

Internal distribution code:

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

**Datasheet for the decision
of 30 June 2014**

Case Number: T 0912/10 - 3.5.06

Application Number: 02787597.0

Publication Number: 1442372

IPC: G06F9/48

Language of the proceedings: EN

Title of invention:

PROVIDING ISOLATION THROUGH PROCESS ATTACHABLE VIRTUAL
MACHINES

Applicant:

SAP AG
Kuck, Norbert
Kuck, Harald
Lott, Edgar
Rohland, Hans-Christoph
Schmidt, Oliver

Headword:

Process attachable virtual machine/SAP

Relevant legal provisions:

EPC Art. 87(1), 123(2)

Keyword:

Priority - (yes)
Amendments - added subject-matter (no)

Decisions cited:

G 0001/03

Catchword:



**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 0912/10 - 3.5.06

D E C I S I O N
of Technical Board of Appeal 3.5.06
of 30 June 2014

Appellant:
(Applicant 1)

SAP AG
Dietmar-Hopp-Allee 16
69190 Walldorf (DE)

Appellant:
(Applicant 2)

Kuck, Norbert
Karl-Hoffman-Strasse 3
76137 Karlsruhe (DE)

Appellant:
(Applicant 3)

Kuck, Harald
Kleine Mantel Gasse 7
69117 Heidelberg (DE)

Appellant:
(Applicant 4)

Lott, Edgar
Benz Strasse 2
69226 Nussloch (DE)

Appellant:
(Applicant 5)

Rohland, Hans-Christoph
Neue Heimat Strasse 8
68789 St. Leon-Rot (DE)

Appellant:
(Applicant 6)

Schmidt, Oliver
Hehleichweg 5
76189 Karlsruhe (DE)

Representative:

Müller-Boré & Partner Patentanwälte PartG mbB
Friedenheimer Brücke 21
80639 München (DE)

Decision under appeal:

**Decision of the Examining Division of the
European Patent Office posted on 18 December
2009 refusing European patent application No.
02787597.0 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman D. Rees
Members: S. Krischer
 C. Heath

Summary of Facts and Submissions

- I. The appeal is directed against the decision of the examining division, posted on 18 December 2009, to refuse the application 02787597.
- II. The examining division decided that the priority claimed (Article 88 EPC) from the following US provisional patent application was not valid (Article 87(1) EPC) for the main request (corresponding to the main request in appeal), the first and the third auxiliary requests:
- P2 US 60/400,491, filed on 1 August 2002
- III. The reasons for the refusal were lack of novelty (Article 54(1) and (2) EPC) for the main and the third auxiliary requests, lack of an inventive step (Article 56 EPC) for the first auxiliary request and lack of original disclosure (Article 123(2) EPC) for the second auxiliary request. A fourth auxiliary request was not admitted into the procedure (Rule 137(3) EPC).
- The objections of lack of novelty and lack of an inventive step for the main, the first and the third auxiliary requests were over the document:
- D1 N. Kuck et al.: "SAP VM Container: Using Process Attachable Virtual Machines to Provide Isolation and Scalability for Large Servers"; 2 August 2002; XP001208545; 2nd Java Virtual Machine Research and Technology Symposium (JAVA VM'02), San Francisco/USA; downloaded from <http://bitser.net/isolate-interest/papers/PAVM.pdf>.

- The authors of this document are the same as the inventors of the present application and the content of the document is identical to the content of the first two pages of P2 from which priority had been claimed.
- IV. A notice of appeal was received on 29 January 2010. The fee was received on 4 February 2010. A statement of the grounds of appeal was received on 25 March 2010. Claim sets of a main and two auxiliary requests were filed with the grounds. Oral proceedings were requested if the main request was rejected.
- V. In its summons to oral proceedings, the board gave reasons for its preliminary opinion that the priority was valid.
- VI. In a letter dated 23 June 2014, the appellant withdrew its request for oral proceedings.
- VII. Oral proceedings were cancelled.
- VIII. The appellant requests that the decision be set aside and a patent be granted on the basis of claims 1-20 of a main request (corresponding to the refused main request), claims 1-20 of a first auxiliary request (corresponding to the refused second auxiliary request), or claims 1-3 of a second auxiliary request, all filed (or re-filed) with the grounds of appeal. The further text is: description pages 1, 5-16 as published; pages 2, 2a, 3, 4 and 17 filed with the grounds of appeal (pages 2, 2a, 3, 4 identical to those filed on 7 March 2005; page 17 identical to the page filed on 4 November 2005); drawing sheets 1-5 as published.

