

**Internal distribution code:**

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

**Datasheet for the decision  
of 19 January 2023**

**Case Number:** T 3006/19 - 3.5.07

**Application Number:** 15771812.3

**Publication Number:** 3191983

**IPC:** G06F17/30

**Language of the proceedings:** EN

**Title of invention:**

Dynamic code management

**Applicant:**

Liveperson Inc.

**Headword:**

Dynamic code management/LIVEPERSON

**Relevant legal provisions:**

EPC Art. 56, 123(2)

**Keyword:**

Amendments - main request (not allowable)  
Inventive step - auxiliary requests 1 and 2 (no)

**Decisions cited:**

G 0001/19



**Beschwerdekammern**  
**Boards of Appeal**  
**Chambres de recours**

Boards of Appeal of the  
European Patent Office  
Richard-Reitzner-Allee 8  
85540 Haar  
GERMANY  
Tel. +49 (0)89 2399-0  
Fax +49 (0)89 2399-4465

Case Number: T 3006/19 - 3.5.07

**D E C I S I O N**  
**of Technical Board of Appeal 3.5.07**  
**of 19 January 2023**

**Appellant:** Liveperson Inc.  
(Applicant) 475 10th Avenue  
5th Floor  
New York, NY 10018 (US)

**Representative:** Mewburn Ellis LLP  
Aurora Building  
Counterslip  
Bristol BS1 6BX (GB)

**Decision under appeal:** **Decision of the Examining Division of the  
European Patent Office posted on 17 June 2019  
refusing European patent application  
No. 15771812.3 pursuant to Article 97(2) EPC**

**Composition of the Board:**

**Chair** J. Geschwind  
**Members:** R. de Man  
M. Jaedicke

## **Summary of Facts and Submissions**

I. The appellant (applicant) filed an appeal against the decision of the examining division refusing European patent application No. 15771812.3, which was published as international application WO 2016/040494.

II. The contested decision cited the following document:

D1: US 2012/072488 A1, 22 March 2012.

The examining division decided that the main request and auxiliary requests 1, 2 and 3 did not comply with Article 123(2) EPC and that the subject-matter of their independent claims lacked inventive step over document D1.

III. With its statement of grounds of appeal, the appellant maintained auxiliary requests 1 and 3 considered in the decision under appeal as its main request and auxiliary request 1, and filed amended auxiliary requests 2 and 3.

IV. In a communication issued under Rule 100(2) EPC, the board introduced the following documents:

D2: B. Smus: "A non-responsive approach to building cross-device webapps", 24 April 2012, archived at <https://web.archive.org/web/20120503225147/http://www.html5rocks.com/en/mobile/cross-device/>;

D3: US 2013/0174012 A1, 4 July 2013.

It expressed the preliminary view that the main request did not comply with Articles 84 and 123(2) EPC and that

the subject-matter of its claim 1 lacked an inventive step both over a combination of documents D1 and D2 and over a combination of documents D1 and D3. The auxiliary requests were likewise not allowable.

- V. With a letter dated 9 May 2022, the appellant maintained its main request and replaced its pending auxiliary requests with new auxiliary requests 1 and 2.
- VI. In a communication accompanying the summons to oral proceedings, the board expressed the preliminary opinion that the main request did not comply with Articles 84 and 123(2) EPC and that auxiliary requests 1 and 2 did not comply with Article 56 EPC.
- VII. In a letter dated 29 November 2022, the appellant maintained its requests.
- VIII. Oral proceedings took place on 19 January 2023. At the end of the oral proceedings, the Chair announced the board's decision.
- IX. The appellant's final requests were that the decision under appeal be set aside and that a patent be granted on the basis of the claims of the main request or, in the alternative, of one of auxiliary requests 1 and 2.
- X. Claim 1 of the main request reads as follows:

"A computer-implemented method, comprising:

loading a webpage (720, 1106, 1708) on a device (110, 702, 704, 706), the webpage including a static tag (726);

during loading the webpage, determining a factor, wherein factors include a device-specific attribute;

during loading the webpage, executing instructions included in the static tag by the device, thereby causing injecting of dynamic code from a code server into the webpage, wherein the dynamic code includes a plurality of pieces (736, 738, 740) of available code;

evaluating a rule (724, 734, 252) using the factor to select a piece of the available code to be executed; and

executing the selected piece of available code to cause a device-specific application to load on the device."

Claim 2 of the main request reads as follows:

"The method of claim 1, wherein the factor includes customer identification information, and wherein the available code to be executed is selected based on the customer identification information."

