

**Internal distribution code:**

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

**Datasheet for the decision  
of 12 July 2013**

**Case Number:** T 1137/09 - 3.5.06

**Application Number:** 04252689.7

**Publication Number:** 1480124

**IPC:** G06F 9/46

**Language of the proceedings:** EN

**Title of invention:**

Method and system for associating resource pools with  
operating system partitions

**Applicant:**

Oracle America, Inc.

**Headword:**

Operating system partitions/ORACLE

**Relevant legal provisions:**

EPC Art. 123(2)

**Relevant legal provisions (EPC 1973):**

EPC Art. 56

**Keyword:**

"Original disclosure (yes)"

"Inventive step (yes)"

**Decisions cited:**

-

**Catchword:**

-



Case Number: T 1137/09 - 3.5.06

**D E C I S I O N**  
of the Technical Board of Appeal 3.5.06  
of 12 July 2013

**Appellant:** Oracle America, Inc.  
(Applicant) 500 Oracle Parkway  
Redwood City, CA 94065 (US)

**Representative:** Harris, Ian Richard  
D Young & Co LLP  
120 Holborn  
London EC1N 2DY (GB)

**Decision under appeal:** Decision of the Examining Division of the  
European Patent Office posted 25 November 2008  
refusing European patent application  
No. 04252689.7 pursuant to Article 97(2) EPC.

**Composition of the Board:**

**Chairman:** D. H. Rees  
**Members:** S. Krischer  
W. Sekretaruk

## Summary of Facts and Submissions

I. The appeal is directed against the *decision* of the examining division, posted on 25 November 2008, to refuse the application 04252689 for lack of inventive step of claim 1 over document:

D1 US 2002/156824 A1, 24 October 2002.

II. A notice of appeal was received on 22 January 2009. The fee was received the same day. A statement of the grounds of appeal was received on 23 March 2009.

III. In its summons to oral proceedings, the board gave reasons for its opinion that the arguments set out in the appealed decision (sections 3 and 6) did not convincingly demonstrate that claim 1 lacked an inventive step. Nonetheless on the basis of prior art cited by the appellant itself, the board considered that independent claims 1, 12 and 13 lacked novelty and that the dependent claims lacked novelty or inventive step over document:

D2 P.-H. Kamp et al.: "Jails: Confining the omnipotent root."; Proceedings of the 2nd International System Administration and Networking Conference (SANE 2000); 22-25 May 2000; Maastricht, The Netherlands; pages 1-15; downloadable from <http://www.sane.nl/events/sane2000/papers.html> and <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.118.3596> .

- IV. In a letter dated 29 May 2013, the appellant filed a new set of claims.
- V. Oral proceedings were held on 12 July 2013 during which the appellant filed a new set of claims, which it made the basis of its sole request. At the end of the oral proceedings, the board announced its decision.
- VI. The appellant *requests* that the decision be set aside and a patent be granted on the basis of claims 1-15 filed during oral proceedings; description pages 1, 2 filed on 29 May 2013, pages 3, 4, 6, 8, 10-15 as originally filed, pages 5, 7, 9 and 16 filed during oral proceedings; drawing sheets 1-5 as originally filed.
- VII. Claim 1 reads as follows (differences with the refused claim are marked in *italics*; additions filed during oral proceedings are *in italics and underlined*):

"1. A method performed by an operating system *executed on a computer system (500)*, the method comprising:

establishing, within a global operating system environment (100) provided by the operating system and having a kernel (150), a plurality of non-global operating system partitions(140a, 140b) which serve to isolate processes(170) running within one non-global operating system partition from other non-global operating system partitions within the global operating system environment, wherein enforcement of boundaries between the non-global operating system partitions is carried out by the kernel:

associating, *in an association data structure (204)*, a *zone ID for* a particular non-global operating

system partition with a reference to a first resource pool (202) comprising one or more resources, wherein the resources in the first resource pool are a subset of the total set of resources available on the computer system; and

ensuring that processes running within the particular non-global operating system partition are allowed to utilize only the resources in the first resource pool, including associating each process running within the particular non-global operating system partition with the first resource pool in a data structure associated with the process using the zone ID and the reference to the first resource pool;

receiving an indication that the particular non-global operating system partition is to be associated with a second resource pool instead of the first resource pool, wherein the second resource pool is different from the first resource pool, and wherein the second resource pool comprises one or more resources;

associating, in the association data structure, the zone ID for the particular non-global operating system partition with a reference to the second resource pool instead of the first resource pool; and

ensuring that processes running within the particular non-global operating system partition are allowed to utilize only the resources in the second resource pool including associating each process running within the particular non-global operating system partition with the second resource pool instead of the first resource pool in the data structure associated with the process using the zone ID and the reference to the second resource pool.

