Internal distribution code:

(A) [-] Publication in OJ
(B) [-] To Chairmen and Members
(C) [-] To Chairmen
(D) [X] No distribution

Datasheet for the decision of 15 November 2016

Case Number: T 1042/14 - 3.5.06

Application Number: 07250044.0

Publication Number: 1806674

IPC: G06F21/00

Language of the proceedings: EN

Title of invention:
Method and apparatus for protection domain based security

Applicant:
Oracle America, Inc.

Headword:
Protection domains/ORACLE

Relevant legal provisions:
EPC 1973 Art. 84, 114(1)

Keyword:
Claims - clarity (no)
Remittal to the department of first instance - (no)

Decisions cited:

Catchword:

Europäisches Patentamt
European Patent Office
Office européen des brevets

Beschwerdekammern
Boards of Appeal
Chambres de recours

European Patent Office
D-80298 MUNICH
GERMANY
Tel. +49 (0) 89 2399-0
Fax +49 (0) 89 2399-4465

Case Number: T 1042/14 - 3.5.06

D E C I S I O N
of Technical Board of Appeal 3.5.06
of 15 November 2016

Appellant: (Applicant)	Oracle America, Inc. 500 Oracle Parkway Redwood City, CA 94065 (US)
Representative:	D Young & Co LLP 120 Holborn London EC1N 2DY (GB)
Decision under appeal:	**Decision of the Examining Division of the European Patent Office posted on 13 December 2013 refusing European patent application No. 07250044.0 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman	W. Sekretaruk
Members:	M. Müller
	G. Zucka

Summary of Facts and Submissions

I. The appeal is against the decision of the examining division, with reasons dated 13 December 2013, to refuse European patent application No. 07250044.0 for lack of inventive step over the document

D7: Chen Z, "Java Card Technology for Smart Cards", Addison-Wesley, 2000.

Earlier during examination, reference was made *inter alia* to the documents

D1: Gong L *et al.*, "Implementing Protection Domains in the Java(TM) Development Kit 1.2", in Proc. of the Internet Society Symposium on Network and Distributed System Security, 1998, pages 1-10, and
D3: Gong L, "Java Security Architecture (JDK1.2)", Sun Microsystems Inc., 1998, pages 1-62.

II. Notice of appeal was filed on 9 January 2014, the appeal fee being paid on the same day. A statement of grounds of appeal was received on 8 April 2014. The appellant requested that the decision be set aside and that a patent be granted on the basis of the documents specified on page 2 of the decision.

III. In an annex to a summons to oral proceedings, the board informed the appellant of its preliminary opinion that the claims lacked clarity, Article 84 EPC 1973, and an inventive step, Article 56 EPC 1973.

IV. In response to the summons, with a letter dated 13 October 2016, the appellant filed amended claims as

a new main request, the previous claims being retained as an auxiliary request.

V. During the oral proceedings, which were held on 15 November 2016, the appellant filed further auxiliary requests but withdrew them again after discussion with the board. At the end of the oral proceedings, the chairman announced the decision of the board.

VI. The appellant's final requests were

(a) that the decision be set aside, and
(b) that the case be remitted to the examining division for further prosecution or, alternatively,
(c) that a patent be granted on the basis of claims 1-10 as filed on 13 October 2016 (main request) or claims 1-19 as filed on 10 November 2008 (auxiliary request), each in combination with description pages 2 and 2a as filed on 10 November 2008, and description pages 1 and 3-17 and drawing sheets 1/6-6/6 as originally filed.

VII. Claim 1 according to the main request reads as follows:

"A machine implemented method for providing security, the method comprising:

binding, during installation, application code (104) to a protection domain (110), based on credentials (112) with which the application code is signed;
associating a first application instance (108) with the protection domain (110) to which the application code (104) is bound, wherein the application code, when executed, gives rise to the application instance;

executing the first application instance in a first execution context (102), wherein the first execution context is isolated by a firewall (103) from other execution contexts;
receiving an indication by a firewall associated with the second execution context that the first application instance seeks access to protected functionality (208), wherein the protected functionality is exposed by a second application instance (204) that is executing in a second execution context (206A);
specifying by the second application instance a set of one or more protection domains, each of which provides permission to access the protected functionality; and
in response to receiving the indication, determining, by the firewall associated with the second execution context, whether the first application instance has permission to access the protected functionality by:
determining the protection domain with which the first application instance is associated; and
determining if the protection domain with which the first application instance is associated is in the set of one or more protection domains."

Claim 1 according to the auxiliary request reads as follows:

"A machine implemented method for providing security, the method comprising:
associating a first application instance (108) with a protection domain (110) based on credentials (112) associated with a set of application code

(104) that, when executed, gives rise to the application instance;

executing the first application instance in a first execution context (102), wherein the first execution context is isolated by a firewall (103) from other execution contexts;

receiving an indication that the first application instance seeks access to protected functionality (208), wherein the protected functionality is exposed by a second application instance (204) that is executing in a second execution context (206A), and wherein access to the protected functionality is allowed if the entity seeking access belongs to a protection domain in a set of one or more protection domains; and

in response to receiving the indication, determining whether the first application instance has permission to access the protected functionality by:

determining the protection domain with which the first application instance is associated; and

determining if the protection domain with which the first application instance is associated is in the set of one or more protection domains."