IX. Claim 1 of the main request reads as follows:

"1. A client/server method comprising:
initializing (502) a process attachable virtual machine (61, 62, 112, 122) for a user session, wherein initializing includes starting the user session by allocating a memory area out of shared memory, creating the virtual machine (61, 62, 112, 122), and storing the virtual machine (61, 62, 112, 122) and a user context in the memory area;
receiving (504) a request corresponding to the user session; and binding (508) the virtual machine (61, 62, 112, 122) to a work process (52, 54, 56, 150, 160) to process (510) the request, wherein a state of the virtual machine (61, 62, 112, 122) is unpersisted ;
detaching (512) the virtual machine (61, 62, 112, 122) from the work process (52, 54, 56, 150, 160) after processing the request, wherein the state of the virtual machine (61, 62, 112, 122) is persisted."

The main request also comprises equivalent independent claims 17 for a "computer program product" and 19 for a system.

X. Claim 1 of the first auxiliary request reads as follows:

"A client/server method comprising:
initializing (502) a process attachable virtual machine (61, 62, 112, 122) for a user session, wherein initializing includes starting the user session by allocating a memory area out of shared memory, creating the virtual machine (61, 62, 112, 122), and storing the virtual machine (61, 62, 112, 122) and a user context in the memory area;

receiving (504) a request corresponding to the user session; and
binding (508) the virtual machine (61, 62, 112, 122) to a work process (52, 54, 56, 150, 160) to process (510) the request, so that the user context and a state of the process attachable virtual machine (61, 62, 112, 122) have been mapped into the address space of the work process (52, 54, 150, 160);
detaching (512) the virtual machine (61, 62, 112, 122) from the work process (52, 54, 56, 150, 160) after processing the request,
wherein only a single process attachable virtual machine at a time is mapped to the work process."

The first auxiliary request also comprises equivalent independent claims 17 for a "computer program product" and 19 for a system.

XI. Claim 1 of the second auxiliary request reads as follows:

"1. A computer program product, tangibly stored on a computer-readable medium, comprising instructions operable to cause a programmable processor to:
 initialize (502) a process attachable virtual machine (61, 62, 112, 122) for a user session;
 receive (504) a request corresponding to the user session;
 select (506) a process from a plurality of work processes (52, 54, 56, 58);
 bind (508) the virtual machine to the selected process to process the request;
 wherein initializing comprises starting the user session by allocating a memory area (182, 184) out of shared memory, creating the virtual machine (61, 62,

112, 122), and storing the virtual machine (61, 62, 112, 122) and a user context in the memory area;

wherein the memory area (182, 184) is accessible by each of the plurality of work processes (52, 54, 56, 58);

wherein binding comprises mapping a portion of the memory area into an address space of the selected process; and

wherein the mapped portion of the memory area comprises the computational state of the virtual machine."

Reasons for the Decision

1. *Overview*

1.1 The application as claimed *relates* to isolating user sessions in a server. The server runs for example Java servlets or Enterprise Java Beans (EJBs) in a virtual machine (VM; page 1, lines 9-18; see also the priority application P2, page 1, paragraph 3). According to the board's understanding, a user session can be thought as a series of related http requests of the same user who operates on his client computer a browser program which sends these http requests calling a Java servlet or a function of an EJB on the server.

Typically, a large number of user sessions is running within a single VM (page 1, lines 16-18; P2, page 1, paragraph 4). This makes isolation of the user sessions difficult. In the application, *each user session is provided with its own VM* (page 2, lines 5-6; P2, page 1, paragraph 6), a so-called "process attachable virtual machine" (PAVM) which can be

attached to *and* detached from the process in which it is executed (page 7, paragraph 1; P2, page 1, paragraph 6). When a user session starts, its corresponding PAVM is created and a memory area (called "session memory" in the description) is allocated in shared memory. This session memory stores the PAVM, the PAVM context (stack and heap) and the context of the user session (called "user context" in the claims; page 9, paragraph 2; figure 4; first step of initializing/creating a PAVM and allocating the session memory in claim 1 of all requests; P2, page 1, paragraph 8). The user context consists of a user stack and a user heap (claims 4, 7 of the main request; original claims 7, 11; page 14, paragraphs 3 and 4), and is included in the PAVM context in the case of a Java VM (page 8, paragraph 3; P2, page 11, paragraph 2 and page 15, section 5.1.2).

