewise loaded and monitored. When the program is to perform a function or access a resource, the associated PAI is monitored to confirm that the operation is within the defined program limits. If the program attempts to do anything outside the authorized limits, then the program execution is halted.”</p> <p><i>Fischer</i>, 2:34-48. Indeed, <i>Fischer</i> discloses a PAI associated with zero permissions (e.g., “no known trustworthiness” that leads to “a wide range of restrictions”), and a PAI associated with more permissions (e.g., “an unlimited number of different resources and functions to be controlled”):</p> <p>“Even programs with no known trustworthiness can be used after program authorization information associates a wide range of restrictions to thereby allow potentially beneficial programs to be safely used—even if they do not have an official certification of trust.</p> <p>The present invention also allows an unlimited number of different resources and functions to be controlled. For example, some useful resources/functions which may be controlled include: the ability to limit a program to certain files or data sets; the ability to transmit data via electronic mail to someone outside the user’s domain; the ability of a program to create or solicit digital signatures; the ability to limit access to a program of certain security classes, etc.”</p> <p><i>Fischer</i>, 3:48-61.</p>
establishing an association between said one or more protection domains and one or more classes of one or more objects;	<i>Fischer</i> discloses the instructions causing the program to establish an association between one or more protection domains and one or more classes of one or more objects.

U.S. Patent No. 6,125,447 – Claim 10	Fischer
and	<p>For example, <i>Fischer</i> discloses that a PAI (i.e., a protection domain) is associated with a program:</p> <p>“Once defined, the program authorization information [(PAI)] is thereafter associated with each program to be executed to thereby delineate the types of resources and functions that the program is allowed to utilize. The PAI associated with a particular program may be assigned by a computer system owner/user or by someone who the computer system owner/user implicitly trusts.”</p> <p><i>Fischer</i>, 2:26-33.</p> <p><i>Fischer</i> further discloses that a program with which a PAI is associated may be part of an object:</p> <p>“The present invention is directed to providing reliable security, even when operating with complex data structures, e.g., objects, containing their own program instructions, which are transmitted among users.”</p> <p><i>Fischer</i>, 2:6-9.</p> <p>“Through the use of the present invention, general object oriented data may be transferred from user to user without exposing users to the potential dangers of viruses or mischievous users.”</p> <p><i>Fischer</i>, 4:10-13.</p> <p>“In one contemplated embodiment of the present invention, programs may be part of data objects, which are written in a high-level control language and are executed by a standardized interpreter program which executes this high-level language. In this case, part of the interpreter’s task is to verify that the functions encountered in the high level logic are, in fact, permissible. If such tasks are not permissible, the interpreter then suppresses the execution of the program not authorized to perform such tasks.”</p>

U.S. Patent No. 6,125,447 – Claim 10	<i>Fischer</i>
	<p><i>Fischer</i>, 3:11-20.</p> <p>“In accordance with the present invention, a PAI is associated with programs to be executed. FIGS. 3A through 3D depict four exemplary approaches for associating program authorization information with a program. . . .</p> <p>...</p> <p>FIG. 3C shows an important application in which a PAI data structure is associated with a program according to an embodiment of the present invention. FIG. 3C shows an illustrative data structure for a secure exchangeable ‘object’. The data structure may be signed by a trusted authority. The signing of such a data structure allows the object to be securely transmitted from user to user. Although the data structure shown in FIG. 3 is set forth in a general format, it may be structured as set forth in the inventor’s copending application filed on Apr. 6, 1992 and entitled ‘Method and Apparatus for Creating, Supporting and Processing a Travelling Program’ (U.S. Ser. No. 07/863,552.), which application is hereby expressly incorporated herein by reference.</p> <p>... The program authorization information is embedded in a segment 116 which specifies the authorization for the object’s program or programs in a manner to be described more fully hereinafter.”</p> <p><i>Fischer</i>, 7:14-18, 7:49-8:2.</p> <p>“Thereafter, the PAI is stored using, for example, one of the approaches set forth in FIGS. 3A through 3D so that it is associated with its program 272”</p> <p><i>Fischer</i>, 15:24-26. As just described, FIG. 3C discloses the program as part of an object/class. See <i>Fischer</i>, 7:49-8:2.</p> <p>To the extent <i>Fischer</i> does not expressly disclose “classes” of the objects, one of ordinary</p>

skill in the art would understand that a class, as that term is used in the '447 Patent, is a necessarily present feature of the objects disclosed in *Fischer*. To be sure, Figure 3C in *Fischer* shows an object-oriented data structure including a type segment (114), program (e.g., method) segment (118), and a data segment (120), necessarily implicating an object-oriented program architecture.



Fischer, FIG. 3C.

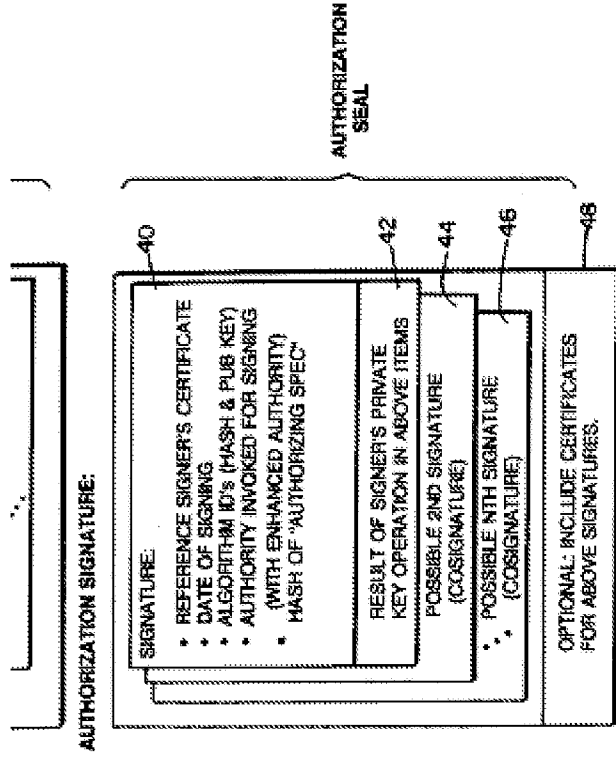
The '447 Patent discloses that a class is a high-level abstraction or definition of an object, such that "[e]ach object belonging to a class has the same fields ('attributes') and the same methods." '447 Patent, 7:4-5; *see also*, '447 Patent, 6:63-7:25.

Thus, *Fischer*'s disclosure of a data object in Figure 3C that includes fields/attributes (labeled "DATA ASSOCIATED WITH THIS INSTANCE OF THE OBJECT") as well as methods (labeled "OBJECT PROGRAM(S)") is consistent with the the '447 Patent's description of a class. In addition, Figure 3C of *Fischer* mentions that the depicted object is an "instance," which further shows that *Fischer*'s disclosure includes object-oriented data structures, which

U.S. Patent No. 6,125,447 – Claim 10	Fischer
	<p>are necessarily part of a class, as that term is used in the '447 Patent. <i>See</i> '447 Patent, 7:7-8 ("An object is said to be an 'instance' of the class to which the object belongs.").</p> <p>Additionally, the '447 Patent admits that the ideas of classes and object instances were well known to those skilled in the art: "[C]lass definitions are generated from source code written by a programmer. For example, a programmer using a Java Development Kit enters source code that conforms to the Java programming language into a source file. The source code embodies class definitions and other instructions which are used to generate byte code which controls the execution of a code executor (i.e. a Java virtual machine). <i>Techniques for defining classes and generating code executed by a code executor, such as a Java virtual machine, are well known to those skilled in the art.</i>" '447 Patent, 7:15-24.</p> <p><i>Fischer's</i> disclosure is entirely consistent with this admission, as <i>Fischer</i> discloses that "programs may be part of data objects, which are written in a high-level control language and are executed by a standardized interpreter program which executes the high-level language." <i>Fischer</i>, 3:12-15. In light of these disclosures in the '447 Patent and <i>Fischer</i>, there can be no doubt that one of ordinary skill in the art would view the object disclosed in <i>Fischer</i> as an instance of a class, such that the class, if not expressly disclosed, is necessarily present in the <i>Fischer</i> disclosure.</p>
<p>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.</p>	<p><i>Fischer</i> discloses the instructions causing the program 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.</p> <p>For example,</p> <p>“FIGS. 10 and 11 illustrate the sequence of operations of a supervisor program for controlling the processing of a program being executed in accordance with its program authorization information.”</p> <p><i>Fischer</i>, 15:56-59.</p> <p>“Depending on the processing in block 316 [of FIG. 10], a decision is made in block 322</p>

U.S. Patent No. 6,125,447 – Claim 10	<i>Fischer</i> whether the signatures are valid, authorized and trusted. If the signatures are not determined to be valid, then the routing branches to block 324 where the execution in program X is suppressed.” <i>Fischer</i> , 16:66-17:3. “If the processing in blocks 322 and 316 reveal that the signatures are valid, then the processing in block 326 is performed.” <i>Fischer</i> , 17:31-33.
U.S. Patent No. 6,125,447 – Claim 11	<i>Fischer</i> <i>Fischer</i> discloses the computer readable medium of claim 10. See claim chart above for further details. <i>Fischer</i> discloses at least one protection domain of said one or more protection domains is associated with a code identifier. For example, the ‘447 Patent discloses a code identifier as “describing the source of code that defines a class, a set of public cryptographic keys associated with the source of code, or other information which describes the source of code, or any combination thereof. A ‘source of code’ is an entity from which computer instructions are received. Examples 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 Patent, 3:13-21. Figure 3 of the ‘447 Patent discloses a policy file that includes a URL (i.e., file://somesource) and a key name (i.e., “somekey”), and describes both as code identifiers. ‘447 Patent, 9:26-37 & Fig. 3. Similar to this disclosure of the “code identifier” in the ‘447 Patent, Figure 2 in <i>Fischer</i> discloses the PAI data structure (i.e., protection domain data structure), which explicitly associates the protection domain with a “source of code” such as the signer of a digital

certificate:



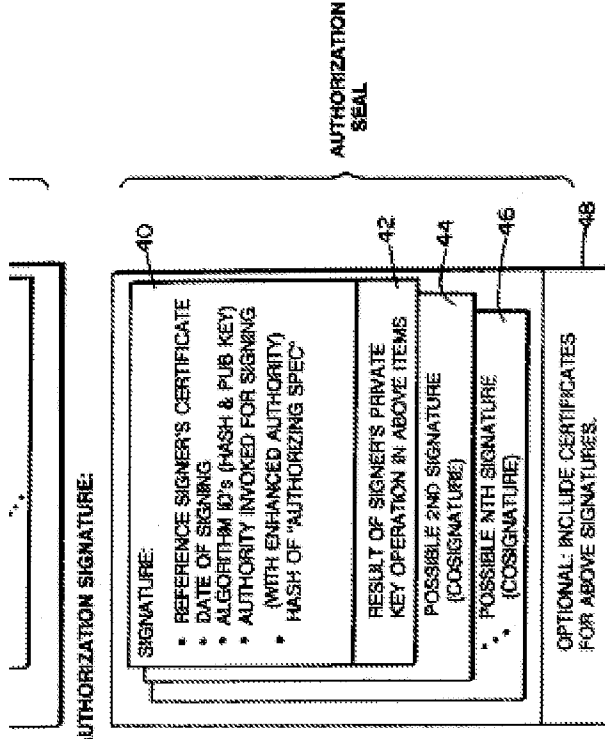
“The authorization signature includes a signature segment 40. The signature segment 40 may include a reference to the signer’s certificate, i.e., an identifier for identifying the signer’s certificate. In accordance with a preferred embodiment of the present invention, such a digital certificate is a digital message created by a trusted entity which contains the user’s public key and the name of the user (which is accurate to the entity’s satisfaction) and possibly a representation of the authority which has been granted to the user by the party who signs the digital message.”