Both requests also contain independent claims for a machine-readable medium and an apparatus which correspond *mutatis mutandis* to the respective claim 1.

Reasons for the Decision

The invention

1. In the context of computing environments running several software applications side by side, the application relates to the problem of striking a balance between protecting the applications against each other and nonetheless allowing interaction between them.

1.1 A known solution to this problem is referred to as "context isolation" (see the description as originally filed, pages 1-2). Each application or "application bundle" has its own "execution context" which determines *inter alia* which "objects" or "platform functionalities" the application can access (see page 1, lines 14-16; page 2, lines 1-2; page 6, lines 1-4). Applications are isolated from each other by means of firewalls. Communication across these firewalls is exceptionally possible, for instance via so-called "shareable interface objects" (see page 1, lines 16-19).

1.2 The application seeks to provide a more flexible solution by combining the "context isolation-based security model" with a "protection domain-based security model" (page 3, last paragraph).

1.3 A "protection domain" is said to "define[] a set of permissions [...] which may be granted to" one or more "application bundle[s]". A protection domain will be bound to an application bundle if the latter can establish trust by producing credentials which "can be authenticated against [...] the protection domain

credentials". When an application requests access to some protected functionality, permission may be granted or denied based on the protection domain to which the application belongs (page 5, paragraphs 1-3; page 6, paragraph 3; page 13, lines 4-8). One protection domain may be associated with several applications or application bundles (page 6, paragraph 2).

1.4 The invention is preferably intended for smart cards using the Java Card platform, but is not limited to them (see page 3, last line, to page 4, line 2).

The prior art

2. Both D1 and D3 relate to the security architecture in the Java Development Kit JDK 1.2 which relies on a concept of "protection domains".

2.1 D1 refers to the "classical definition of a protection domain" (page 2, section 3, paragraph 1), according to which "a domain is scoped by the set of objects that are currently directly accessible by [...] an entity in the computer system to which authorizations [...] can be granted". In similar words, D3 refers to the "protection domain" as a "fundamental concept and important building block of system security", defined as "a domain [which] can be scoped by the set of objects that are currently directly accessible by a principal [...]". Both cite a 1975 paper by Saltzer and Schroeder as the basis for the concept.

2.2 Both D1 and D3 also disclose that protection domains in JDK 1.2 are implemented as dedicated data structures (section 3, last paragraph) and act as an indirection

between classes and objects on the one hand and permissions on the other hand (section 3, paragraph 2).

3. D7 relates to a mechanism provided by the Java Card platform for controlling access between packages by means of "applet firewalls" and "shareable interface objects" (see sections 9.1 and 9.2, and figures 9.1 and 9.2). The term "protection domain" is not used.

Clarity, Article 84 EPC 1973

4. The claimed invention turns on the terms "execution context" and "protection domain". Neither is defined in the independent claims of the pending requests. While the board considers that the term "execution context" of an application is clear to the skilled person, "protection domain" lacks an established meaning in the art.

4.1 The term itself may be understood as referring to some sort of "region" or "set" which represents a notion of "protection", for instance against unauthorised access, but is insufficient to imply any specific technical feature. As such, "protection domain" may denote an entirely abstract concept.

4.2 The board concedes that the term "protection domain" has been used in the art before.

4.2.1 This alone however does not mean that the term is clear. In the field of computing there are quite a number of terms which are frequently used although they do not have a clearly defined technical meaning, and in other cases a term may not have a clear technical meaning unless confined to a particular subfield.

4.2.2 Documents D1 and D3 establish that the term was used in the context of JDK 1.2, where protection domains were are also implemented in a particular manner. The claims are not, however, limited to this context, as they do not mention JDK 1.2 or Java explicitly. Nor can it be said that the mere mention of the term in the claims acts as an implicit reference to the Java context, in particular because the term apparently has been used since 1975 and thus well before Java was created. The board also notes that the description states that the invention should not be limited to the Java Card platform (page 4, lines 1-2) and thus seems to confirm that the Java context is not meant to be implied by the invention as claimed. Therefore, any specific meaning the term "protection domain" might have in the Java context cannot be taken into account in construing its meaning.

4.2.3 In the board's view, the reference to the 1975 paper in D1 and D3 is also insufficient to establish that the skilled person would understand the term "protection domain" as having a clear technical meaning in the claimed invention, and what that meaning is.

4.3 The appellant argued in its letter of 13 October 2016 (point 2.3) that "the skilled person reading claim 1 would readily understand that a protection domain is different from an execution context". Even if that were true, however, knowing that execution contexts are different from protection domains falls short of knowing what protection domains are or how they are processed.

4.4 The appellant further argued that the fact that the search examiner was able to produce documents using the term "protection domain" with the intended meaning

showed that this was its established meaning in the art.

