

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

UNILOC 2017 LLC,
Patent Owner.

Case IPR2020-00023
Patent 6,467,088 B1

Before SALLY C. MEDLEY, MIRIAM L. QUINN, and
SEAN P. O'HANLON, *Administrative Patent Judges*.

QUINN, *Administrative Patent Judge*.

DECISION
Granting Institution of *Inter Partes* Review
35 U.S.C. § 314

I. INTRODUCTION

Microsoft Corp. (“Petitioner”) filed a Petition (Paper 2, “Pet.”) requesting *inter partes* review of claims 1–4, 6–14, and 16–21 of U.S. Patent No. 6,467,088 B1 (Ex. 1001, “the ’088 patent”). Uniloc 2017 LLC (“Patent Owner”) timely filed a Preliminary Response (Paper 6, “Prelim. Resp.”).

Under 35 U.S.C. § 314(a), an *inter partes* review may not be instituted unless the information presented in the petition “shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.” For the reasons stated below, we determine that Petitioner has demonstrated a likelihood of prevailing with respect to at least one challenged claim. We, therefore, institute *inter partes* review.

A. Related Matters

The parties identify the following district court proceedings involving the ’088 patent: *Uniloc USA, Inc. and UNILOC Luxembourg, S.A. v. Apple Inc.*, 1:18-cv-00296 (W.D. Tex.), filed April 9, 2018; *Uniloc 2017 LLC v. Microsoft Corporation*, 8:19-cv-00956 (C.D. Cal.), filed May 20, 2019; and *Uniloc USA, Inc. and Apple Inc.*, 6:19-cv-00532 (W.D. Tex.), filed September 10, 2019. Pet. x; Prelim. Resp. 11–12; Paper 4, 2.

In addition to this Petition, the ’088 patent was also challenged by a different party, Apple Inc., in IPR2019-00056 (“the Apple IPR”).

B. The ’088 Patent

The ’088 patent is directed to techniques for upgrading or reconfiguring software and/or hardware components in electronic devices.

Ex. 1001, 1:6–9. The '088 patent explains that prior art systems developed for updating components of electronic devices rely on a central computer system that tracks all software configurations for a number of remote systems. *Id.* at 1:31–36. These prior art systems updated software by the central computer transmitting patches to each of the remote systems. *Id.* at 1:39–42; *see also id.* at 2:4–10 (explaining that a distributed system transmits patches to mobile units). Other known techniques for software update involve assuming that each desktop computer has a set of resources determined in accordance with a set of enterprise policies or a central server maintaining a master list that is used to keep files on a remote device updated to the latest version. *Id.* at 1:49–52, 1:60–65. According to the '088 patent, all of the above techniques fail to avoid potential conflicts and ensure compatibility because they do not account for interdependencies of the resources required by the desktops or the files resident in the remote devices. *Id.* at 1:41–45, 1:52–56, 1:65–2:3, 2:10–14.

The '088 patent solves the problem by providing a list or listing, that indicates “which of a set of software components supported by the manager 10 are known to work well together or are otherwise compatible.” *Id.* at 3:36–42. For instance, Figure 1 of the '088 patent, reproduced below, illustrates reconfiguration manager 10 that includes a listing 16 of known configurations, and a repository 18 of software components. *Id.* at 3:27–29.