Fischer, 6:25-35 (emphasis added) & Fig. 2 (excerpted).

Fischer expressly discloses that the digital signature (i.e., code identifier) may be associated with a manufacturer of the program (i.e., a source of code or “an entity from which computer instructions are received”):

U.S. Patent No. 6,125,447 – Claim 11	Fischer
	<p>“The present invention <i>allows PAI information to be associated in any appropriate manner</i>, so that in principle a user could define one or more levels of PAI which are then combined together with perhaps a more universal PAI, <i>or with a PAI which was signed and supplied by the or [sic] manufacturer of this program.</i>”</p> <p><i>Fischer</i>, 9:3-8 (emphasis added).</p> <p>“FIGS. 6 through 9 is a flowchart illustrating an exemplary sequence of operations of a utility program for establishing program authorization information. Such a utility program prompts a user, i.e., the end user, the user’s agent, <i>or even the manufacturer</i>, to define a range of authorities which are associated with a program to be executed by the user’s system.”</p> <p><i>Fischer</i>, 11:7-13 (emphasis added).</p> <p>“If no PAI has yet been associated with the program, then a check is made to determine whether the program has an associated signed ‘pedigree’ from the manufacturer (306). Thus, if a well known manufacturer of programs has signed the program with a public key or digital certificate, then, if desired, such a program may be assigned whatever level of authority desired depending upon how much the manufacturer is trusted and the system may permit execution of such program. Such a digital signature <i>from the manufacturer can be used to verify that the associated program had not been infected with a virus since it can be determined whether or not the program is exactly the same as it was when it was generated by the manufacturer.</i>”</p> <p><i>Fischer</i>, 16:12-25 (emphasis added).</p>
at least one class of said one or more classes is associated with said code identifier; and	<i>Fischer</i> discloses at least one class of said one or more classes is associated with said code identifier. For example, as discussed above, Figures 2 and 3C of <i>Fischer</i> disclose that the PAI data structure may contain a manufacturer’s signature (i.e., code identifier) (see Fig. 2), and that the PAI with the code identifier is associated with the object/class data structure

U.S. Patent No. 6,125,447 – Claim 11	<p style="text-align: center;">Fischer</p>
	<p>because it is expressly included as part of the object/class data structure.</p> <div style="text-align: center;"> </div> <p style="text-align: center;">Fischer, Fig. 3C (“PROGRAM(S) SIGNED AUTHORIZATION (PAI)” included as element 116 of the disclosed object/class data structure).</p>
<p>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.</p>	<p><i>Fischer</i> discloses associating said one or more protection domains and said one or more classes based on said code identifier. For example, <i>Fischer</i> discloses that authorization (e.g., protection domains) may be associated with a program based on the digital signature (i.e., code identifier) included in an object/class:</p> <p style="padding-left: 40px;">“Thus, if a well known manufacturer of programs has signed the program with a public key or digital certificate, then, if desired, <i>such a program may be assigned whatever level of authority desired depending upon how much the manufacturer is trusted</i> and the system may permit execution of such program.”</p> <p style="text-align: right;"><i>Fischer</i>, 16:15-20 (emphasis added).</p>
U.S. Patent No. 6,125,447 – Claim 12	<p style="text-align: center;">Fischer</p>

<p>U.S. Patent No. 6,125,447 – Claim 12</p> <p>12. 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.</p>	<p>U.S. Patent No. 6,125,447 – Claim 12</p> <p><i>Fischer</i></p> <p><i>Fischer</i> discloses the code identifier indicates a source of code used to define each class of said one or more classes. For example, <i>Fischer</i> indicates that each object/class may include the digital signature (i.e., a code identifier):</p> <p>“FIG. 3C shows an illustrative data structure for a secure exchangeable ‘object’. The data structure may be signed by a trusted authority. The signing of such a data structure allows the object to be securely transmitted from user to user.”</p> <p><i>Fischer</i>, 7:51-56.</p>
<p>U.S. Patent No. 6,125,447 – Claim 13</p> <p>13. The computer readable medium of claim 11, wherein said code identifier indicates a key associated with each class of said one or more classes.</p>	<p>U.S. Patent No. 6,125,447 – Claim 13</p> <p><i>Fischer</i></p> <p><i>Fischer</i> discloses the code identifier indicates a key associated with each class of said one or more classes. For example, Figure 2 in <i>Fischer</i> discloses the digital certificate (i.e., code identifier) includes a public key that may be associated with the object/class:</p> 

U.S. Patent No. 6,125,447 – Claim 13	<p><i>Fischer</i></p> <p>“The authorization signature includes a signature segment 40. The signature segment 40 may include a reference to the signer’s certificate, i.e., an identifier for identifying the signer’s certificate. In accordance with a preferred embodiment of the present invention, such a digital certificate is a digital message created by a trusted entity <i>which contains the user’s public key</i> and the name of the user (which is accurate to the entity’s satisfaction) and possibly a representation of the authority which has been granted to the user by the party who signs the digital message.”</p> <p><i>Fischer</i>, 6:25-35 (emphasis added) & Fig. 2 (excerpted).</p>
U.S. Patent No. 6,125,447 – Claim 14	<p><i>Fischer</i></p> <p><i>Fischer</i> discloses the code identifier indicates a source of code used to define each class of said one or more classes.</p> <p>For example, the ‘447 Patent discloses the “source of code” of a code identifier as “an entity from which computer instructions are received. Examples 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 Patent, 3:15-21.</p> <p><i>Fischer</i> expressly discloses that the digital signature (i.e., code identifier) may be associated with a manufacturer of the program (i.e., a source of code or “an entity from which computer instructions are received”):</p> <p>“The present invention <i>allows PAI information to be associated in any appropriate manner</i>, so that in principle a user could define one or more levels of PAI which are then combined together with perhaps a more universal PAI, <i>or with a PAI which was signed and supplied by the or [sic] manufacturer of this program.</i>”</p> <p><i>Fischer</i>, 9:3-8 (emphasis added).</p> <p>“FIGS. 6 through 9 is a flowchart illustrating an exemplary sequence of operations of a</p>

U.S. Patent No. 6,125,447 – Claim 14	Fischer
	<p>utility program for establishing program authorization information. Such a utility program prompts a user, i.e., the end user, the user's agent, <i>or even the manufacturer</i>, to define a range of authorities which are associated with a program to be executed by the user's system."</p> <p><i>Fischer</i>, 11:7-13 (emphasis added).</p> <p>"If no PAI has yet been associated with the program, then a check is made to determine whether the program has an associated signed 'pedigree' from the manufacturer (306). Thus, if a well known manufacturer of programs has signed the program with a public key or digital certificate, then, if desired, such a program may be assigned whatever level of authority desired depending upon how much the manufacturer is trusted and the system may permit execution of such program. Such a digital signature <i>from the manufacturer can be used to verify that the associated program had not been infected with a virus since it can be determined whether or not the program is exactly the same as it was when it was generated by the manufacturer.</i>"</p> <p><i>Fischer</i>, 16:12-25 (emphasis added).</p>
<p>... [wherein said code identifier] indicates a key associated with each class of said one or more classes.</p>	<p><i>Fischer</i> discloses the code identifier indicates a key associated with each class of said one or more classes. For example Figure 2 in <i>Fischer</i> discloses the digital certificate (i.e., code identifier) includes a public key may be associated with the object/class:</p>

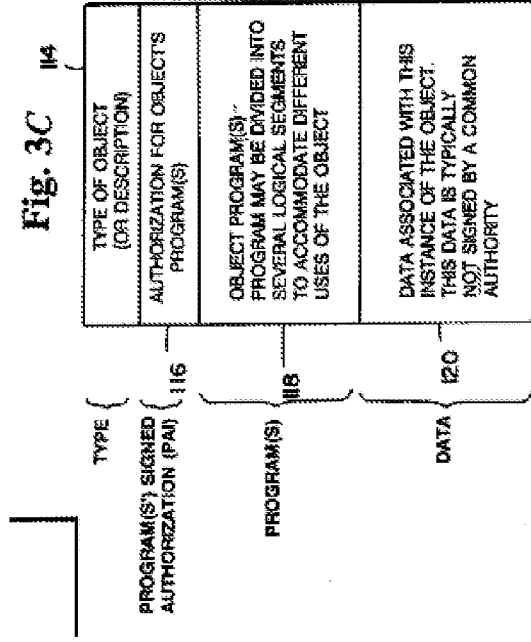
U.S. Patent No. 6,125,447 – Claim 14	<div data-bbox="191 184 837 1383"> <div data-bbox="191 184 228 1383">Fischer</div> <div data-bbox="228 184 837 1383"> <div data-bbox="228 184 285 1383">AUTHORIZATION SIGNATURE:</div> <div data-bbox="285 184 837 1383"> <div data-bbox="285 184 326 1383">SIGNATURE:</div> <div data-bbox="326 184 537 1383"> <ul style="list-style-type: none"> • REFERENCE SIGNER'S CERTIFICATE • DATE OF SIGNING • ALGORITHM ID's (HASH & PUB KEY) • AUTHORITY INVOKED FOR SIGNING (WITH ENHANCED AUTHORITY) • HASH OF "AUTHORIZING SPEC" </div> <div data-bbox="537 184 651 1383"> <div data-bbox="537 184 578 1383">RESULT OF SIGNER'S PRIVATE KEY OPERATION ON ABOVE ITEMS</div> <div data-bbox="578 184 651 1383">POSSIBLE 2ND SIGNATURE (COSIGNATURE)</div> </div> <div data-bbox="651 184 732 1383"> <div data-bbox="651 184 691 1383">POSSIBLE NTH SIGNATURE (COSIGNATURE)</div> <div data-bbox="691 184 732 1383">OPTIONAL: INCLUDE CERTIFICATES FOR ABOVE SIGNATURES</div> </div> </div> </div> <div data-bbox="537 422 578 579">AUTHORIZATION SEAL</div> </div>
--------------------------------------	---

“The authorization signature includes a signature segment 40. The signature segment 40 may include a reference to the signer’s certificate, i.e., an identifier for identifying the signer’s certificate. In accordance with a preferred embodiment of the present invention, such a digital certificate is a digital message created by a trusted entity *which contains the user’s public key* and the name of the user (which is accurate to the entity’s satisfaction) and possibly a representation of the authority which has been granted to the user by the party who signs the digital message.”

Fischer, 6:25-35 (emphasis added) & Fig. 2 (excerpted).