A pool of special OS processes (called "work processes") exists in the server to allow a PAVM to run in one of the processes in order to serve a request of the corresponding user session (page 2, lines 6-12; page 7, paragraph 2; P2, page 1, paragraph 6). This means, when a user session sends a request to the server, the PAVM corresponding to the user session is *attached* or *bound* to one of the available processes, and then can be *detached* from said process after the request has been served by the PAVM (page 15, paragraph 3; P2, page 1, paragraph 3, last line: "thread pool based architecture" and page 11, line 3: "process pool architecture"). To achieve this, the computational state of the PAVM needs to be persisted, when the PAVM is *detached* from a process, and unpersisted when the PAVM is (*re-*)*attached* to a process (page 8, paragraph 2; P2, page 1, paragraph 7). Through the use of shared memory, persisting and unpersisting become low cost operations

(page 8, lines 10-13; page 9, lines 12-14; page 15, lines 16-19; page 16, lines 5-7; P2, page 1, paragraph 8): The PAVM's session memory simply has to be mapped to/unmapped from the address space of the process; no data is actually moved or copied (page 9, lines 13-16; P2, page 1, paragraph 8). By mapping only one session memory into a process at a time, the user sessions are fully isolated (page 13, paragraph 3; page 5, paragraph 4; not claimed in the main request and the second auxiliary request; P2, page 1, paragraph 14). The description (page 15, lines 19-23; page 16, lines 7-11) also mentions an alternative persisting/unpersisting method, namely to/from files which is much slower and more complicated than using shared memory. This is missing in the priority application P2.

To summarise, a one-to-one correspondence between a user session and its PAVM is kept during the lifetime of a user session, whereas the process in which the PAVM is executed may change from request to request of that user session. The isolation between the user sessions is achieved through this one-to-one correspondence (page 7, first paragraph; P2, page 1, paragraph 6) and through a separate session memory for each user session (page 9, paragraph 2; P2, page 1, paragraph 8).

- 1.2 The board judges that all three requests are in respect of the same invention (Article 87(1) EPC) as priority application P2.

- 1.3 Claim 1 of the first auxiliary request satisfies the requirements of Article 123(2) EPC. This was denied in the appealed decision with respect to corresponding claim 1 of the then second auxiliary request.

2. *Priority*

2.1 Main request

2.1.1 According to the appealed decision (3.3, last paragraph) as it concerns the validity of the priority of application P2 for the refused main request (whose claims correspond to the current main request in appeal), P2 does not disclose that the feature of "mapping" is optional for the subject-matter of P2. It only discloses "binding" in combination with "mapping".

2.1.2 According to the grounds of appeal (page 10, paragraph 3), P2 discloses "attaching" (= "binding") without "mapping" on page 1, paragraph 6, last sentence: "User requests are dispatched to work processes, which attach the appropriate PAVM and have it process the request". According to paragraph 4 of the grounds, there is another explicit disclosure of "binding" without "mapping" in P2, namely in claims 1, 2 and 5 on page 30.

2.1.3 According to the decision (3.6), P2 consists of 5 separate documents and it is inadmissible to combine features from the first and the last part of P2, i.e. from pages 1 and 2 on the one hand and from claims 1, 2 and 5 on page 30 on the other hand. The features of the two parts are different.

2.1.4 The board agrees with the grounds that at least the second passage in P2 (i.e. claims 1, 2, 5 on page 30) explicitly discloses "binding", "detaching" (i.e. unbinding) and "(re-)binding" without "mapping". These claims do not even (at least explicitly) contain the features of unpersisting and persisting the state of

the PAVM, in contrast to claim 1 of the present main request. It is only claimed that, after its creation, the PAVM is stored. Thus, P2 even discloses the invention in a more general way than the present claims. Furthermore, the board cannot see why pages 1 and 2 on the one hand and page 30 on the other hand should represent separate embodiments. The skilled person would understand that the two parts relate to the same invention, but on a different level of detail.

2.1.5 As to the question whether "mapping" is essential (see decision 3.4; grounds, pages 8-13), the board agrees with the grounds (page 11, paragraph 3) that improving the robustness and scalability of server with virtual machines might be taken as the (subjective) technical problem. To solve this problem, the pool of processes and the one-to-one correspondence between user sessions and PAVS seems to be essential. The "mapping", on the other hand, is not. It merely adds another technical effect to the invention, namely that of efficient persisting/unpersisting. The decision seems to restrict the invention to that technical effect. The board considers this to be inappropriate.

2.1.6 In order to check the above reasoning, the board proposes the following thought experiment: suppose an application were filed with P2 as the description and the claims of page 30 of P2 as original claims. If the claims of the main request were filed as amendments would they violate Article 123(2) EPC? The board judges that they would not, since claim 1 of the main request has more details than claim 1 of page 30, and it is not disputed that each of these details is disclosed in what would be the original description (i.e. in P2). For example, claim 1 additionally contains allocating a

memory area from shared memory, persisting the state of the PAVM during detaching and unpersisting during binding. Moreover the specific combination of features specified in the claim would not add any technical teaching not already contained in P2. According to the board's understanding of the enlarged board's of appeal decision G1/03, it follows that if claim 1 would comply with Article 123(2) when taking P2 as its description, then the application can validly claim priority from P2 for this claim.