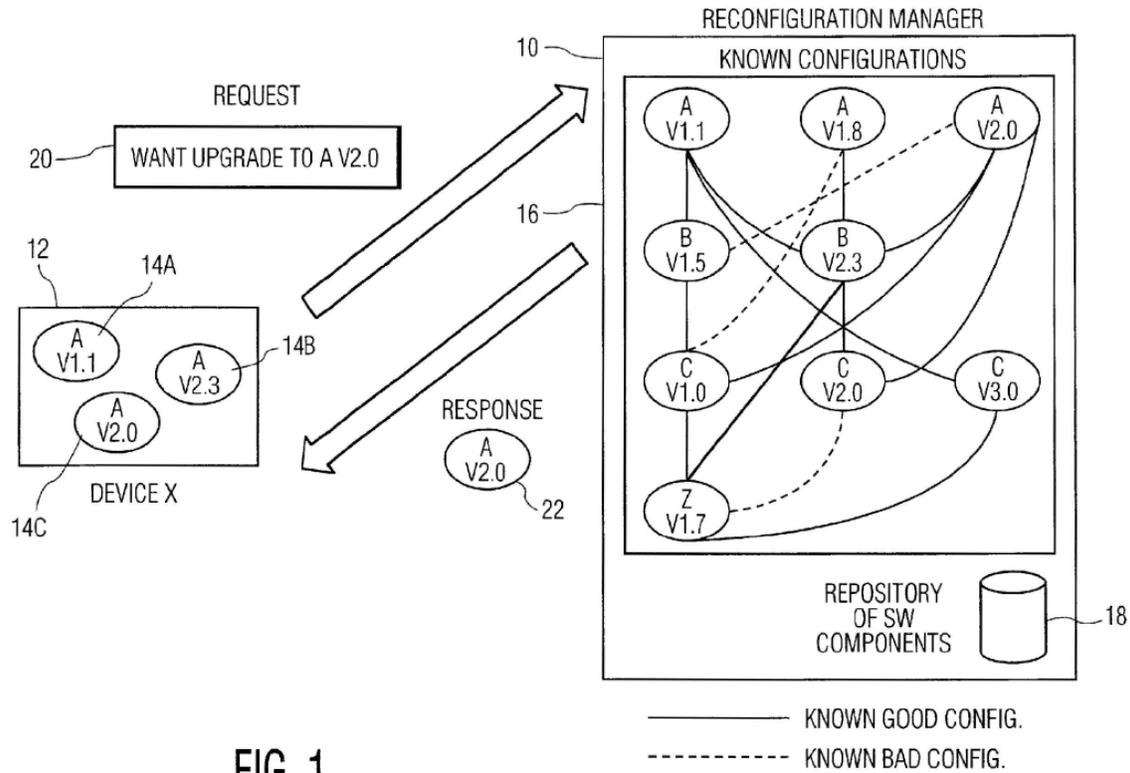


FIG. 1

Figure 1, above, illustrates a reconfiguration manager 10 interacting with an electronic device 12, also referred to as "Device X." *Id.* at 3:14–16. When reconfiguration manager 10 receives a request for an upgrade from Device X, the request indicates that the device wants to upgrade to version 2.0 of software component A and includes a list of the components currently on the device, i.e., version 1.1 of component A, version 2.0 of component C, and version 2.3 of component B. *Id.* at 4:12–19. Reconfiguration manager 10 processes the request, and if appropriate, delivers the requested version 2.0 of software component A. *Id.* at 4:22–26. Processing the request involves generating a potential upgrade configuration that will satisfy the received request, and searching through a set of known "bad" configurations. *Id.* at 4:62–66. A known "bad" configuration is indicated in

Figure 1 as a dashed line between components that are not compatible. *Id.* at 3:58–61. For example, the pair including version 1.8 of component A and version 1.0 of component C is an example of a known bad configuration. *Id.* at 3:61–63.

If the upgrade configuration corresponds to a bad configuration, the reconfiguration manager attempts to find a set or sets of potential upgrade configurations from a set of known “good” configurations. *Id.* at 4:67–5:3. A known “good” configuration is indicated in Figure 1 by a solid line between a given pair of components indicating that the components work well together or are otherwise compatible. *Id.* at 3:52–55.

C. Illustrative Claim

Petitioner challenges claims 1–4, 6–14, and 16–21 of the ’088 patent. Pet. 2. The ’088 patent recites three independent claims: 1, 11, and 21. Challenged claim 1, reproduced below, is illustrative of the recited subject matter:

1. A processor-implemented method for controlling the reconfiguration of an electronic device, the method comprising the steps of:
 - receiving information representative of a reconfiguration request relating to the electronic device;
 - determining at least one device component required to implement the reconfiguration request;
 - comparing the determined component and information specifying at least one additional component currently implemented in the electronic device with at least one of a list of known acceptable configurations for the electronic device and a list of known unacceptable configurations for the electronic device; and

generating information indicative of an approval or a denial of the reconfiguration request based at least in part on the result of the comparing step.

Ex. 1001, 6:43–59. We refer to the steps of claim 1 as the receiving step, the determining step, the comparing step, and the generating step, respectively.

D. Asserted Prior Art

Petitioner relies on the following references as prior art (Pet. 3):

- 1) *Apfel*: U.S. Patent No. 5,974,454, filed as Exhibit 1004;
- 2) *Lillich*: U.S. Patent No. 5,613,101, filed as Exhibit 1005;
- 3) *Todd*: U.S. Patent No. 5,867,714, filed as Exhibit 1006; and
- 4) *Pedrizetti*: U.S. Patent No. 6,151,708, filed as Exhibit 1007.