Fischer	<p><i>Fischer</i> discloses 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.</p>
---------	--

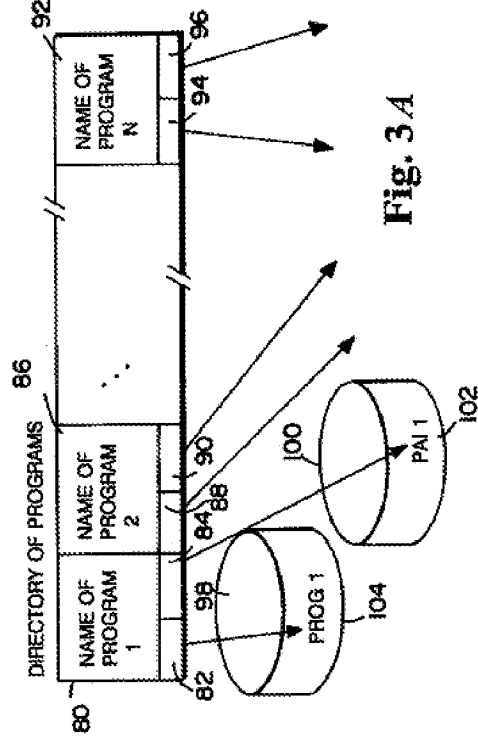
U.S. Patent No. 6,125,447 – Claim 15	<i>Fischer</i>
<p>based on said code identifier further includes associating said one or more protection domains and said one or more classes based on data persistently stored,</p>	<p>For example, the ‘447 Patent discloses persistently stored data such as instructions stored in a file, mappings stored in a database system, and mapping attributes of a persistent object:</p> <p>“Storing instructions in a file is just one method of representing the policy of the system with persistently stored data. Other methods are possible for representing the policy with persistent data. For example, data in a database system can be used to map code identifiers to authorized permissions, or attributes of a persistent object can be used to map code identifiers to authorized permissions.”</p> <p>‘447 Patent, 9:19-25.</p> <p>Similar to this disclosure of the “persistently stored” data in the ‘447 Patent, <i>Fischer</i> discloses that the association between the protection domains and the one or more classes is based on data that is persistently stored as an attribute of a persistent object:</p> <p>“FIG. 3C shows an important application in which a PAI data structure is associated with a program according to an embodiment of the present invention. FIG. 3C shows an illustrative data structure for a secure exchangeable ‘object’. The data structure may be signed by a trusted authority. The signing of such a data structure allows the object to be securely transmitted from user to user. Although the data structure shown in FIG. 3 is set forth in a general format, it may be structured as set forth in the inventor’s copending application filed on Apr. 6, 1992 and entitled ‘Method and Apparatus for Creating, Supporting and Processing a Travelling Program’ (U.S. Ser. No. 07/863,552.), which application is hereby expressly incorporated herein by reference.</p> <p>... The program authorization information is embedded in a segment 116 which specifies the authorization for the object’s program or programs in a manner to be described more fully hereinafter.”</p> <p><i>Fischer</i>, 7:49-8:2.</p> <p>Figure 3C in <i>Fischer</i> shows these attributes of a persistent object:</p>



Fischer, FIG. 3C (see, e.g., element 116).

In addition to its disclosure of persistent data stored as an attribute of a persistent object, *Fischer* discloses more generally associating the PAI information (i.e., protection domains) based on other types of persistently stored data. For example, *Fischer* discloses storing PAI information on a separate/remote storage device or in the same memory as the program:

“FIG. 3A shows an exemplary schematic representation of a system’s directory of programs. . . .
 . . . Additionally, associated with each of the program related identifiers is an indicator 84, 90, . . . 96, respectively, which identifies the location of its associated program authorization information, e.g., PAI 1. Although the program authorization information, PAI 1, is depicted as being stored in a separate memory device 100, it may, if desired, be stored in the same memory media as its associated program.”



Fischer, 7:20-35 & Fig. 3A.

wherein said data associates code identifiers with a set of one or more permissions.

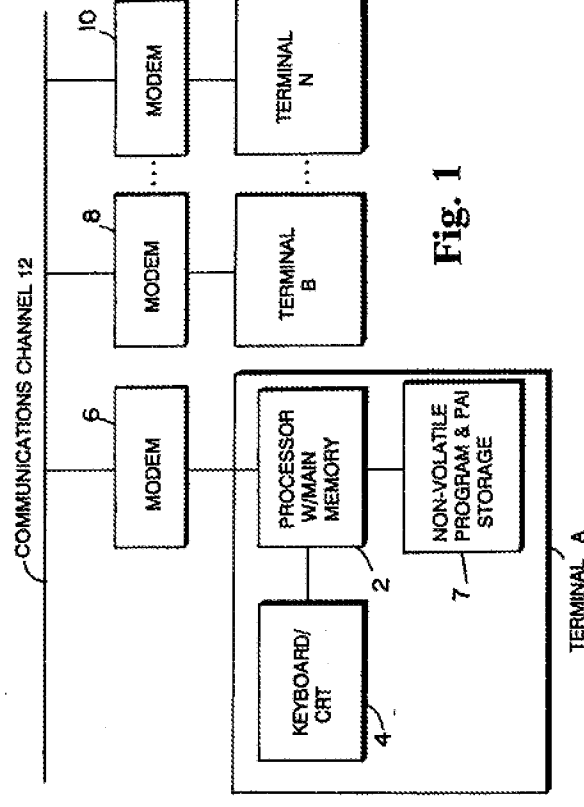
Fischer discloses that the persistently stored data associates code identifiers with a set of one or more permissions. As discussed above, *Fischer* discloses that the persistently stored PAI segment of the Figure 3C object/class associates code identifiers (e.g., a digital signature) with a set of one or more permissions:

“The present method an apparatus utilizes a unique operation system design that includes a system monitor which limits the ability of a program about to be executed to the use of predefined resources (e.g., data files, disk writing capabilities etc.). The system monitor builds a data structure including a set of authorities defining that which a program is permitted to do and/or that which the program is precluded from doing.

The set of authorities and/or restrictions assigned to a program to be executed are referred to herein as ‘program authorization information’ (or ‘PAI’). Once defined, the program authorization information is thereafter associated with each program to be executed to thereby delineate the types of resources and functions that the program is allowed to utilize.”

U.S. Patent No. 6,125,447 – Claim 15	Fischer
	<p><i>Fischer</i>, 2:16-30.</p> <p>“The PAI defines the range of operations that a program may execute and/or defines those operations that a program cannot perform. The program is permitted to access what has been authorized and nothing else. In this fashion, the program may be regarded as being placed in a program capability limiting ‘safety box.’ This ‘safety box’ is thereafter associated with the program such that whenever the system monitor runs the program, the PAI for that program is likewise loaded and monitored. When the program is to perform a function or access a resource, the associated PAI is monitored to confirm that the operation is within the defined program limits. If the program attempts to do anything outside the authorized limits, then the program execution is halted.”</p> <p><i>Fischer</i>, 2:34-48.</p> <p>“Even programs with no known trustworthiness can be used after program authorization information associates a wide range of restrictions to thereby allow potentially beneficial programs to be safely used—even if they do not have an official certification of trust.</p> <p>The present invention also allows an unlimited number of different resources and functions to be controlled. For example, some useful resources/functions which may be controlled include: the ability to limit a program to certain files or data sets; the ability to transmit data via electronic mail to someone outside the user’s domain; the ability of a program to create or solicit digital signatures; the ability to limit access to a program of certain security classes, etc.”</p> <p><i>Fischer</i>, 3:48-61.</p> <p>“Additionally, in block 340, an examination is made of the PAI information stored in the process control block. As a follow up to, or associated with, the processing in block 340, a check is made in block 342 to determine whether the examined PAI is allowed access to the required resources or allowed to perform the required functions. For example, if an</p>

U.S. Patent No. 6,125,447 – Claim 15	<p data-bbox="185 730 228 840"><i>Fischer</i></p> <p data-bbox="228 218 342 1346">attempt is made to use electronic mail, a check is made of the PAI to determine whether the program is authorized to perform electronic mail functions and if so whether the mailing is limited to a set of mail identifiers.</p> <p data-bbox="375 218 521 1346">If the check at 342 reveals that the PAI does not allow the attempted function or resource access, then a error message is generated in block 344 to indicated that the program is attempting to exceed its limits, access to the resource or function is denied and an appropriate error code or message is generated. . . .</p> <p data-bbox="553 275 667 1346">If the check in block 342 reveals that the PAI does allow access to the function or resource, then a check is made in block 346 to apply conventional access controls to ensure that the user of the program is still within the bounds of his authority.”</p> <p data-bbox="699 1010 743 1381"><i>Fischer</i>, 19:16-33, 19:51-55.</p>
U.S. Patent No. 6,125,447 – Claim 16	<p data-bbox="813 730 857 840"><i>Fischer</i></p> <p data-bbox="857 239 927 1381"><i>Fischer</i> discloses a computer-readable medium carrying one or more sequences of one or more instructions . . . which . . . are executed by one or more processors.</p> <p data-bbox="959 218 1073 1381">For example, <i>Fischer</i> discloses a system including IBM PC computers having processors, a memory (i.e., computer-readable medium), and a program (i.e., one or more sequences of instructions) stored in the memory:</p> <p data-bbox="1105 218 1331 1346">“FIG. 1 shows in block diagram form an exemplary communications system which may be used in conjunction with the present invention. . . . Terminals, A, B . . . N may, by way of example only, be IBM PC’s having a processor (with main memory) 2 which is coupled to a conventional keyboard/CRT display 4. Additionally, each processor is preferably coupled to a non-volatile program and program authorization information (PAI) storage 7 which may be a disk memory device.”</p>



Fischer, 4:45-58 & Fig. 1.

“Turning back to FIG. 1, a program of unknown trust may be injected into the system via communications channel 12 or from a floppy disk loaded into terminal A. The program may be initially stored in, for example, the user’s program disk memory 7.”

Fischer, 9:64-68..

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

Fischer discloses establishing one or more protection domains, wherein a protection domain is associated with zero or more permissions.