2.1.7 The fact that P2 (page 1, paragraphs 7, 8) only discloses one embodiment of persisting/unpersisting, namely by unmapping/mapping shared memory, and that the description (page 15, lines 19-23; page 16, lines 7-11) explicitly discloses a second embodiment, namely by storing to/loading from a file, does not change the scale of possible implementations covered by the features of persisting/unpersisting in claim 1, since the board considers that storing to a file is a straightforward implementation of persisting which would be known to the skilled person as part of the common general knowledge in the art. Indeed the word persisting might be regarded as slightly misleading if one did not know the whole context of the application, since it usually means "storing to a file". In this context however, it means "storing until the end of a user session". This can be achieved either by storing in (shared) memory or by storing in a file.

2.1.8 Thus, the priority of P2 is valid for the claims of the main request.

2.2 First and second auxiliary request

Since claim 1 of these requests contain the feature of "mapping", the priority of P2 is also valid for them.

2.3 Consequently, D1 does not belong to the prior art for any of the present requests and the novelty/inventive step objections over D1 raised in the decision are invalid.

3. *Original disclosure of the first auxiliary request*

3.1 According to the appealed decision (7.2), the last feature "detaching ... wherein only a single process attachable virtual machine at a time is mapped to the work process" of claim 1 of the then second auxiliary request (corresponding to claim 1 of the first auxiliary request in appeal) is not originally disclosed, since it implies "that unmapping takes place at some point between the start of the operation of detaching and the moment that another PAVM is attached to the work process". However, the application as filed only discloses "that unmapping takes place as part of the operation of detaching, i.e. between the start of the operation of detaching and the end of the operation of detaching".

3.2 The board supposes that the objection relates to the second part of the feature ("wherein ..."), since the first part also appears in the main request and is not objected to in the decision with respect to the main request.

3.3 The board cannot see a contradiction in the above cited two statements of the decision: It seems to be correct that the session memory is *unmapped before another PAVM is bound/attached* to a work process in order to guaran-

tee that only one PAVM at at time is bound/attached to a work process. It is also correct to say that *unmapping takes place as part of the operation of detaching*, since the description (page 9, lines 14-16) states that

"the process of detaching the PAVM from the OS process simply requires unmapping the PAVM's session memory from the address space of the OS process."

3.4 However, there is no indication anywhere in the application of the *duration* of the "operation of detaching". The board considers that the operation of detaching is essentially functionally defined by the description, so that the "end of the operation of detaching" would be understood by the skilled person to be after the unmapping has happened, independently of when the unmapping happens (immediately after the detaching from a certain work process or just before binding another PAVM to that work process; the latter is what the decision calls "lazy unmapping" on page 11, paragraph 2). With this definition, there is no contradiction between the two statements.

Note that if unmapping is deferred until shortly before re-binding the work process to another PAVM ("lazy unmapping"), then also detaching is deferred (one could say "lazy detaching" in analogy). Moreover the board also reads P2 as allowing "lazy detaching" (including "lazy unmapping"), since there is also no indication of the point in time when the detaching takes place.

3.5 Furthermore, the board considers the two passages in the original description (page 4, lines 22, 23 and page 13, lines 17-19) as directly and unambiguously disclosing the second part of the feature ("where-

in ..."). It cannot see any reason why this part should not be added to the first part of the feature, even though the decision (7.3, last paragraph) states that these passages could not be added to an embodiment not having the unmapping feature.

The board agrees with this statement as it stands, but is of the opinion that detaching implicitly includes unmapping (because of the above description passage stating that detaching is achieved by unmapping). Thus, the embodiment contains unmapping and the feature of mapping only one single PAVM at a time can be added.

3.6 Therefore, claim 1 of this request complies with Article 123(2) EPC.

4. *Remittal*

4.1 The decision under appeal mainly discussed the validity of the priority and examined novelty and inventive step only with respect to document D1 which constituted prior art as a consequence of the denied priority of P2.

4.2 Since the examining division has not yet the opportunity to determine the closest prior art on the basis of a valid priority of P2, and therefore also has not yet examined novelty and inventive step of the present claims with respect to such a closest prior art, the board remits the case to the first instance for further prosecution.

Order

For these reasons it is decided that:

- 1) The decision under appeal is set aside.
- 2) The application is remitted to the department of first instance for further prosecution.

The Registrar:

The Chairman:



B. Atienza Vivancos

D. Rees

Decision electronically authenticated