E. Asserted Grounds of Unpatentability

Petitioner asserts the following grounds of unpatentability (Pet. 3–4):

Claims Challenged	35 U.S.C. §	References
1–4, 6–14, 16–21	§ 103	Apfel, Lillich, Todd
9, 19	§ 103	Apfel, Lillich, Todd, Pedrizetti
1–3, 9–13, 19–21	§ 103	Apfel, Lillich
1, 3, 4, 6–11, 13, 14, 16–21	§ 103	Apfel, Todd

Petitioner further relies on the Declaration of John Villasenor to support its patentability challenge. Ex. 1003 (“Villasenor Declaration”).

II. ANALYSIS

A. Level of Ordinary Skill in the Art

Petitioner contends that a person of ordinary skill in the art at the time of invention of the '088 patent “would have had a Bachelor’s Degree in Electrical Engineering, Computer Science, or a related subject, and one or more years of experience working with configuring hardware and software components in electronic devices” and that “[l]ess work experience may be compensated by a higher level of education, such as a Master’s Degree, and vice versa.” Pet. 16–17 (citing Villasenor Decl. ¶¶ 31–34). At this stage, Patent Owner “does not dispute Petitioner’s definition of a POSITA.” Prelim. Resp. 13. Given the lack of dispute, and for purposes of this decision, we adopt Petitioner’s definition of the level of ordinary skill in the art, as it is consistent with the level of skill in the art reflected in the prior art of record. *See Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001).

B. Claim Construction

In an *inter partes* review proceeding, a claim of a patent is construed using the same standard used in federal district court, including construing the claim in accordance with the ordinary and customary meaning of the claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent. 37 C.F.R. § 42.100(b) (2019). The Petition addresses three claim terms/phrases. Pet. 18–25. Two of the proposed constructions (“list” and “known . . . for the electronic device”) are derived from our preliminary claim construction in the Decision Denying Institution in the Apple IPR. *See* Ex. 1012, 7–8. Patent Owner “requests that the Board adopt the ordinary and customary meaning of the claim term as

understood by one of ordinary skill in the art.” Prelim. Resp. 15. None of Patent Owner’s arguments point out disagreement with Petitioner’s proposed constructions, and, upon review of those proposed constructions, at this juncture, we find them reasonable. For purposes of this Decision, we adopt Petitioner’s proposed constructions as follows:

“list”	any stored representation of information indicative of component compatibility
“known . . . for the electronic device”	The term “known” means “previously determined” and the phrase “for the electronic device” does not require that the “list” of known acceptable and known unacceptable configurations be specific to the electronic device
“at least one of”	To be read in accordance with the plain meaning of the phrase, namely as conjunctive lists of their respective elements, i.e., “at least one of [a] and at least one of [b].”

Further, on this record, we determine that no other claim terms require an express construction to resolve the issues presented by the patentability challenges. *See Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (holding that only terms “that are in controversy” must be construed and “only to the extent necessary to resolve the controversy”).

C. Overview of the Prior Art

1. Overview of Apfel

Apfel is concerned with installing and updating a software program module component. Ex. 1004, Abstract. In particular, Apfel describes a

system for automatically updating a software program module component stored on a computer, as shown in Figure 3 reproduced below.