For example, *Fischer* discloses that the system monitor builds (i.e., establishes) a Program Authorization Information (“PAI”) data structure as a protection domain:

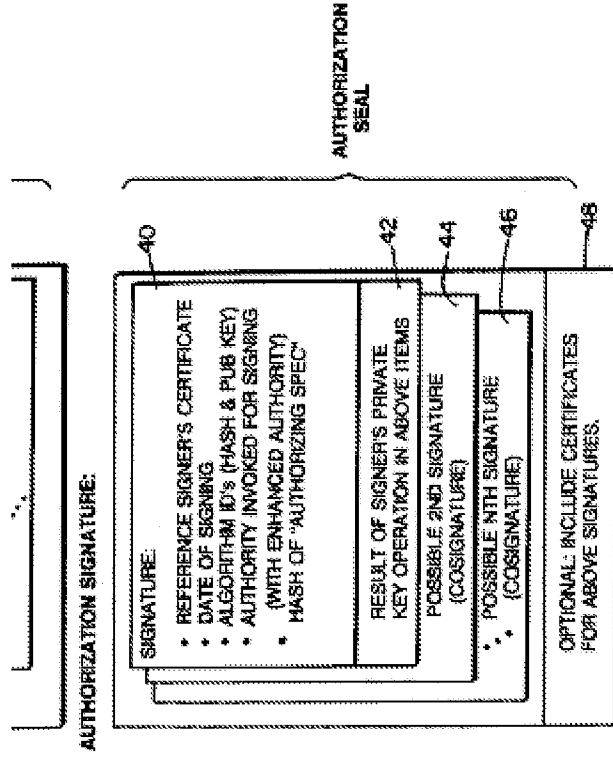
“The present method an apparatus utilizes a unique operation system design that includes

U.S. Patent No. 6,125,447 – Claim 16	Fischer
	<p>a system monitor which limits the ability of a program about to be executed to the use of predefined resources (e.g., data files, disk writing capabilities etc.). The system monitor builds a data structure including a set of authorities defining that which a program is permitted to do and/or that which the program is precluded from doing.</p> <p>The set of authorities and/or restrictions assigned to a program to be executed are referred to herein as ‘program authorization information’ (or ‘PAI’). Once defined, the program authorization information is thereafter associated with each program to be executed to thereby delineate the types of resources and functions that the program is allowed to utilize.”</p> <p><i>Fischer</i>, 2:16-30.</p> <p><i>Fischer</i> further discloses that PAI information for a program may be combined, as appropriate, with the PAI associated with a calling program. :</p> <p>“Thereafter, the program X’s program authorizing information is combined, as appropriate, with the PAI associated with the PCB of the calling program, if any. This combined PAI, which may include multiple PAI’s, is then stored in an area of storage which cannot generally be modified by the program and the address of the PAI is stored in the process control block (PCB) as indicated in field 156 of FIG. 5. Thus, if program X is called by a calling program, it is subject to all its own constraints as well as being combined in some way with the constraints of the calling program, which aggregate constraints are embodied into program X’s PAI. In this fashion, a calling program may not be permitted to exceed its assigned bounds by merely calling another program.”</p> <p><i>Fischer</i>, 19:40-54.</p> <p><i>Fischer</i> further discloses that the PAI is associated with zero or more permissions, such as a range of operations that a program may execute or may be precluded from executing:</p> <p>“The PAI defines the range of operations that a program may execute and/or defines those</p>

U.S. Patent No. 6,125,447 – Claim 16	<i>Fischer</i>
	<p>operations that a program cannot perform. The program is permitted to access what has been authorized and nothing else. In this fashion, the program may be regarded as being placed in a program capability limiting ‘safety box.’ This ‘safety box’ is thereafter associated with the program such that whenever the system monitor runs the program, the PAI for that program is likewise loaded and monitored. When the program is to perform a function or access a resource, the associated PAI is monitored to confirm that the operation is within the defined program limits. If the program attempts to do anything outside the authorized limits, then the program execution is halted.”</p> <p><i>Fischer</i>, 2:34-48. Indeed, <i>Fischer</i> discloses a PAI associated with zero permissions (e.g., “no known trustworthiness” that leads to “a wide range of restrictions”), and a PAI associated with more permissions (e.g., “an unlimited number of different resources and functions to be controlled”):</p> <p>“Even programs with no known trustworthiness can be used after program authorization information associates a wide range of restrictions to thereby allow potentially beneficial programs to be safely used--even if they do not have an official certification of trust.</p> <p>The present invention also allows an unlimited number of different resources and functions to be controlled. For example, some useful resources/functions which may be controlled include: the ability to limit a program to certain files or data sets; the ability to transmit data via electronic mail to someone outside the user’s domain; the ability of a program to create or solicit digital signatures; the ability to limit access to a program of certain security classes, etc.”</p> <p><i>Fischer</i>, 3:48-61.</p>
<p>establishing an association between said one or more protection domains and one or more sources of code; and</p>	<p><i>Fischer</i> discloses establishing an association between said one or more protection domains and one or more sources of code.</p> <p>For example, the ‘447 Patent discloses a “source of code” as “an entity from which computer instructions are received. Examples of sources of code include a file or persistent</p>

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 Patent, 3:15-21.

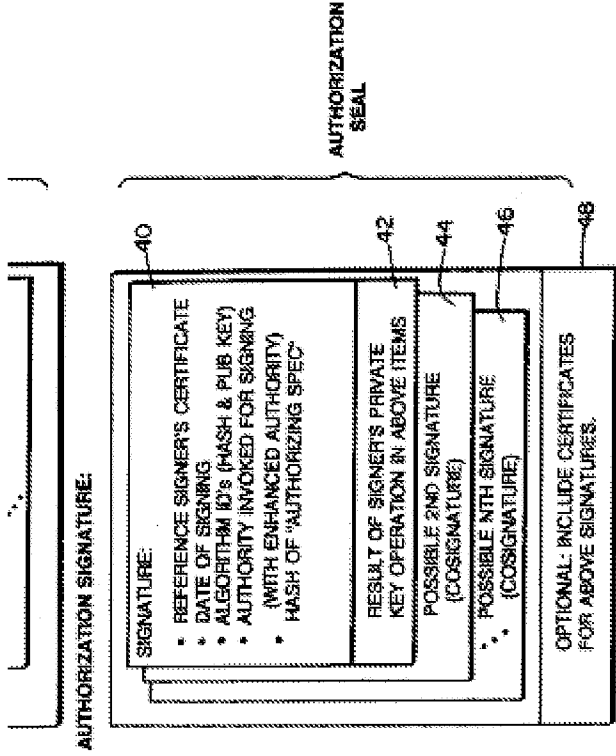
Similar to this disclosure of the “source of code” in the ‘447 Patent, *Fischer* discloses the PAI data structure (i.e., protection domain data structure), which explicitly associates the protection domain with a “source of code” such as the signer of a digital certificate:



“The authorization signature includes a signature segment 40. The signature segment 40 may include a reference to the signer’s certificate, i.e., an identifier for identifying the signer’s certificate. In accordance with a preferred embodiment of the present invention, such a digital certificate is a digital message created by a trusted entity which contains the user’s public key and the name of the user (which is accurate to the entity’s satisfaction) and possibly a representation of the authority which has been granted to the user by the party who signs the digital message.”

U.S. Patent No. 6,125,447 – Claim 16	Fischer
	<p><i>Fischer</i>, 6:25-35 (emphasis added) & Fig. 2 (excerpted).</p> <p><i>Fischer</i> expressly discloses that the digital signature may be associated with a manufacturer of the program (i.e., a source of code or “an entity from which computer instructions are received”):</p> <p>“The present invention <i>allows PAI information to be associated in any appropriate manner</i>, so that in principle a user could define one or more levels of PAI which are then combined together with perhaps a more universal PAI, <i>or with a PAI which was signed and supplied by the or [sic] manufacturer of this program.</i>”</p> <p><i>Fischer</i>, 9:3-8 (emphasis added).</p> <p>“FIGS. 6 through 9 is a flowchart illustrating an exemplary sequence of operations of a utility program for establishing program authorization information. Such a utility program prompts a user, i.e., the end user, the user’s agent, <i>or even the manufacturer</i>, to define a range of authorities which are associated with a program to be executed by the user’s system.”</p> <p><i>Fischer</i>, 11:7-13 (emphasis added).</p> <p>“If no PAI has yet been associated with the program, then a check is made to determine whether the program has an associated signed ‘pedigree’ from the manufacturer (306). Thus, if a well known manufacturer of programs has signed the program with a public key or digital certificate, then, if desired, such a program may be assigned whatever level of authority desired depending upon how much the manufacturer is trusted and the system may permit execution of such program. Such a digital signature <i>from the manufacturer can be used to verify that the associated program had not been infected with a virus since it can be determined whether or not the program is exactly the same as it was when it was generated by the manufacturer.</i>”</p>

U.S. Patent No. 6,125,447 – Claim 16	<i>Fischer</i>
<p>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.</p>	<p><i>Fischer</i>, 16:12-25 (emphasis added).</p> <p><i>Fischer</i> discloses 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.</p> <p>For example, based on the digital signature (i.e., a source of code identifying the manufacturer of the code), <i>Fischer</i> discloses determining whether a program that is executing is permitted to perform an action:</p> <p>“FIGS. 10 and 11 illustrate the sequence of operations of a supervisor program for controlling the processing of a program being executed in accordance with its program authorization information.”</p> <p><i>Fischer</i>, 15:56-59.</p> <p>“Depending on the processing in block 316 [of FIG. 10], a decision is made in block 322 whether the signatures are valid, authorized and trusted. If the signatures are not determined to be valid, then the routing branches to block 324 where the execution in program X is suppressed.”</p> <p><i>Fischer</i>, 16:66-17:3.</p> <p>“If the processing in blocks 322 and 316 reveal that the signatures are valid, then the processing in block 326 is performed.”</p> <p><i>Fischer</i>, 17:31-33.</p> <p>Accordingly, <i>Fischer</i> is clear that execution of the code is conditioned on verifying the source of code (i.e., digital signature), and that the requested action will not be permitted if the signature is not valid.</p>

U.S. Patent No. 6,125,447 – Claim 16	Fischer
<p>U.S. Patent No. 6,125,447 – Claim 17</p> <p>17. 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.</p>	<p><i>Fischer</i></p> <p><i>Fischer</i> discloses 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.</p> <p>For example, Figure 2 in <i>Fischer</i> discloses the digital signature (i.e., source of code) includes an associated public key:</p>  <p>“The authorization signature includes a signature segment 40. The signature segment 40 may include a reference to the signer’s certificate, i.e., an identifier for identifying the signer’s certificate. In accordance with a preferred embodiment of the present invention, such a digital certificate is a digital message created by a trusted entity <i>which contains the user’s public key</i> and the name of the user (which is accurate to the entity’s satisfaction)</p>

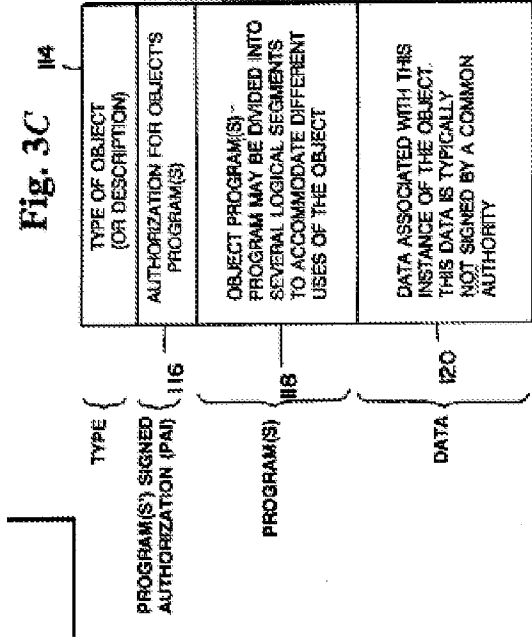
U.S. Patent No. 6,125,447 – Claim 17	<i>Fischer</i> and possibly a representation of the authority which has been granted to the user by the party who signs the digital message.” <i>Fischer</i> , 6:25-35 (emphasis added) & Fig. 2 (excerpted). Thus, <i>Fischer</i> discloses that a user’s public key may be associated with the digital signature (i.e., a source of code).
U.S. Patent No. 6,125,447 – Claim 18 18. 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.	<i>Fischer</i> <i>Fischer</i> discloses establishing an association between one or more protection domains and one or more sources of code and one or more keys associated with one or more sources of code based on data persistently stored. For example, the ‘447 Patent discloses persistently stored data such as instructions stored in a file, mappings stored in a database system, and mapping attributes of a persistent object: “Storing instructions in a file is just one method of representing the policy of the system with persistently stored data. Other methods are possible for representing the policy with persistent data. For example, data in a database system can be used to map code identifiers to authorized permissions, or attributes of a persistent object can be used to map code identifiers to authorized permissions.” ‘447 Patent, 9:19-25. Similar to this disclosure of the “persistently stored” data in the ‘447 Patent, <i>Fischer</i> discloses that the association between the protection domains and the digital signatures (i.e., sources of code) and their associated keys are based on data that is persistently stored as an attribute of a persistent object: “FIG. 3C shows an important application in which a PAI data structure is associated with a program according to an embodiment of the present invention. FIG. 3C shows an illustrative data structure for a secure exchangeable ‘object’. The data structure may be signed by a trusted authority. The signing of such a data structure allows the object to be securely transmitted from user to user. Although the data structure shown in FIG. 3 is

set forth in a general format, it may be structured as set forth in the inventor's copending application filed on Apr. 6, 1992 and entitled 'Method and Apparatus for Creating, Supporting and Processing a Travelling Program' (U.S. Ser. No. 07/863,552.), which application is hereby expressly incorporated herein by reference.