4.5 The board disagrees. The fact that some documents use a term with the intended meaning does not exclude the possibility that the same term is used with a different meaning elsewhere. And the search examiner's choice of documents is typically based on their suitability for an assessment of novelty and inventive step. If a document shares terminology with the claimed invention, it may thus be simpler to compare its disclosure with a claim independently of what the terminology specifically means and, in particular, independently of whether the terminology itself is clear.

5. Under these circumstances the board concludes that the term "protection domain" does not imply any specific technical feature but that its meaning for the purpose of construing the claim is determined by the claim features referring to it, i.e. by the way in which the term is "used" in the claim.

6. Claim 1 of the main request specifies that a protection domain is "bound to" application code (see lines 2-3) and that it is determined whether one protection domain is in a set of one or more protection domains (see the last three lines), "each of which provides permission to access the protected functionality" (see the "specifying" step in claim 1). For brevity, the latter is henceforth referred to as the "inclusion check".

6.1 It can be deduced that any application to which a protection domain is bound is "trusted" because it must have produced its credentials (see also the

description, page 5, lines 22-23). In claim 1, this applies in particular to the first application.

6.2 However, the role of the set of protection domains specified by the second application instance is ambiguous.

6.2.1 The appellant argues in essence that the protection domains must be construed as giving access permissions independently of the inclusion check, and that the inclusion check implements the check of whether the protection domain bound to the first application instance permits the requested access.

6.2.2 The board however considers that another reasonable interpretation is also possible. According to this interpretation, the "protection domains" specified by the second instance "provide[] permission to access the protected functionality" only *due to the fact* that they are specified *for just that purpose* by the second application instance and used accordingly. For example, the second application instance may decide to trust the first application instance based on the knowledge that it must have produced its credentials in order to be bound to its protection domain, irrespective of whether access to "the protected functionality [...] exposed by [the] second application instance" is specifically permitted by the protection domain of the first application instance.

6.3 These two interpretations are substantially different from each other. In the first one, the claimed inclusion check is a way of checking the permissions provided by a protection domain, i.e. a way of enforcing the protection domains. In the second one, the claimed inclusion check provides a way of trust-

based access control which is entirely independent of the nature of the protection domains themselves: rather, in this interpretation the "protection domain" to which a first application is bound is used as a "proxy" for the credentials produced to enable the binding, and the second application instance grants access to protected functionality based on the trust thereby implied.

6.4 The appellant argued during the oral proceedings that these two interpretations, should they exist, did not render the claim unclear but only broad. Accordingly, the board should not raise a clarity objection but interpret the claim in the broadest possible way and then assess inventive step on that basis.

6.5 The board disagrees, because the two interpretations are not just alternatives of an otherwise well-understood general method but, as just explained, rather different in nature. This difference would also, in the board's view, substantially affect the assessment of inventive step.

6.5.1 In the first case, the board would have to decide *inter alia* whether the inclusion check is an obvious way of implementing the enforcement of "protection domains".

6.5.2 In the second case, it would however appear that the meaning, implementation and enforcement of protection domains *per se* were entirely independent of the claimed inclusion check. In this case, a protection domain could not be distinguished from an identifier (e.g. a number) identifying, for instance, a vendor or an application package. In particular, this reading is consistent with the term "protection domain" as such:

in the board's view, it is a plausible concept to associate access permission to application instances of a particular vendor or of a particular software package, and therefore the vendor or package identifiers may well be conceptualised as denoting a domain of protection.

7. In view of the foregoing, the board concludes that claim 1 fails to comply with Article 84 EPC 1973 due to the lack of clarity of the term "protection domain" *per se* and of the role the claimed "protection domain" plays in the claimed method, especially in the inclusion check which is of central importance for the method.

8. This conclusion applies *a fortiori* to claim 1 of the auxiliary request, which is more general than claim 1 of the main request. In particular, claim 1 of the auxiliary request lacks the "specifying" step. If anything, claim 1 therefore contains less information about the nature of protection domains or their role in the claimed method. Therefore, the board finds that claim 1 of the auxiliary request likewise lacks clarity, Article 84 EPC 1973.

Request for remittal

9. The appellant pointed out that the examining division had decided on inventive step without raising any clarity objection and expressed surprise that, under these circumstances, the board was minded to decide the case based on a lack of clarity. Therefore, the board should remit the case to the examining division for further prosecution.

10. The board disagrees.

10.1 Firstly, the board notes that it has the power under Article 114(1) EPC 1973 to examine the facts of its own motion. Therefore, it is not barred from raising a clarity objection simply because the examining division did not. This holds true in general, but even more so if, as in the present case, the clarity problem must be resolved before other issues (here: inventive step) can reasonably be addressed.

10.2 Secondly, the summons was clear about the fact that the clarity of the term "protection domain" and its role in the claimed method would be an issue in the oral proceedings (see the summons, point 6).

10.3 Thirdly, when the board raises and maintains an objection it is inappropriate to remit the case to the department of first instance for further prosecution before the objection has been overcome to the board's satisfaction.

11. Therefore, the board rejects the request for remittal.

Order

For these reasons it is decided that:

1. The request for remittal to the examining division is rejected.
2. The appeal is dismissed.

The Registrar: The Chairman:

B. Atienza Vivancos W. Sekretaruk

Decision electronically authenticated