VIII. Claim 8 is an independent computer program claim referring to any preceding (method) claim.

IX. Claim 9 is a corresponding independent computer system claim.

## **Reasons for the Decision**

### 1. *Overview*

1.1 The application *relates* to isolating processes in "non-global operating system partitions" (also called "zones" in description paragraph [6] of the A2 publication; see also figure 1). The partitions reside on top of one single operating system kernel ([8], second sentence; figure 1). An isolated process can only use resources from a resource pool associated to its partition. The resource pool can contain file systems, logical network interfaces ([7]; figure 1), as well as processors, memory or any other system resource ([19]). The enforcement of the partition boundaries is carried out by the kernel ([11]).

1.2 The application fulfils the requirements of the EPC, in particular those of Articles 123(2) EPC and 56 EPC 1973.

### 2. *Original disclosure*

2.1 The examining division did not raise any objections under Article 123(2) EPC in its decision and the board concurs that there was no reason to do so with respect to the claims as refused.

2.2 The independent claims of the present sole request have been rewritten in two steps.

2.3 First, the amendments filed with the letter dated 29 May 2013 consisted of moving the content of original claims 7-10 to claim 1, and of an analogous amendment of independent claim 13, which became claim 9.

2.4 Second, the amendments filed during the oral proceedings (see the underlined passages marked in italics in claim 1 above) are based on original description page 9, lines 6-9 (for the association between the partition and the resource pool), lines 16-21 (for the association between the init process and the resource pool), 26-30 and page 10, lines 5 and 6 (for the association between the remaining processes, which are all children of the init process, and the resource pool). These passages also disclose that each of the three associations uses a reference to a resource pool in its data structure.

2.5 As to the amendments of the *description*, they concern acknowledgment of prior art documents and clarifications that the features added to claim 1 no longer belong to an embodiment.

2.6 Thus, it is confirmed that the application satisfies the requirements of Article 123(2) EPC.

### 3. *Inventiveness of claim 1*

3.1 The appealed *decision* (3.1) identifies as the *difference* between the refused claim 1 and D1 that in claim 1 the operating system (OS) kernel enforces the

boundaries between the partitions instead of the hypervisor as in D1. The examining division agrees that kernel and hypervisor are different, however the "exact functionality of the kernel" is said not to be "defined" with the exception of the functionality that takes care of the enforcement of the partitions/zones. Since a portion of the hypervisor code shares the level of the OS kernel, it was an obvious choice of design to distribute the required functionality over components (i.e. to move the partition enforcement from the hypervisor to the kernel) or "equivalently to rename/relabel said functionality".

3.2 The *grounds* of appeal (5.1) argue in response that a "kernel" is a very well-known concept in the art, and that the skilled person understands what a kernel is and does, over and above the features specifically recited in the claim. Further (6.1), the hypervisor-based system of D1 has one OS kernel booted *in each* of the partitions, whereas the claimed invention has only *one single kernel* which creates the partitions on top of itself. This allows a cleaner partition management in the single kernel instead of "having to bolt the hypervisor management ad hoc into one of the OS kernel partitions as in D1" (6.1; i.e. the hypervisor code sharing the level of an OS kernel). Therefore (6.2-6.4), the kernel of the claim is not a relabelled hypervisor; they are different.

3.3 The board agrees. According to what would appear to be the normal terminology, the virtualisation technique used by the claimed invention is called "operating system-level virtualisation" (e.g. see [http://en.wikipedia.org/wiki/Operating\\_system-](http://en.wikipedia.org/wiki/Operating_system-)



level\_virtualization). In that field, a "partition" or "zone" is also called a "container", "virtual private server" or "jail". The expression "jails" is mentioned in the US provisional application US 2003 469558 P from which the current application claims priority. Part 1 "Virtualisation and Namespace Isolation in Solaris", chapter 2 "Related work", page 7, paragraph 2, last sentence of this priority application reads:

"Zones are based on the basic idea of jails, but extend the concept to provide a comprehensive facility that is integrated with core operating system services."

- 3.4 OS-level virtualisation has to be separated from the field of "hardware (HW) virtualisation" where one or more complete computers ("virtual machines") are simulated at the hardware level by a control program, usually called a "hypervisor" or "virtual machine monitor". Each virtual machine has then to boot its own OS, in contrast to OS-level virtualisation where one kernel simulates one or more running operating systems as containers.
- 3.5 Thus, it is not only a question of moving functionality from the hypervisor to the kernel, but a question of different functionalities in the OS-level virtualised kernel and the HW-virtualised hypervisor of D1.
- 3.6 It follows that the argumentation of the appealed decision is unable to demonstrate that refused claim 1 lacks an inventive step over D1. However, given that the priority document (but not the application) already identifies another, potentially closer prior art, the

board deemed it appropriate to check the novelty and inventive step of the claimed subject-matter against this prior art.