... The program authorization information is embedded in a segment 116 which specifies the authorization for the object's program or programs in a manner to be described more fully hereinafter."

Fischer, 7:49-8:2.

Figure 3C in Fischer shows these attributes of a persistent object:

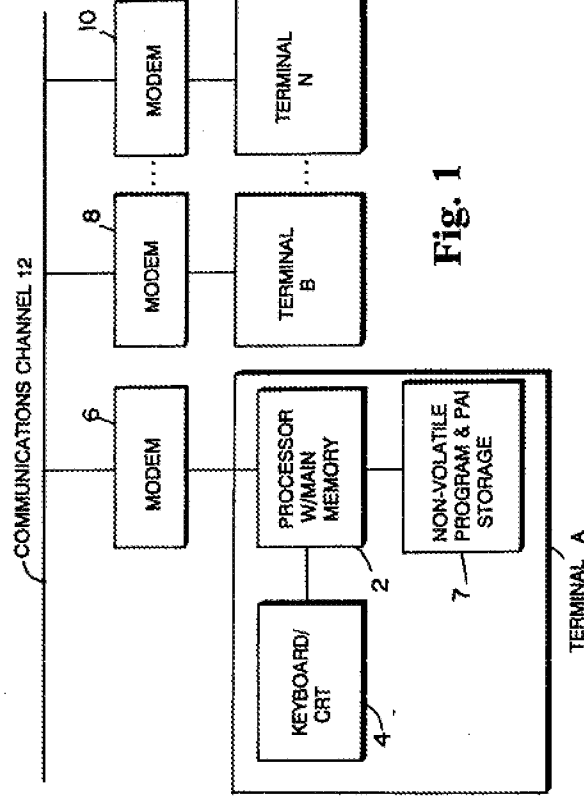


Fischer, FIG. 3C (see, e.g., element 116).

In addition to its disclosure of persistent data stored as an attribute of a persistent object, Fischer discloses more generally associating the PAI information (i.e., protection domains)

U.S. Patent No. 6,125,447 – Claim 18	<div data-bbox="196 730 228 835" data-label="Text"> <p><i>Fischer</i></p> </div> <div data-bbox="233 243 342 1377" data-label="Text"> <p>based on other types of persistently stored data. For example, <i>Fischer</i> discloses storing PAI information (which includes the digital signature/keys) on a separate/remote storage device or in the same memory as the program:</p> </div> <div data-bbox="380 296 451 1346" data-label="Text"> <p>“FIG. 3A shows an exemplary schematic representation of a system’s directory of programs. . . .</p> </div> <div data-bbox="488 195 670 1346" data-label="Text"> <p>. . . Additionally, associated with each of the program related identifiers is an indicator 84, 90, . . . 96, respectively, which identifies the location of its associated program authorization information, e.g., PAI 1. Although the program authorization information, PAI 1, is depicted as being stored in a separate memory device 100, it may, if desired, be stored in the same memory media as its associated program.”</p> </div> <div data-bbox="678 600 1149 1331" data-label="Diagram"> <p>The diagram illustrates a directory structure. A central box labeled '86' and titled 'DIRECTORY OF PROGRAMS' contains three entries: 'NAME OF PROGRAM 1' (82), 'NAME OF PROGRAM 2' (84), and 'NAME OF PROGRAM N' (92). To the right of these entries are indicators 88, 90, and 94 respectively. Arrows point from these indicators to external components: indicator 88 points to a cylinder labeled 'PROG 1' (104); indicator 90 points to a cylinder labeled 'PAI 1' (100); and indicator 94 points to a cylinder labeled 'PAI 1' (102). There is also an arrow from 'NAME OF PROGRAM 2' (84) to a cylinder labeled '100'. The entire figure is labeled 'Fig. 3A'.</p> </div> <div data-bbox="1198 1020 1230 1377" data-label="Text"> <p><i>Fischer, 7:20-35 & Fig. 3A.</i></p> </div>
<div data-bbox="1313 1400 1346 1902" data-label="Text"> <p>U.S. Patent No. 6,125,447 – Claim 19</p> </div> <div data-bbox="1351 1453 1388 1902" data-label="Text"> <p>19. A computer system comprising:</p> </div>	<div data-bbox="1313 730 1346 835" data-label="Text"> <p><i>Fischer</i></p> </div> <div data-bbox="1351 226 1421 1377" data-label="Text"> <p><i>Fischer</i> discloses a computer system. For example, <i>Fischer</i> discloses a system including IBM PC computers:</p> </div>

“FIG. 1 shows in block diagram form an exemplary communications system which may be used in conjunction with the present invention. . . . Terminals, A, B . . . N may, by way of example only, be IBM PC's having a processor (with main memory) 2 which is coupled to a conventional keyboard/CRT display 4. Additionally, each processor is preferably coupled to a non-volatile program and program authorization information (PAI) storage 7 which may be a disk memory device.”



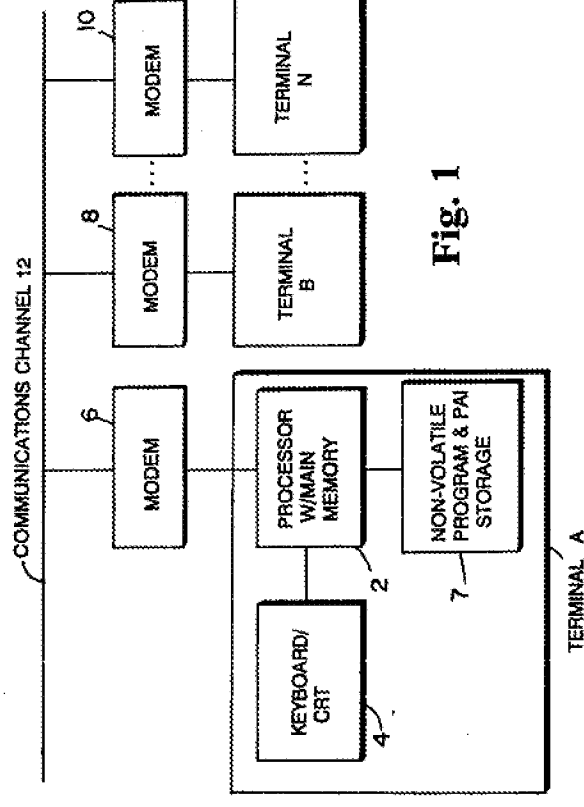
Fischer, 4:45-58 & Fig. 1.

a processor;

Fischer discloses a processor:

“FIG. 1 shows in block diagram form an exemplary communications system which may be used in conjunction with the present invention. . . . Terminals, A, B . . . N may, by way of example only, be IBM PC's having a processor (with main memory) 2 which is coupled to a conventional keyboard/CRT display 4. Additionally, each processor is

preferably coupled to a non-volatile program and program authorization information (PAI) storage 7 which may be a disk memory device.”



Fischer, 4:45-58 & Fig. 1.

one or more protection domains stored as objects in said memory,

Fischer discloses one or more protection domains stored as objects in said memory.

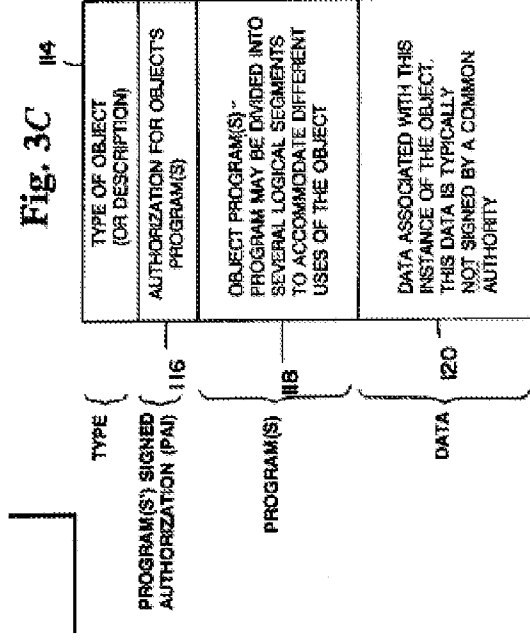
For example, *Fischer* discloses a Program Authorization Information (“PAI”) data structure as a protection domain:

“The present method an apparatus utilizes a unique operation system design that includes a system monitor which limits the ability of a program about to be executed to the use of predefined resources (e.g., data files, disk writing capabilities etc.). The system monitor builds a data structure including a set of authorities defining that which a program is permitted to do and/or that which the program is precluded from doing.