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

"A computer-implemented method, comprising:

loading a webpage (720, 1106, 1708) on a device (110, 702, 704, 706), the webpage including a static tag (726);

during loading the webpage, accessing a factor, wherein the factor includes a device-specific attribute;

during loading the webpage, executing instructions included in the static tag by the device, thereby causing injecting of dynamic code from a code server into the webpage, wherein the dynamic code includes a plurality of pieces (736, 738, 740) of available web code;

evaluating a rule (724, 734, 252) using the factor to select a piece of the available web code to be executed; and

executing the selected piece of available web code to cause a device-specific web application to be presented on the device."

XII. Claim 1 of auxiliary request 2 reads as follows:

"A computer-implemented method, comprising:

loading a webpage (720, 1106, 1708) on a device (110, 702, 704, 706), the webpage including a static tag (726);

during loading the webpage, executing instructions included in the static tag by the device, thereby causing injecting of dynamic code from a code server into the webpage, wherein the dynamic code includes a plurality of pieces (736, 738, 740) of available web code, executable code and a rule;

during loading the webpage, executing the executable code to access a factor, wherein the factor includes a device-specific attribute;

evaluating the rule (724, 734, 252) using the factor to select a piece of the available web code to be executed; and

executing the selected piece of available web code to cause a device-specific web application to be presented on the device."

XIII. The appellant's arguments, where relevant to this decision, are discussed in detail below.

## **Reasons for the Decision**

### 1. *The invention*

1.1 Paragraph [0028] of the application explains that a supplier of web code, such as a web application, can develop multiple pieces of web code targeted for different environments or situations. For example, a first piece of web code can present a first web application (version) designed for a first type of mobile device, a second piece of web code can present a second web application (version) designed for a second type of mobile device, and a third piece of web code can present a third web application (version) designed for a desktop device. According to paragraph [0027], web code can be "any JavaScript tag executable by the user device".

1.2 The invention relates to a technique for allowing a web browser running on a user device to selectively execute the piece of web code which targets the specific type of user device.

Essentially, a webpage is loaded which includes instructions which cause a plurality of pieces of web code to be retrieved from a code server and injected into the webpage as "dynamic code". On the basis of a "device-specific attribute", the piece of web code corresponding to the specific user device is selected.

*Main request*

2. *Added subject-matter - Article 123(2) EPC*

2.1 According to the appellant, claim 1 of the main request is based on a combination of original claims 1, 2 and 5 and features taken from paragraph [0131] of the original description.

2.2 According to the last step of claim 1, a selected piece of available code is executed "to cause a device-specific application to load on the device". This feature is not present in any of original claims 1, 2 and 5.

Paragraph [0131] discloses that pieces of web code 736, 738, 740, which are part of "available code 750" included in dynamic code 722, include the programming necessary to *present* respective applications 742, 744, 746. Paragraphs [0135], [0136] and [0137] confirm that executing a piece of web causes a web application to be *presented* on the device. Hence the pieces of web code 736, 738, 740, which are already loaded on the device, implement device-specific versions of the application and do not cause such versions to be loaded.

The passages cited by the appellant therefore do not provide a basis for the last step of claim 1.

2.3 Since the board is not aware of any other passage in the published application which could provide a basis, it concludes that the subject-matter of claim 1 extends beyond the content of the application as filed and that the main request therefore does not comply with Article 123(2) EPC.



*Auxiliary request 1*

3. *Inventive step - Article 56 EPC*

3.1 Claim 1 is directed to a computer-implemented method which allows a device to select and execute a version of a web application which is specific to the type of the device.

First, the device loads a webpage which includes a static tag containing instructions.

During the loading of the webpage, the instructions are executed. This causes "dynamic code" to be retrieved from a code server and injected into the webpage.

The dynamic code includes a plurality of pieces of available web code. Each piece of code is specific to a particular device type (see paragraph [0131]).

The device selects the appropriate piece of code by accessing a "factor" which includes a device-specific attribute and by "evaluating a rule" using the "factor". The selected piece of code is executed, which causes the device-specific web application to be presented on the device.

3.2 Document D1 relates to a content management system for delivering customised content or program instructions to a user device (see abstract). Part of this system is a client-side tag manager program 204 written in JavaScript (paragraphs [0030] and [0037]).

3.3 First, the user device loads a webpage which includes a static "<script>" tag referencing the tag manager JavaScript code 204 (paragraph [0034]).

The board notes that a "<script>" tag in a webpage causes the web browser of the user device to retrieve an external script file and to include, i.e. "inject", the content of the file into the webpage.

- 3.4 Once the tag manager code has been injected into the webpage, it is executed and may request "page specific instructions" from a server component acting as code server (paragraph [0042]). The requested code is then executed by the web browser at the user device (ibid.).
- 3.5 In one embodiment, the "page specific instructions" are customised to the user device; for example, instructions delivered to an iPhone browser are JavaScript instructions specific to that browser (paragraph [0053]). The customised instructions may be selected by the server component on the basis of, *inter alia*, the user-agent field included in the HTTP request (paragraph [0052]). According to paragraph [0029] of the present application, the user agent string is a device-specific attribute.