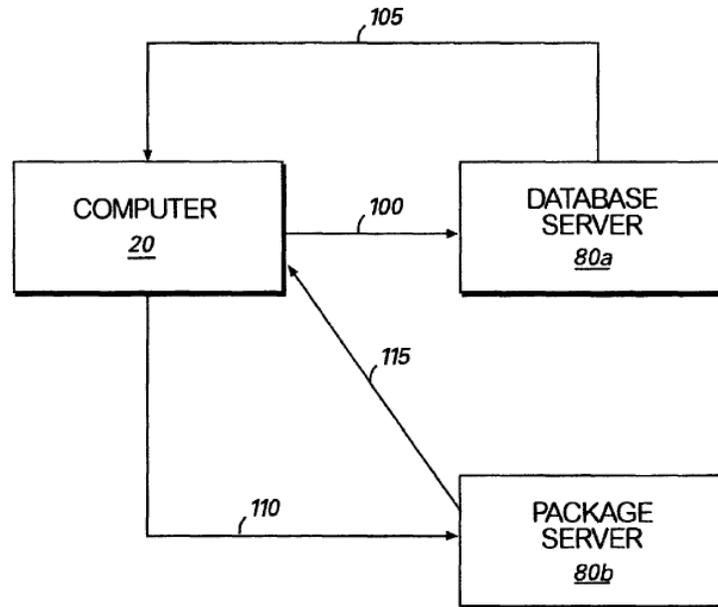


Figure 3 of Apfel (above) illustrates a system including computer 20, database server 80a, and package server 80b. *Id.* at 6:36–37. Computer 20 sends query 100 on or after a predetermined date over the Internet to database server 80a. *Id.* at 6:39–40. Query 100 includes all of the information regarding computer 20 that the database server 80a needs to determine if an upgrade is available and, if an upgrade is available, to determine the location of the upgrade package. *Id.* at 6:49–53. After reviewing query 100, database server 80a returns response 105 over the Internet to computer 20. *Id.* at 6:54–55. If an upgrade is available, then database server 80a will send back response 105 that includes the URL of the upgrade package. *Id.* at 6:63–65. After computer 20 receives response 105 including the URL of the upgrade package, computer 20 will send query

110 to package server 80b at the URL of the update package. *Id.* at 7:4–8. Package server 80b will send update package 115 to computer 20, and computer 20 will then install the update package 115. *Id.* at 7:8–10. The servers are responsible for assessing whether an upgrade is available and whether it should be downloaded based on the information sent by computer 20. *Id.* at 7:13–16.

2. Overview of Lillich

Lillich is titled “Method and Apparatus for Determining at Execution Compatibility Among Client and Provider Components Where Provider Version Linked With Client May Differ From Provider Version Available at Execution.” Ex. 1005, code [54]. Lillich describes a method and apparatus for verifying compatibility between components of a system, which share a client-provider relationship. *Id.* at Abstract. Figure 1 of Lillich is reproduced below.

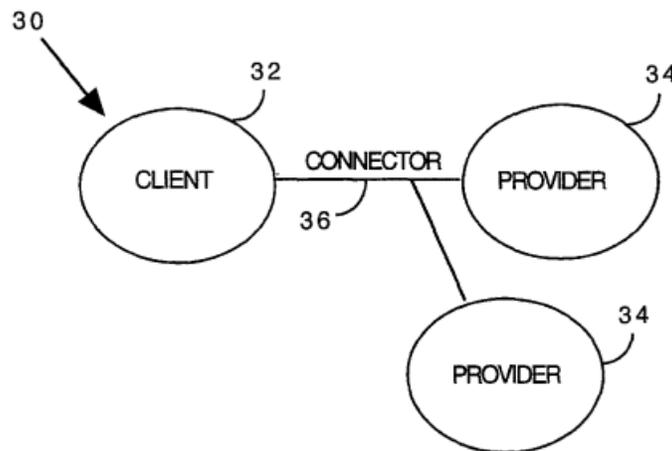


FIG. 1

Figure 1 of Lillich (above) is a symbolic, simplified block diagram of system 30. *Id.* at 5:28. As shown in Figure 1, system 30 may be hardware,

software, or a combination thereof, and includes client 32 and provider 34 connected by connector 36 such that information from the client and provider can be compared. *Id.* at 6:7–10. A provider indicator, a current indicator of a provider, and a compatibility range are defined for each of a client and a provider. *Id.* at 4:5–7. A provider indicator identifies a particular type of provider. *Id.* at 4:7–8. A current indicator of a provider specifies the version of the provider. *Id.* at 4:8–9. When a client is linked with a version of a provider, the current indicator of that provider is stored in the executable client produced, thereby identifying the version of the provider, referred to as a “definition provider,” used to build the client. *Id.* at 4:9–13. Lillich describes that the compatibility range for the client identifies the range of versions of the provider, which can be used to execute the client, i.e., which have an implementation which is compatible with the definitions supplied by the definition provider. *Id.* at 4:13–16. The compatibility range for the client specifies the oldest version of the provider that can be used to execute the client. *Id.* at 4:17–19. The compatibility range for the provider identifies the range of versions of the provider that could be used to build a client capable of operating with the current version of the provider. *Id.* at 4:19–22. The compatibility range for the provider specifies the oldest version of the provider that could have been used to build a compatible client. *Id.* at 4:22–25. The versions within each of the two compatibility ranges are older than or equal in age to the current version. *Id.* at 4:25–27. Lillich defines an “implementation provider” as “the provider which will be used to execute the client.” *Id.* at 4:28–31.