U.S. Patent No. 6,125,447 – Claim 19	<i>Fischer</i>
	<p>The set of authorities and/or restrictions assigned to a program to be executed are referred to herein as ‘program authorization information’ (or ‘PAI’). Once defined, the program authorization information is thereafter associated with each program to be executed to thereby delineate the types of resources and functions that the program is allowed to utilize.”</p> <p><i>Fischer</i>, 2:16-30.</p> <p><i>Fischer</i> further discloses that PAI information for a program may be combined, as appropriate, with the PAI associated with a calling program. :</p> <p>“Thereafter, the program X’s program authorizing information is combined, as appropriate, with the PAI associated with the PCB of the calling program, if any. This combined PAI, which may include multiple PAI’s, is then stored in an area of storage which cannot generally be modified by the program and the address of the PAI is stored in the process control block (PCB) as indicated in field 156 of FIG. 5. Thus, if program X is called by a calling program, it is subject to all its own constraints as well as being combined in some way with the constraints of the calling program, which aggregate constraints are embodied into program X’s PAI. In this fashion, a calling program may not be permitted to exceed its assigned bounds by merely calling another program.”</p> <p><i>Fischer</i>, 19:40-54.</p> <p><i>Fischer</i> discloses the PAI data structure stored as an object in memory:</p> <p>“The present invention is directed to providing reliable security, even when operating with complex data structures, e.g., objects, containing their own program instructions, which are transmitted among users.”</p> <p><i>Fischer</i>, 2:6-9.</p> <p>“Through the use of the present invention, general object oriented data may be transferred</p>

U.S. Patent No. 6,125,447 – Claim 19	Fischer
	<p>from user to user without exposing users to the potential dangers of viruses or mischievous users.”</p> <p><i>Fischer</i>, 4:10-13.</p> <p>“In one contemplated embodiment of the present invention, programs may be part of data objects, which are written in a high-level control language and are executed by a standardized interpreter program which executes this high-level language. In this case, part of the interpreter’s task is to verify that the functions encountered in the high level logic are, in fact, permissible. If such tasks are not permissible, the interpreter then suppresses the execution of the program not authorized to perform such tasks.”</p> <p><i>Fischer</i>, 3:11-20.</p> <p>“In accordance with the present invention, a PAI is associated with programs to be executed. FIGS. 3A through 3D depict four exemplary approaches for associating program authorization information with a program. . . .</p> <p>FIG. 3C shows an important application in which a PAI data structure is associated with a program according to an embodiment of the present invention. FIG. 3C shows an illustrative data structure for a secure exchangeable ‘object’. The data structure may be signed by a trusted authority. The signing of such a data structure allows the object to be securely transmitted from user to user. Although the data structure shown in FIG. 3 is set forth in a general format, it may be structured as set forth in the inventor’s copending application filed on Apr. 6, 1992 and entitled ‘Method and Apparatus for Creating, Supporting and Processing a Travelling Program’ (U.S. Ser. No. 07/863,552.), which application is hereby expressly incorporated herein by reference.</p> <p>... The program authorization information is embedded in a segment 116 which specifies the authorization for the object’s program or programs in a manner to be</p>

described more fully hereinafter.”



Fischer, 7:14-18, 7:49-8:2 & Fig. 3C.

wherein each protection domain is associated with zero or more permissions;

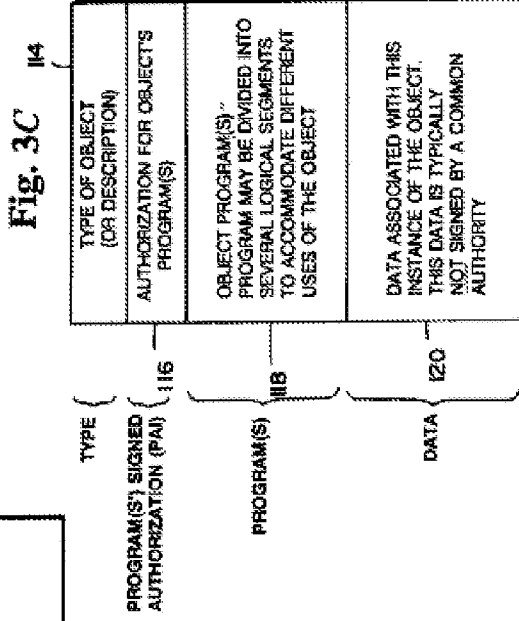
Fischer discloses each protection domain is associated with zero or more permissions. For example, the PAI is associated with zero or more permissions, such as a range of operations that a program may execute or may be precluded from executing:

“The PAI defines the range of operations that a program may execute and/or defines those operations that a program cannot perform. The program is permitted to access what has been authorized and nothing else. In this fashion, the program may be regarded as being placed in a program capability limiting ‘safety box.’ This ‘safety box’ is thereafter associated with the program such that whenever the system monitor runs the program, the PAI for that program is likewise loaded and monitored. When the program is to perform a function or access a resource, the associated PAI is monitored to confirm that the operation is within the defined program limits. If the program attempts to do anything

U.S. Patent No. 6,125,447 – Claim 19	Fischer
	<p>outside the authorized limits, then the program execution is halted.”</p> <p><i>Fischer</i>, 2:34-48. Indeed, <i>Fischer</i> discloses a PAI associated with zero permissions (e.g., “no known trustworthiness” that leads to “a wide range of restrictions”), and a PAI associated with more permissions (e.g., “an unlimited number of different resources and functions to be controlled”):</p> <p>“Even programs with no known trustworthiness can be used after program authorization information associates a wide range of restrictions to thereby allow potentially beneficial programs to be safely used--even if they do not have an official certification of trust.</p> <p>The present invention also allows an unlimited number of different resources and functions to be controlled. For example, some useful resources/functions which may be controlled include: the ability to limit a program to certain files or data sets; the ability to transmit data via electronic mail to someone outside the user’s domain; the ability of a program to create or solicit digital signatures; the ability to limit access to a program of certain security classes, etc.”</p> <p><i>Fischer</i>, 3:48-61.</p>
<p>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</p>	<p><i>Fischer</i> discloses 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.</p> <p>For example, <i>Fischer</i> discloses software that establishes an association between the PAI (i.e., protection domain) and a class/object. First, <i>Fischer</i> discloses that the PAI (i.e., a protection domain) is associated with a program:</p> <p>“Once defined, the program authorization information [(PAI)] is thereafter associated with each program to be executed to thereby delineate the types of resources and functions that the program is allowed to utilize. The PAI associated with a particular program may be assigned by a computer system owner/user or by someone who the</p>

U.S. Patent No. 6,125,447 – Claim 19	Fischer
	<p>computer system owner/user implicitly trusts.”</p> <p><i>Fischer</i>, 2:26-33.</p> <p>Next, <i>Fischer</i> discloses that a program with which a PAI is associated may be part of an object:</p> <p>“The present invention is directed to providing reliable security, even when operating with complex data structures, e.g., objects, containing their own program instructions, which are transmitted among users.”</p> <p><i>Fischer</i>, 2:6-9.</p> <p>“Through the use of the present invention, general object oriented data may be transferred from user to user without exposing users to the potential dangers of viruses or mischievous users.”</p> <p><i>Fischer</i>, 4:10-13.</p> <p>“In one contemplated embodiment of the present invention, programs may be part of data objects, which are written in a high-level control language and are executed by a standardized interpreter program which executes this high-level language. In this case, part of the interpreter’s task is to verify that the functions encountered in the high level logic are, in fact, permissible. If such tasks are not permissible, the interpreter then suppresses the execution of the program not authorized to perform such tasks.”</p> <p><i>Fischer</i>, 3:11-20.</p> <p>“In accordance with the present invention, a PAI is associated with programs to be executed. FIGS. 3A through 3D depict four exemplary approaches for associating program authorization information with a program. . . .</p>

U.S. Patent No. 6,125,447 – Claim 19	Fischer
	<p>...</p> <p>FIG. 3C shows an important application in which a PAI data structure is associated with a program according to an embodiment of the present invention. FIG. 3C shows an illustrative data structure for a secure exchangeable 'object'. The data structure may be signed by a trusted authority. The signing of such a data structure allows the object to be securely transmitted from user to user. Although the data structure shown in FIG. 3 is set forth in a general format, it may be structured as set forth in the inventor's copending application filed on Apr. 6, 1992 and entitled 'Method and Apparatus for Creating, Supporting and Processing a Travelling Program' (U.S. Ser. No. 07/863,552.), which application is hereby expressly incorporated herein by reference.</p> <p>... The program authorization information is embedded in a segment 116 which specifies the authorization for the object's program or programs in a manner to be described more fully hereinafter."</p> <p><i>Fischer</i>, 7:14-18, 7:49-8:2.</p> <p>"Thereafter, the PAI is stored using, for example, one of the approaches set forth in FIGS. 3A through 3D so that it is associated with its program 272 ..."</p> <p><i>Fischer</i>, 15:24-26. As just described, FIG. 3C discloses the program as part of an object/class. See <i>Fischer</i>, 7:49-8:2.</p> <p>To the extent <i>Fischer</i> does not expressly disclose "classes" of the objects, one of ordinary skill in the art would understand that a class, as that term is used in the '447 Patent, is a necessarily present feature of the objects disclosed in <i>Fischer</i>. To be sure, Figure 3C in <i>Fischer</i> shows an object-oriented data structure including a type segment (114), program (e.g., method) segment (118), and a data segment (120), necessarily implicating an object-oriented program architecture.</p>



Fischer, FIG. 3C.

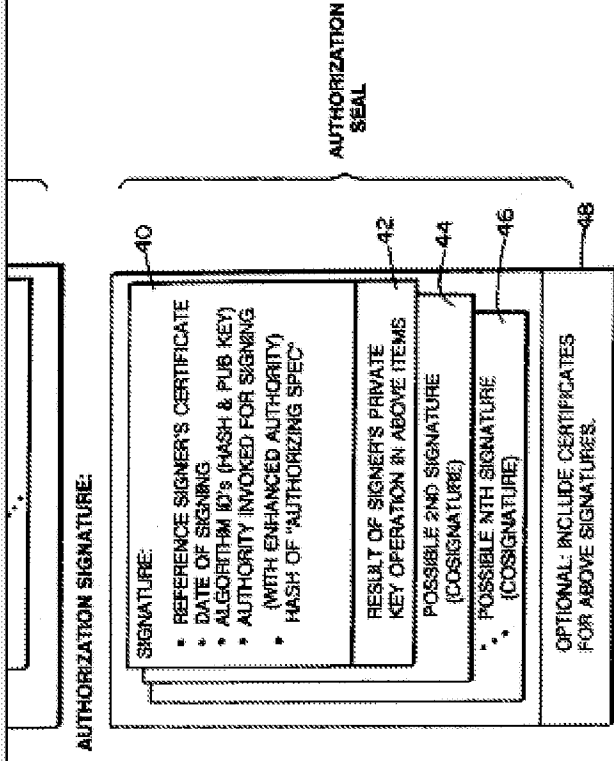
The '447 Patent discloses that a class is a high-level abstraction or definition of an object, such that "[e]ach object belonging to a class has the same fields ('attributes') and the same methods." '447 Patent, 7:4-5; *see also*, '447 Patent, 6:63-7:25.

Thus, *Fischer*'s disclosure of a data object in Figure 3C that includes fields/attributes (labeled "DATA ASSOCIATED WITH THIS INSTANCE OF THE OBJECT") as well as methods (labeled "OBJECT PROGRAM(S)") is consistent with the '447 Patent's description of a class. In addition, Figure 3C of *Fischer* mentions that the depicted object is an "instance," which further shows that *Fischer*'s disclosure includes object-oriented data structures, which are necessarily part of a class, as that term is used in the '447 Patent. *See* '447 Patent, 7:7-8 ("An object is said to be an 'instance' of the class to which the object belongs.").