The board considers that the evaluation of the user-agent field and any other relevant factors to select the appropriate set of customised instructions involves evaluating a "rule" (see also paragraph [0045] of document D1, which refers to "rules" for delivering executable instructions to a user device).

Thus, in this embodiment, the dynamic code injected into the webpage is a piece of device-specific available web code which, when executed, will cause a "device-specific application" to be presented.

3.6 Hence, the subject-matter of claim 1 differs from the disclosure of document D1 in that the dynamic code received from the code server includes a plurality of pieces of web code corresponding to different devices, from which the user device selects the appropriate device-specific piece of web code for execution. In document D1, the code server selects the appropriate device-specific piece of web code and transmits only that piece to the user device.

3.7 The appellant argued that the distinguishing features achieved a technical effect in that the "the order of steps in loading the webpage [was] more adaptable". In the invention, the steps of (i) accessing a factor and (ii) executing instructions included in the static tag could occur in any order because the code server did not need to have knowledge of the device-specific attribute. In document D1, the factor had to be accessed before the code server could select the appropriate web code for transmission and injection.

It is true that claim 1 does not specify the order in which steps (i) and (ii) have to be performed. However, an appeal to the broadness of a claim is not a valid argument in support of inventive step. Claim 1 encompasses embodiments in which the order of the steps is fixed and which therefore lack the alleged adaptability. The alleged effect is thus not achieved over the whole scope of the claim and therefore cannot be considered as the basis for the inventive step analysis (see decision G 1/19, OJ EPO 2021, A77, point 82).

3.8 The appellant further argued that, by injecting all of the available code into the webpage, it was possible to

re-evaluate the rule later and re-select the same or a different piece of code.

However, such possibilities are not claimed. Moreover, there would be no point in selecting and executing a different piece of web code which is specific to a different type of user device.

- 3.9 The appellant also argued that determining the factor and evaluating the rule not at the user device, instead of at the server, was desirable for security reasons.

The application as filed does not mention or suggest any need to hide the "factor" or the device-specific attribute from the server for security reasons. In the context of the problem-and-solution approach, it is therefore questionable that an alleged security improvement can permissibly be included in the formulation of the problem to be solved (see Case Law of the Boards of Appeal, 10th edition, 2022, I.D. 4.4.1).

Moreover, the device-specific "user agent" attribute is included in the header of every HTTP request (see D1, paragraph [0052]; D2, page 4, last full paragraph). Consequently, modifying the method of document D1 in accordance with the distinguishing features would not prevent the "user agent" attribute from being communicated to the code server and therefore would not increase security in the manner suggested by the appellant.

- 3.10 The appellant further argued that the distinguishing features resulted in a reduced processing load for the code server, since they shifted the processing power required to select the appropriate piece of web code

from the code server to the user device. In addition, they reduced the required number of communications between the code server and the user device since there was no need for the user device to send the factor to the code server.

However, the distinguishing features also result in the code server having to transmit not only the piece of web code appropriate for the requesting user device but also the pieces of web code targeting other types of user device. Whether there is any reduction of processing load at the server is therefore highly doubtful and will depend on details of the implementation, such as the number of pieces of web code and their sizes, which are not specified in the claim.

Moreover, there is no reduction in the number of communications between the user device and the code server, since, as mentioned in point 3.9 above, the device-specific "user agent" attribute is anyway included in the regular header of the HTTP request transmitted to the server.

These alleged effects are therefore also not achieved over the whole scope of the claim.

- 3.11 In the board's view, the claimed method does not provide a specific technical advantage over the method of document D1 over the whole scope of the claim. The board therefore considers that the distinguishing features solve the problem of providing an alternative implementation of selectively executing device-specific web code in a web browser.

The claimed solution consists in retrieving the (dynamic) code for all device types and selectively executing the version appropriate for the device, for example by means of well-known conditional execution.

- 3.12 The appellant argued that document D1 taught away from the distinguishing features because it consistently discussed sending only selected program instructions to the user terminal. The entirety of document D1 reinforced the idea that the server determined the appropriate program instructions and that only those program instructions were sent to the user device. Moreover, the skilled person would not have considered the claimed solution because it was less efficient in that it required the code server to transmit more code to the user device.

The board agrees that document D1 does not disclose the distinguishing features and instead proposes letting the code server select the appropriate piece of web code. However, this means that the claimed subject-matter is novel, not that document D1 "teaches away" from the claimed invention in the sense that it would dissuade the skilled person from considering alternative solutions.

The board further notes that, contrary to the appellant's view, the fact that the skilled person would need to accept a foreseeable disadvantage to arrive at the claimed invention is not an indication of non-obviousness (see Case Law of the Boards of Appeal, *supra*, I.D.9.21.1).