Lillich further describes a connector for connecting, at runtime, the client to the implementation provider to determine compatibility between the client and the implementation provider. *Id.* at 4:28–32. Compatibility checks are performed between a client and available versions of the provider(s), implementation providers, with which it has been linked. *Id.* at 4:32–35.

3. *Overview of Todd*

Todd is titled “System and Method for Distributing Configuration-Dependent Software Revisions to a Computer System.” Ex. 1006, code [54]. Todd describes a system for detecting and avoiding faults stemming from conflicts in hardware and/or software configurations in a computer system. *Id.* at Abstract. Figure 1 of Todd is reproduced below.

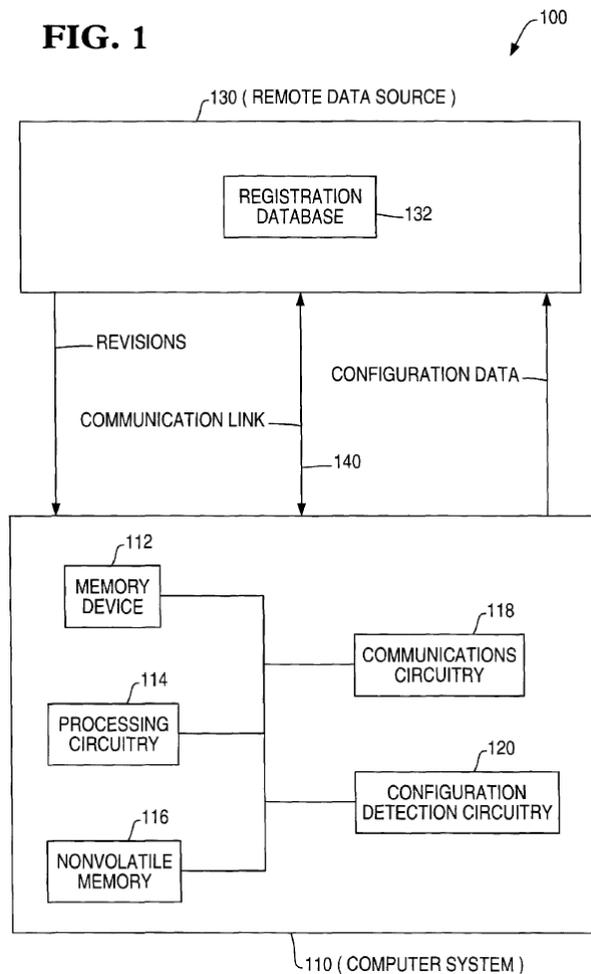


Figure 1 of Todd (above) is a block diagram of system 100. *Id.* at 5:28. As shown in Figure 1, system 100 comprises computer system 110 and remote data source 130. Computer system 110 comprises memory device 112 to store current configuration data pertaining to the hardware and software configuration of the computer system 110. *Id.* at 6:7–9. Computer system 110 also comprises processing circuitry 114, nonvolatile memory 116, and communications circuitry 118. *Id.* at 5:66–6:41. Computer system 110 also comprises configuration detection circuitry 120 responsible for obtaining data pertaining to at least a portion of the current hardware and

software configuration of the computer system 110. *Id.* at 6:42–46. Todd describes that the configuration detection circuitry 120 may be as simple as a software program, executable within the computer system 110, for querying the user as to the current hardware and software configuration. *Id.* at 6:46–49. However, Todd describes that configuration detection circuitry 120 determines the hardware and software configuration automatically, by polling hardware components and cataloging software components to create a list of current configuration data setting forth the components that comprise the computer system 110. *Id.* at 6:50–55. As shown in Figure 1 of Todd, remote data source 130 contains a database of software revisions that may be communicated to computer system 110 as a function of the current configuration data transmitted from computer system 110 to remote data source 130 and diagnostic and analytic processes within the remote data source 130 that analyze the current configuration data to identify conflicts. *Id.* at 12:1–8.

D. Obviousness Ground Based on Apfel, Lillich, and Todd

Petitioner asserts that claims 1–4, 6–14, and 16–21 would have been obvious over the combined teachings of Apfel, Lillich, and Todd. Pet. 26–64. Patent Owner challenges Petitioner’s assertions by arguing, inter alia, that “there exists no motivation within Apfel to be combined with either the teachings of Lillich or Todd.” Prelim. Resp. 20.