Additionally, the '447 Patent admits that the ideas of classes and object instances were well known to those skilled in the art: "[C]lass definitions are generated from source code written by a programmer. For example, a programmer using a Java Development Kit enters source

U.S. Patent No. 6,125,447 – Claim 19	Fischer
	<p>code that conforms to the Java programming language into a source file. The source code embodies class definitions and other instructions which are used to generate byte code which controls the execution of a code executor (i.e. a Java virtual machine). <i>Techniques for defining classes and generating code executed by a code executor, such as a Java virtual machine, are well known to those skilled in the art.</i>" '447 Patent, 7:15-24.</p> <p><i>Fischer's</i> disclosure is entirely consistent with this admission, as <i>Fischer</i> discloses that "programs may be part of data objects, which are written in a high-level control language and are executed by a standardized interpreter program which executes the high-level language." <i>Fischer</i>, 3:12-15. In light of these disclosures in the '447 Patent and <i>Fischer</i>, there can be no doubt that one of ordinary skill in the art would view the object disclosed in <i>Fischer</i> as an instance of a class, such that the class, if not expressly disclosed, is necessarily present in the <i>Fischer</i> disclosure.</p>
<p>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.</p>	<p><i>Fischer</i> discloses the 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.</p> <p>For example,</p> <p>“FIGS. 10 and 11 illustrate the sequence of operations of a supervisor program for controlling the processing of a program being executed in accordance with its program authorization information.”</p> <p><i>Fischer</i>, 15:56-59.</p> <p>“Depending on the processing in block 316 [of FIG. 10], a decision is made in block 322 whether the signatures are valid, authorized and trusted. If the signatures are not determined to be valid, then the routing branches to block 324 where the execution in program X is suppressed.”</p> <p><i>Fischer</i>, 16:66-17:3.</p>

U.S. Patent No. 6,125,447 – Claim 19	<i>Fischer</i> “If the processing in blocks 322 and 316 reveal that the signatures are valid, then the processing in block 326 is performed.” <i>Fischer</i> , 17:31-33.
U.S. Patent No. 6,125,447 – Claim 20 20. 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;	<i>Fischer</i> <i>Fischer</i> discloses the computer system of claim 19. See claim chart above for further details. <i>Fischer</i> discloses at least one protection domain of said one or more protection domains is associated with a code identifier. For example, the ‘447 Patent discloses a code identifier as “describing the source of code that defines a class, a set of public cryptographic keys associated with the source of code, or other information which describes the source of code, or any combination thereof. A ‘source of code’ is an entity from which computer instructions are received. Examples 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 Patent, 3:13-21. Figure 3 of the ‘447 Patent discloses a policy file that includes a URL (i.e., file://somesource) and a key name (i.e., “somekey”), and describes both as code identifiers. ‘447 Patent, 9:26-37 & Fig. 3. Similar to this disclosure of the “code identifier” in the ‘447 Patent, Figure 2 in <i>Fischer</i> discloses the PAI data structure (i.e., protection domain data structure), which explicitly associates the protection domain with a “source of code” such as the signer of a digital certificate:



“The authorization signature includes a signature segment 40. The signature segment 40 may include a reference to the signer's certificate, i.e., an identifier for identifying the signer's certificate. In accordance with a preferred embodiment of the present invention, such a digital certificate is a digital message created by a trusted entity which contains the user's public key and the name of the user (which is accurate to the entity's satisfaction) and possibly a representation of the authority which has been granted to the user by the party who signs the digital message.”

Fischer, 6:25-35 (emphasis added) & Fig. 2 (excerpted).

Fischer expressly discloses that the digital signature (i.e., code identifier) may be associated with a manufacturer of the program (i.e., a source of code or “an entity from which computer instructions are received”):

“The present invention allows PAI information to be associated in any appropriate

U.S. Patent No. 6,125,447 – Claim 20	Fischer
	<p><i>manner, so that in principle a user could define one or more levels of PAI which are then combined together with perhaps a more universal PAI, or with a PAI which was signed and supplied by the or [sic] manufacturer of this program.”</i></p> <p><i>Fischer, 9:3-8 (emphasis added).</i></p> <p>“FIGS. 6 through 9 is a flowchart illustrating an exemplary sequence of operations of a utility program for establishing program authorization information. Such a utility program prompts a user, i.e., the end user, the user’s agent, or even the manufacturer, to define a range of authorities which are associated with a program to be executed by the user’s system.”</p> <p><i>Fischer, 11:7-13 (emphasis added).</i></p> <p>“If no PAI has yet been associated with the program, then a check is made to determine whether the program has an associated signed ‘pedigree’ from the manufacturer (306). Thus, if a well known manufacturer of programs has signed the program with a public key or digital certificate, then, if desired, such a program may be assigned whatever level of authority desired depending upon how much the manufacturer is trusted and the system may permit execution of such program. Such a digital signature from the manufacturer can be used to verify that the associated program had not been infected with a virus since it can be determined whether or not the program is exactly the same as it was when it was generated by the manufacturer.”</p> <p><i>Fischer, 16:12-25 (emphasis added).</i></p>
at least one class of said one or more classes is associated with said code identifier; and	<p><i>Fischer</i> discloses at least one class of said one or more classes is associated with said code identifier. For example, as discussed above, Figures 2 and 3C of <i>Fischer</i> disclose that the PAI data structure may contain a manufacturer’s signature (i.e., code identifier) (see Fig. 2), 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.</p>

U.S. Patent No. 6,125,447 – Claim 20	<div data-bbox="191 184 228 1383" data-label="Page-Header">Fischer</div> <div data-bbox="228 184 849 1383"> <div data-bbox="228 184 849 987"> <div data-bbox="228 184 300 987"> <div data-bbox="228 184 300 296">TYPE</div> <div data-bbox="228 296 300 987">PROGRAM(S) SIGNED AUTHORIZATION (PAI)</div> </div> <div data-bbox="300 184 418 987"> <div data-bbox="300 184 418 296">PROGRAM(S)</div> <div data-bbox="300 296 418 987">PROGRAM(S)</div> </div> <div data-bbox="418 184 574 987"> <div data-bbox="418 184 574 296">DATA</div> <div data-bbox="418 296 574 987">DATA</div> </div> </div> <div data-bbox="228 987 849 1383"> <div data-bbox="228 987 849 1098"> <div data-bbox="228 987 300 1098">TYPE OF OBJECT (OR DESCRIPTION)</div> <div data-bbox="228 1098 300 1383">AUTHORIZATION FOR OBJECT'S PROGRAM(S)</div> </div> <div data-bbox="300 987 418 1383"> <div data-bbox="300 987 418 1098">OBJECT PROGRAM(S)</div> <div data-bbox="300 1098 418 1383">PROGRAM MAY BE DIVIDED INTO SEVERAL LOGICAL SEGMENTS TO ACCOMMODATE DIFFERENT USES OF THE OBJECT</div> </div> <div data-bbox="418 987 574 1383"> <div data-bbox="418 987 574 1098">DATA ASSOCIATED WITH THIS INSTANCE OF THE OBJECT.</div> <div data-bbox="418 1098 574 1383">THIS DATA IS TYPICALLY NOT SIGNED BY A COMMON AUTHORITY</div> </div> </div> </div>
<p>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.</p>	<p><i>Fischer</i>, Fig. 3C (“PROGRAM(S) SIGNED AUTHORIZATION (PAI)” included as element 116 of the disclosed object/class data structure).</p> <p><i>Fischer</i> discloses associating said one or more protection domains and said one or more classes based on said code identifier. For example, <i>Fischer</i> discloses that authorization (e.g., protection domains) may be associated with a program based on the digital signature (i.e., code identifier) included in an object/class:</p> <p>“Thus, if a well known manufacturer of programs has signed the program with a public key or digital certificate, then, if desired, <i>such a program may be assigned whatever level of authority desired depending upon how much the manufacturer is trusted and the system may permit execution of such program.</i>”</p> <p><i>Fischer</i>, 16:15-20 (emphasis added).</p>
U.S. Patent No. 6,125,447 – Claim 21	<div data-bbox="1255 184 1292 1383" data-label="Page-Header">Fischer</div> <p><i>Fischer</i> discloses the code identifier indicates a source of code used to define each class of said one or more classes. For example, <i>Fischer</i> indicates that each object/class may</p>

<p>U.S. Patent No. 6,125,447 – Claim 21</p> <p>source of code used to define each class of said one or more classes.</p>	<p style="text-align: center;"><i>Fischer</i></p> <p>include the digital signature (i.e., a code identifier):</p> <p>“FIG. 3C shows an illustrative data structure for a secure exchangeable ‘object’. The data structure may be signed by a trusted authority. The signing of such a data structure allows the object to be securely transmitted from user to user.”</p> <p><i>Fischer</i>, 7:51-56.</p>
<p>U.S. Patent No. 6,125,447 – Claim 22</p> <p>22. The computer system of claim 20, wherein said code identifier indicates a key associated with each class of said one or more classes.</p>	<p style="text-align: center;"><i>Fischer</i></p> <p><i>Fischer</i> discloses the code identifier indicates a key associated with each class of said one or more classes. For example, Figure 2 in <i>Fischer</i> discloses the digital certificate (i.e., code identifier) includes a public key that may be associated with the object/class:</p> <div style="text-align: center;"> </div> <p>“The authorization signature includes a signature segment 40. The signature segment 40 may include a reference to the signer’s certificate, i.e., an identifier for identifying the signer’s certificate. In accordance with a preferred embodiment of the present invention,</p>

U.S. Patent No. 6,125,447 – Claim 22	<p><i>Fischer</i></p> <p>such a digital certificate is a digital message created by a trusted entity <i>which contains the user's public key</i> and the name of the user (which is accurate to the entity's satisfaction) and possibly a representation of the authority which has been granted to the user by the party who signs the digital message.”</p> <p><i>Fischer</i>, 6:25-35 (emphasis added) & Fig. 2 (excerpted).</p>
U.S. Patent No. 6,125,447 – Claim 23	<p><i>Fischer</i></p> <p><i>Fischer</i> discloses the code identifier indicates a source of code used to define each class of said one or more classes.</p> <p>For example, the ‘447 Patent discloses the “source of code” of a code identifier as “an entity from which computer instructions are received. Examples of sources of code include a file or persistent object stored on a data server connected over a network, a FLASH_EEPROM reader that reads instructions stored on a FLASH_EEPROM, or a set of system libraries.” ‘447 Patent, 3:15-21.</p> <p><i>Fischer</i> expressly discloses that the digital signature (i.e., code identifier) may be associated with a manufacturer of the program (i.e., a source of code or “an entity from which computer instructions are received”):</p> <p>“The present invention <i>allows PAI information to be associated in any appropriate manner</i>, so that in principle a user could define one or more levels of PAI which are then combined together with perhaps a more universal PAI, <i>or with a PAI which was signed and supplied by the or [sic] manufacturer of this program.</i>”</p> <p><i>Fischer</i>, 9:3-8 (emphasis added).</p> <p>“FIGS. 6 through 9 is a flowchart illustrating an exemplary sequence of operations of a utility program for establishing program authorization information. Such a utility program prompts a user, i.e., the end user, the user's agent, <i>or even the manufacturer</i>, to define a range of authorities which are associated with a program to be executed by the user's system.”</p>