- 3.13 Starting from document D1 and faced with the above-mentioned problem, the skilled person would consider document D3, which relates to generating a mobile-

optimised website (see abstract) and is therefore in the same technical field as the invention.

Document D3 discloses inserting, by a website generation engine, audio auto-detection and configuration code into a webpage (paragraph [0094]). When executed by a browser, this code detects the mobile device type by parsing the user agent string and then selects and executes audio configuration code specific to the detected device (paragraph [0095]). The configuration code versions for the different supported device types are included in the code inserted into the webpage (see paragraphs [0097] and [0098]). Although paragraphs [0094] to [0096] refer to the audio auto-detection and configuration code as ".html" code, paragraphs [0097] and [0098] make clear that it is JavaScript code, and thus "web code" (see point 1.1 above).

Document D3 thus discloses selectively executing device-specific code in a web browser by letting the web browser retrieve the (dynamic) code for all device types and selectively executing the version appropriate for the device by means of conditional execution. The retrieved dynamic code includes executable code for detecting the device type and selecting the piece of code appropriate for the device type.

Hence, the skilled person, starting from document D1 and faced with the above-mentioned problem, would find the claimed solution to the problem posed in document D3 and apply it to document D1. Although in document D3 the code is already part of the webpage as downloaded from the server, in document D1 the tag manager retrieves the selected piece of code from the code server for injection into the web page, and the skilled

person would therefore let the tag manager retrieve the pieces of code for all device types from the code server. Since the tag manager itself is (retrieved, injected and) executed as a result of the web browser executing the static <source> tag (see point 3.3 above), the resulting method falls within the scope of claim 1.

3.14 Referring to the sentence in paragraph [0095] of document D3 stating that "the mobile website generation engine 172 generates a user agent string query to get the user agent string, and then extracts the browser type and/or device type", the appellant argued that, in document D3, it was the server-side mobile website generation engine 172, not the browser, which selected the appropriate piece of web code on the basis of the user agent string.

However, from the surrounding text it is clear that this sentence in paragraph [0095] is formulated inaccurately. According to paragraph [0094], the website generation module 172 adds HTML code to automatically detect the browser and/or mobile device type and to select and configure the correct audio player to work on the mobile device that requested a webpage. The first sentence of paragraph [0095] refers to Figure 9B, which illustrates "the operation of the audio autodetection and configuration .html code when encountered in a m. web page by a browser 105". Hence, the detection is carried out by the browser. It is correct that the mobile website generation engine 172 "generates a user agent string query to get the user agent string", i.e. the inserted code includes an instruction, to be executed by the browser, that queries/accesses the user agent string, but it is the browser, not the mobile website generation engine 172,



which executes the inserted code and "extracts the browser type and/or device type".

- 3.15 The appellant further argued that if the skilled person were to adapt the method of document D1 to implement the solution disclosed by document D3, they would replace the tag manager and directly retrieve a webpage which included the plurality of pieces of available web code. The resulting method would not fall within the scope of claim 1, which required the pieces of available web code to be retrieved and injected into the webpage by instructions contained in a static tag included in a webpage.

The board does not agree that the skilled person, starting from the embodiment in document D1 referred to in point 3.5 above, would be prompted by document D3 to discard the tag manager and the <script> tag which retrieves and injects the tag manager code. The tag manager is central to document D1, and its purpose goes beyond its function of retrieving program instructions adapted to the browser of the user device. Moreover, even if the tag manager were an optional aspect of document D1 that could be omitted for the sake of simplification, the decision not to do so could not support an inventive step.

- 3.16 In view of the above, the subject-matter of claim 1 of the first auxiliary request lacks inventive step (Article 56 EPC).

*Auxiliary request 2*

4. *Inventive step*

4.1 Claim 1 of auxiliary request 2 adds to claim 1 of the auxiliary request 1 that:

- the dynamic code further includes executable code and a rule; and
- the factor, including a device-specific attribute, is accessed by executing the executable code.

4.2 In document D3, the code for accessing the device-specific attribute and for selecting a piece of code for execution by evaluating a rule using the factor is part of the same "dynamic code" that includes the plurality of pieces of available code (see point 3.13 above, and see in particular the JavaScript code listing shown in paragraph [0098] of document D3).

Hence, when applying the solution disclosed in document D3 to the method of document D1, the skilled person would also arrive at the features added to claim 1 of auxiliary request 2.

4.3 The appellant's arguments in support of inventive step, as far as relevant to the board's reasoning, have already been dealt with in point 3. above.

4.4 It follows that the subject-matter of claim 1 of auxiliary request 2 lacks inventive step (Article 56 EPC).

**Order**

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

The appeal is dismissed.

The Registrar:

The Chair:



S. Lichtenvort

J. Geschwind

Decision electronically authenticated