- 3.7 The aforementioned priority application states that the concept of "zones" is based on that of "jails". It further cites (US provisional application US 2003 469558 P, part 1 "Virtualisation and Namespace Isolation in Solaris", chapter 2 "Related work", page 7, paragraph 2, and "Bibliography", page 80) document D2 as defining the term "jails".
- 3.8 Therefore it was necessary to assess whether and to what extent the presently claimed subject-matter is distinguished from a "jail".
- 3.9 The board considers D2 to be the *closest prior art* document to current claim 1. D2 *discloses* several (non-global) operating system partitions which serve to isolate processes (page 2, paragraph 4: "... of a partitioning solution, in which customer processes and storage are isolated from those of other customers", and section 4. "The Jail Partitioning Solution"). The kernel enforces the boundaries between the partitions (section 6. "Implementation jail in the FreeBSD kernel."). A particular partition is associated with a resource pool comprising several resources (page 5, paragraph 4, second sentence: "When a jail is created, it is bound to a particular file system root."; and paragraph 5, first line: "Each jail is bound to a single IP address:"). The processes running in a particular partition are allowed to utilise only the resources in the associated resource pool (page 6, last line: "• Accessing network resources not associated with

the jail is prohibited."; page 5, paragraph 2, first line: "A process in a partition is referred to as 'in jail'."; paragraph 3, first sentence: "Membership in a jail involves a number of restrictions: access to the file name-space is restricted in the style of chroot(2), the ability to bind network resources is limited to a specific IP address,") .

3.10 In the light of these passages, the refused claim 1 lacked novelty (Article 54 EPC 1973). However, current claim 1 differs from refused claim 1 in the passages marked in italics in section VII above. Of these passages, the board considers not to be disclosed in D2 those features which relate to:

- changing the association of a partition from a first to a second resource pool (original claims 9 and 10),
- an association data structure (204) using a zone ID and a reference to a resource pool (part of the underlined passage), and
- a second data structure associated with each process containing a reference to the resource pool of the respective process (remaining part of the underlined passage).

3.11 As to the first point, the appellant argued that changing the resource pool would have the *technical effect* of increasing the flexibility of managing the resources of a partition.

3.12 However, firstly the board considers that it is an obvious wish to make the association of resources to a partition modifiable. Secondly, the board considers

that "*increasing the flexibility*" is in general too vague to be considered an appropriate technical effect for the assessment of an inventive step. Furthermore D2, page 12, paragraph 2 discloses modifying the filesystem of a jail (= partition) in the host environment. This can be considered to be changing a resource of a jail. Changing all the resources at once (= changing the resource pool) is merely a repeated application of this principle. Therefore, this feature group alone does not establish an inventive step.

3.13 As to the second point, at first glance it would seem arguable that it would be an obvious choice of a skilled person implementing the invention to use two designators (e.g. IDs or references; one for the partition and one for the resource pool) to store the association between them in a data structure.

3.14 However, one could wonder why after all identifiable (e.g. named) resource *pools* are used in the invention, since a change of the set of resources available to a partition could be easily effected by de-associating each single resource of the first set of resources from the partition, and associating each resource of the second set of resources with the partition.

3.15 The explanation is that resource pools and the redundant storing *per process* of its association to a resource pool in a second data structure serves the purpose of saving time to access the resources available to a process in its partition.

3.16 Without the second data structure and without resource pools, the operating system would have to first look up

- the zone ID for the process in a table, and then look up the association data structure with this zone ID to control access to any resource associated with the partition and its process.
- 3.17 Adding the second data structure alone (without named resource pools), the operating system would have to look up this data structure using the process ID to control access to any resource. The data structure would have to store references to all the resources available in the jail.
- 3.18 But when one uses named resource pools in addition to the second data structure, then there is only one lookup using the process ID to the reference of the resource pool, i.e. to *all* associated resources. And only one reference for all resources has to be stored. Taking the number of processes which usually exists, this reduces the storage requirements, while at the same time the access time is shortened by using the second data structure.
- 3.19 To summarise, this combination of the second data structure with references to resource pools solves in a non-obvious way the technical problem of shortening the access time to system resources while being storage efficient.
- 3.20 Thus, claim 1 is inventive in the sense of Article 56 EPC 1973.
- 3.21 The other independent claims (claim 8: computer program; claim 9: computer system) contain features correspon-

ding to each feature of claim 1. Therefore, they also are inventive.

## **Order**

### **For these reasons it is decided that:**

1. The decision under appeal is set aside.
2. The case is remitted to the department of first instance with the order to grant a patent on the basis of claims 1-15 filed on 12 July 2013 during the oral proceedings; description pages 1, 2 filed on 29 May 2013, pages 3, 4, 6, 8, 10-15 as originally filed, pages 5, 7, 9 and 16 filed during the oral proceedings; drawing sheets 1-5 as originally filed.

The Registrar:

The Chairman:

B. Atienza Vivancos

D. H. Rees