1. Analysis

Petitioner contends that Apfel teaches or suggests all the limitations of the independent claims, including the comparing step. *See* Pet. 41–42,

45–46 (arguing that Apfel inherently describes or implicitly teaches the comparing step with regard to the known acceptable configurations, and that Apfel shows obviousness with regard to the known unacceptable configurations). Petitioner further argues that Lillich teaches the comparing step with regard to the known acceptable configurations (*id.* at 42–45), and that Todd teaches the comparing step with regard to the known unacceptable configurations (*id.* at 46–48). As for motivation to combine these references, Petitioner’s arguments can be summarized as follows:

Apfel and Lillich

- a. It would have been obvious to combine the teachings of Apfel and Lillich because a person of ordinary skill in the art would recognize the benefit of ensuring that a requested update will not render the system inaccurate or inoperative. Pet. 27–28.
- b. Further, it would have been obvious to modify Apfel’s database lookup to include the Lillich evaluation of the version numbers of the components in Apfel (such as the Web Authoring Components program module, the HTML converter, and the word processor program module) “to determine whether they are within an appropriate range to ensure compatibility between and among those components and any requested update or combination of updates to one or more of these component versions.” *Id.* at 44–45 (arguing also that the combination would have been nothing more than the application of a well-known known technique for determining compatibility as

evidenced by Apfel's desire to perform a lookup for available upgrades that *should* be downloaded).

Apfel and Todd

- a. It would have been obvious to incorporate the teachings of Todd into Apfel because the combination provides a person of ordinary skill in the art with information about methods to perform network-based updates in a manner that ensures compatibility. *Id.* at 29–30.
- b. It would have been obvious to modify Apfel's server assessment using Todd's described conflict determination so that program updates that would result in known conflicts would not be downloaded. *Id.* at 48 (arguing also that Todd's technique is one of a limited number of solutions for determining incompatibility).

Apfel, Lillich and Todd

- a. A person of ordinary skill in the art would recognize that combining the affirmative approach emphasized in Lillich (verifying compatibility) with the safeguards presented in Todd (identifying conflicts) would result in an improved system as compared with an approach that performed only one of these checks. *Id.* at 30.
- b. The methods described in Lillich and Todd comprise well-known techniques that a person of ordinary skill in the art could have combined with Apfel's database lookup without undue

experimentation, and with a reasonable likelihood of success.

Id. at 31.

With regard to Petitioner's contentions that Apfel alone renders obvious the challenged independent claims, at this juncture, we find reasonable Petitioner's arguments, which appear to be factually supported by the current record. As the Petition sets forth, regarding the recited receiving step, Apfel's database server receives from a computer a request to update the Web Authoring Components program module of Word 8.0 (*id.* at 35–37), and, regarding the recited determining step, Apfel performs a database lookup to determine if an upgrade package is available (*id.* at 37–38). Apfel's request for the update is an HTTP query that includes the information Apfel uses to make the determination: the version of the Web Authoring Components program module to be upgraded, the version of an HTML converter in the word processor program module, the version of the word processor program module, the localization language, and the type of operating system on the computer. *See id.* at 40 (citing Ex. 1004, 8:39–46, 8:53–66).

As to the comparing step, Apfel also discloses that the database stores the upgrade package information and the corresponding configurations that should be downloaded, and that different update packages are available for different version combinations, different operating systems, and different languages. Ex. 1004, 9:36–40. Therefore, at this juncture, it seems reasonable, as Petitioner argues, that Apfel's database lookup performs a *comparison* of the information in the HTTP query with the information

stored in the database, which, Petitioner argues, inherently discloses the list of known acceptable configurations for the electronic device. *See* Pet. 42. It further seems reasonable, as Petitioner argues, that, because Apfel denies the upgrade “if the upgrade is somehow incompatible with computer 20,” it would have been obvious to a person of ordinary skill in the art that Apfel’s database lookup (i.e., comparison) determines the incompatibility by using information previously received about configurations that are not compatible. *See* Pet. 45–46. After the determination whether to allow or deny the upgrade, regarding the recited generating step, Apfel sends an UPDATE message if the upgrade is available or sends a NOUPDATE message if the upgrade is denied. *See id.* at 49–50.