U.S. Patent No. 6,125,447 – Claim 23	Fischer
	<p><i>Fischer</i>, 11:7-13 (emphasis added).</p> <p>“If no PAI has yet been associated with the program, then a check is made to determine whether the program has an associated signed ‘pedigree’ from the manufacturer (306). Thus, if a well known manufacturer of programs has signed the program with a public key or digital certificate, then, if desired, such a program may be assigned whatever level of authority desired depending upon how much the manufacturer is trusted and the system may permit execution of such program. Such a digital signature <i>from the manufacturer can be used to verify that the associated program had not been infected with a virus since it can be determined whether or not the program is exactly the same as it was when it was generated by the manufacturer.</i>”</p> <p><i>Fischer</i>, 16:12-25 (emphasis added).</p>
<p>... [wherein said code identifier] indicates a key associated with each class of said one or more classes.</p>	<p><i>Fischer</i> discloses the code identifier indicates a key associated with each class of said one or more classes. For example Figure 2 in <i>Fischer</i> discloses the digital certificate (i.e., code identifier) includes a public key may be associated with the object/class:</p>

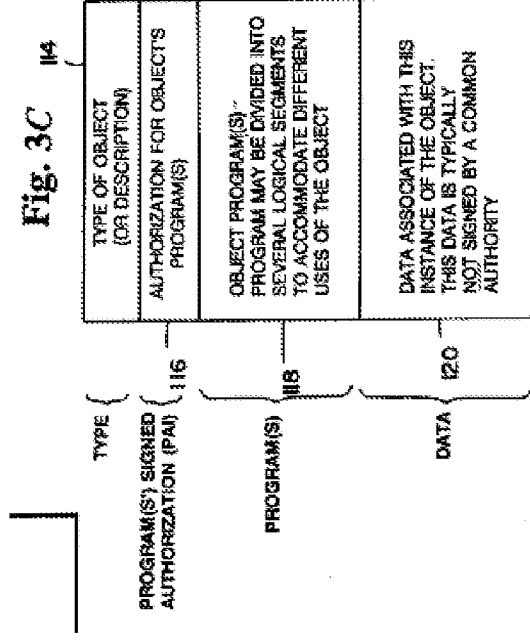
U.S. Patent No. 6,125,447 – Claim 23	<div data-bbox="191 184 837 1383"> <div data-bbox="191 184 228 1383">Fischer</div> <div data-bbox="228 184 837 1383"> <div data-bbox="228 184 289 1383">AUTHORIZATION SIGNATURE:</div> <div data-bbox="289 184 837 1383"> <div data-bbox="289 184 326 1383">SIGNATURE:</div> <div data-bbox="326 184 537 1383"> <ul style="list-style-type: none"> • REFERENCE SIGNER'S CERTIFICATE • DATE OF SIGNING • ALGORITHM ID's (HASH & PUB KEY) • AUTHORITY INVOKED FOR SIGNING (WITH ENHANCED AUTHORITY) • HASH OF "AUTHORIZING SPEC" </div> <div data-bbox="537 184 651 1383"> <div data-bbox="537 184 574 1383">RESULT OF SIGNER'S PRIVATE KEY OPERATION ON ABOVE ITEMS</div> <div data-bbox="574 184 612 1383">POSSIBLE 2ND SIGNATURE (COSIGNATURE)</div> <div data-bbox="612 184 651 1383">POSSIBLE NTH SIGNATURE (COSIGNATURE)</div> </div> <div data-bbox="651 184 688 1383">OPTIONAL: INCLUDE CERTIFICATES FOR ABOVE SIGNATURES.</div> </div> </div> <div data-bbox="537 422 574 575">AUTHORIZATION SEAL</div> </div>
--------------------------------------	--

“The authorization signature includes a signature segment 40. The signature segment 40 may include a reference to the signer’s certificate, i.e., an identifier for identifying the signer’s certificate. In accordance with a preferred embodiment of the present invention, such a digital certificate is a digital message created by a trusted entity *which contains the user’s public key* and the name of the user (which is accurate to the entity’s satisfaction) and possibly a representation of the authority which has been granted to the user by the party who signs the digital message.”

Fischer, 6:25-35 (emphasis added) & Fig. 2 (excerpted).

U.S. Patent No. 6,125,447 – Claim 24	<div data-bbox="1224 184 1261 1383">Fischer</div> <div data-bbox="1261 184 1412 1383"> <div data-bbox="1261 184 1412 1383"> <p>Fischer discloses 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.</p> </div> </div>
--------------------------------------	---

U.S. Patent No. 6,125,447 – Claim 24	Fischer
<p>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,</p>	<p>For example, the ‘447 Patent discloses persistently stored data such as instructions stored in a file, mappings stored in a database system, and mapping attributes of a persistent object:</p> <p>“Storing instructions in a file is just one method of representing the policy of the system with persistently stored data. Other methods are possible for representing the policy with persistent data. For example, data in a database system can be used to map code identifiers to authorized permissions, or attributes of a persistent object can be used to map code identifiers to authorized permissions.”</p> <p>‘447 Patent, 9:19-25.</p> <p>Similar to this disclosure of the “persistently stored” data in the ‘447 Patent, <i>Fischer</i> discloses that the association between the protection domains and the one or more classes is based on data that is persistently stored as an attribute of a persistent object:</p> <p>“FIG. 3C shows an important application in which a PAI data structure is associated with a program according to an embodiment of the present invention. FIG. 3C shows an illustrative data structure for a secure exchangeable ‘object’. The data structure may be signed by a trusted authority. The signing of such a data structure allows the object to be securely transmitted from user to user. Although the data structure shown in FIG. 3 is set forth in a general format, it may be structured as set forth in the inventor’s copending application filed on Apr. 6, 1992 and entitled ‘Method and Apparatus for Creating, Supporting and Processing a Travelling Program’ (U.S. Ser. No. 07/863,552.), which application is hereby expressly incorporated herein by reference.</p> <p>... The program authorization information is embedded in a segment 116 which specifies the authorization for the object’s program or programs in a manner to be described more fully hereinafter.”</p> <p><i>Fischer</i>, 7:49-8:2.</p> <p>Figure 3C in <i>Fischer</i> shows these attributes of a persistent object:</p>

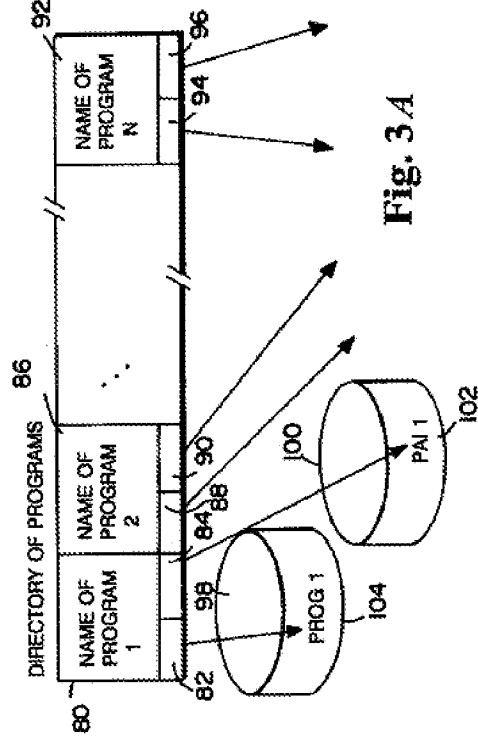


Fischer, FIG. 3C (see, e.g., element 116).

In addition to its disclosure of persistent data stored as an attribute of a persistent object, *Fischer* discloses more generally associating the PAI information (i.e., protection domains) based on other types of persistently stored data. For example, *Fischer* discloses storing PAI information on a separate/remote storage device or in the same memory as the program:

“FIG. 3A shows an exemplary schematic representation of a system’s directory of programs. . . .

. . . Additionally, associated with each of the program related identifiers is an indicator 84, 90, . . . 96, respectively, which identifies the location of its associated program authorization information, e.g., PAI 1. Although the program authorization information, PAI 1, is depicted as being stored in a separate memory device 100, it may, if desired, be stored in the same memory media as its associated program.”



Fischer, 7:20-35 & Fig. 3A.

wherein said data associates code identifiers with a set of one or more permissions.

Fischer discloses that the persistently stored data associates code identifiers with a set of one or more permissions. As discussed above, *Fischer* discloses that the persistently stored PAI segment of the Figure 3C object/class associates code identifiers (e.g., a digital signature) with a set of one or more permissions:

“The present method an apparatus utilizes a unique operation system design that includes a system monitor which limits the ability of a program about to be executed to the use of predefined resources (e.g., data files, disk writing capabilities etc.). The system monitor builds a data structure including a set of authorities defining that which a program is permitted to do and/or that which the program is precluded from doing.

The set of authorities and/or restrictions assigned to a program to be executed are referred to herein as ‘program authorization information’ (or ‘PAI’). Once defined, the program authorization information is thereafter associated with each program to be executed to thereby delineate the types of resources and functions that the program is allowed to utilize.”

U.S. Patent No. 6,125,447 – Claim 24	Fischer
	<p data-bbox="269 1157 302 1375"><i>Fischer</i>, 2:16-30.</p> <p data-bbox="342 195 667 1346">“The PAI defines the range of operations that a program may execute and/or defines those operations that a program cannot perform. The program is permitted to access what has been authorized and nothing else. In this fashion, the program may be regarded as being placed in a program capability limiting ‘safety box.’ This ‘safety box’ is thereafter associated with the program such that whenever the system monitor runs the program, the PAI for that program is likewise loaded and monitored. When the program is to perform a function or access a resource, the associated PAI is monitored to confirm that the operation is within the defined program limits. If the program attempts to do anything outside the authorized limits, then the program execution is halted.”</p> <p data-bbox="708 1157 740 1375"><i>Fischer</i>, 2:34-48.</p> <p data-bbox="781 212 886 1346">“Even programs with no known trustworthiness can be used after program authorization information associates a wide range of restrictions to thereby allow potentially beneficial programs to be safely used—even if they do not have an official certification of trust.</p> <p data-bbox="927 207 1146 1346">The present invention also allows an unlimited number of different resources and functions to be controlled. For example, some useful resources/functions which may be controlled include: the ability to limit a program to certain files or data sets; the ability to transmit data via electronic mail to someone outside the user’s domain; the ability of a program to create or solicit digital signatures; the ability to limit access to a program of certain security classes, etc.”</p> <p data-bbox="1187 1157 1219 1375"><i>Fischer</i>, 3:48-61.</p> <p data-bbox="1260 195 1398 1346">“Additionally, in block 340, an examination is made of the PAI information stored in the process control block. As a follow up to, or associated with, the processing in block 340, a check is made in block 342 to determine whether the examined PAI is allowed access to the required resources or allowed to perform the required functions. For example, if an</p>

U.S. Patent No. 6,125,447 – Claim 24	Fischer
	<p>attempt is made to use electronic mail, a check is made of the PAI to determine whether the program is authorized to perform electronic mail functions and if so whether the mailing is limited to a set of mail identifiers.</p> <p>If the check at 342 reveals that the PAI does not allow the attempted function or resource access, then a error message is generated in block 344 to indicated that the program is attempting to exceed its limits, access to the resource or function is denied and an appropriate error code or message is generated. . . .</p> <p>If the check in block 342 reveals that the PAI does allow access to the function or resource, then a check is made in block 346 to apply conventional access controls to ensure that the user of the program is still within the bounds of his authority.”</p> <p><i>Fischer</i>, 19:16-33, 19:51-55.</p>