As for Petitioner’s additional obviousness contention that the combination of Apfel, Lillich, and Todd teaches the comparing step, we also find that, on the present record, Petitioner’s arguments are factually supported and have merit. In particular, Petitioner asserts that Lillich teaches the comparison with “known acceptable configurations” because Lillich uses a compatibility range that allows it to assess whether the configuration of the provider software is compatible with the client software. Pet. 42–45 (citing Ex. 1005, 1:41–44, 3:66–4:27; Villasenor Decl. ¶¶ 86–89). Todd, according to Petitioner, teaches the comparison with “known unacceptable configurations” because Todd, like Apfel, uses a database to identify incompatibility of conflicts between configurations. *Id.* at 46–48 (citing Ex. 1006, 3:15–30, 3:43–51, 3:55–59, 5:8–13, 12:9–15, 14:17–20; Villasenor Decl. ¶¶ 92–95). As for reasons to combine, as stated

above, Petitioner explains why a person of ordinary skill in the art would have combined Apfel with the teachings of either Lillich or Todd. Petitioner also presents reasons to combine Apfel with both Lillich and Todd. Petitioner's proffered rationales appear reasonable at this stage of the proceeding.

Patent Owner challenges the reasons to combine. We address each of Patent Owner's challenges in turn.

Arguments Targeting Apfel's Teachings

First, Patent Owner argues that Apfel's upgrade requests are fulfilled without regard to compatibility issues, and, therefore, there is no reason to attribute to Apfel the desire to perform a software upgrade based on compatibility. Prelim. Resp. 17–20. That is, Patent Owner argues, there exists no motivation *within Apfel* to combine its teachings with those of either Lillich or Todd. *Id.* at 20. We are not persuaded by this argument. As Petitioner notes, Apfel determines, for each configuration of the computer, which upgrade package to allow (Pet. 41–42)¹ and Apfel evaluates whether, even if an upgrade is available, the upgrade “is somehow incompatible” with the computer and should not be downloaded (Pet. 45–46).² Thus, Petitioner sufficiently shows that Apfel is concerned with basing its determination whether to allow a software upgrade based on compatibility of two types: whether the software component is compatible

¹ The Petition cites Exhibit 1004 (6:65–67, 9:30–42) and Villasenor Declaration (¶¶ 83–85).

² The Petition cites Exhibit 1004 (7:13–19) and Villasenor Declaration (¶¶ 90, 91).

with the computer's configuration or whether the software component is incompatible with the computer's configuration. Accordingly, Patent Owner's argument regarding Apfel's teachings as lacking disclosure of compatibility is unpersuasive.

Second, Patent Owner argues that Apfel should be construed to teach that the user or administrator, who has general knowledge about incompatibility, would perform "a manual action," rather than the system determining the upgrade compatibility. Prelim. Resp. 19. This argument is unpersuasive. The passages in which Apfel describes the determination regarding whether the software is compatible with the computer refer to the actions performed by Apfel's server, not an administrator. Ex. 1004, 7:11–19. We do not see Apfel, nor does Patent Owner point to any persuasive disclosure in the reference, as teaching a manual action of any kind.

Arguments Targeting Lillich's Teachings

Patent Owner argues that Lillich's compatibility-verification technique "is only applicable to a client program component and a provider program component when executed locally on the memory of one computer." Prelim. Resp. 22. Patent Owner also argues that the request for a software upgrade would not work as the claims require because the "compatibility could only be verified after the client program component is downloaded in response to the request, and executed locally on the host computer." *Id.* at 23. According to Patent Owner, "combining a request for a software upgrade as taught by Apfel would destroy the functionality of the

compatibility verification technique as taught by Lillich.” *Id.* We are not persuaded by Patent Owner’s arguments.

The asserted combination of teachings, as we understand the Petition, involves modifying Apfel’s database lookup to include the comparison of Lillich’s version numbers for the various components that Apfel utilizes to determine compatibility. Pet. 44–45. Thus, the combination relies on Apfel’s server utilizing Lillich’s compatibility-check technique. In other words, the comparison using Lillich’s technique occurs at Apfel’s server. Therefore, it is irrelevant, for purpose of the asserted combination, that Lillich teaches performing its technique at a local computer.

Further, because the relevant Lillich teaching is that of the compatibility check, it is also unpersuasive to argue that Lillich’s programs are already downloaded and executed. Apfel provides the teachings of the servers receiving the upgrade requests, checking the compatibility, and servicing the request. *See, e.g.*, Ex. 1004, 7:13–16. We see no evidence in the record that Apfel would not work as intended or be incompatible with Lillich’s compatibility-check technique. Patent Owner’s arguments to the contrary focus on aspects of Lillich that are not part of the asserted combination. Nor has Patent Owner supported any of its arguments with facts tending to show that the teachings of Lillich could not be combined with Apfel’s teachings, as Petitioner asserts.

Arguments Targeting Todd’s Teachings

Patent Owner argues that Todd’s conflict check occurs after the computer has been reconfigured, contrary to the recited order of steps. Prelim. Resp. 28–29. Patent Owner contends that, “even if the software

upgrade request of Apfel was to be incorporated into the teachings of Todd, the resulting combination could not arrive at the recitations of Claims 1, 11, and 21.” *Id.* at 29. The argument is not persuasive.

As with the arguments addressed above, Patent Owner focuses on a combination of teachings that Petitioner did not make. Petitioner relies on Apfel’s server receiving the upgrade requests, checking the compatibility, and servicing the request. Todd is merely supplying the teaching of how to perform a conflict check, which Apfel’s server would perform. Patent Owner’s arguments characterize Petitioner’s asserted combination backward: incorporating Apfel into Todd. The Petition clearly states the obviousness contention differently than argued by Patent Owner: “it would have been obvious to modify Apfel’s server assessment using stored information regarding incompatible configurations . . . regarding known conflicts . . . , as described in Todd.” Pet. 48. It is irrelevant *when* Todd performs the conflict check. The timing is regulated by Apfel’s server, which, according to Petitioner’s contention, would perform the conflict check during Apfel’s database lookup—not after the software upgrade is installed, as Patent Owner seems to argue.

Finally we address Patent Owner’s argument that the combination of teachings of Lillich and Todd would render the resulting combination inoperable for its intended purpose. Prelim. Resp. 29–31. Patent Owner combines the various arguments made above with regard to Lillich performing the compatibility check locally with the argument that Todd is a network-based system. *Id.* According to Patent Owner, because Todd’s technique is applied on a network-based system, it would not function with

the Lillich local-based technique. *Id.* We are not persuaded by this argument for the same reasons as stated above. The Apfel server would be performing the relevant techniques taught by Lillich and Todd. The disclosure of Lillich being a local check and Todd's being network-based say little to nothing about whether these techniques fail to be combinable with Apfel's database lookup process. Further, Patent Owner's arguments presume an incorporation of either Lillich's technique into Todd's system or vice-versa. *Id.* at 30–31 (arguing incompatibility of combining the system of Todd with that of Lillich). These arguments do not respond to the asserted combination of teachings, which involve Apfel's database lookup as the base teaching, and, instead focus on the bodily incorporation of features of the secondary references that are not at-issue.

In conclusion, on the present record, we do not find any of Patent Owner's arguments persuasive to show fault with Petitioner's asserted reasons to combine. Having considered Patent Owner's arguments in light of Petitioner's contentions and supported evidence of record, we are persuaded that Petitioner has demonstrated a reasonable likelihood of prevailing on its assertion that claims 1, 11, and 21 would have been obvious over Apfel alone or the combination of Apfel, Todd, and Lillich.

We are aware that this ground also involves dependent claims 2–4, 6–10, 12–14, and 16–20. Patent Owner did not raise any arguments concerning these dependent claims. Having reviewed these claims and the asserted references, and noting no arguments against institution, we do not discuss these claims further except for our Conclusion section, below.

E. Obviousness Ground Based on Apfel, Lillich, Todd, and Pedrizetti

Petitioner argues that Pedrizetti teaches the limitations recited in claims 9 and 19. Pet. 64–70. Patent Owner does not argue the patentability of claims 9 and 19 apart from their parent independent claims. Prelim. Resp. 32–33.

1. Overview of Pedrizetti

Pedrizetti is titled “Determining Program Update Availability via Set Intersection Over a Sub-Optical Pathway.” Ex. 1007, code [54]. Pedrizetti describes a method of updating software on a client computer wherein a set of software programs on the client computer is compared against a set of updates on a server computer to determine which updates are applicable and should be transferred from the server to the client. *Id.* at 1:41–45. A many-to-one mapping function (e.g. a hash function) is applied to update identifiers to generate a table of single bit entries indicating the presence of particular updates on the server. *Id.* at 1:45–48. Figure 1 of Todd is reproduced below.

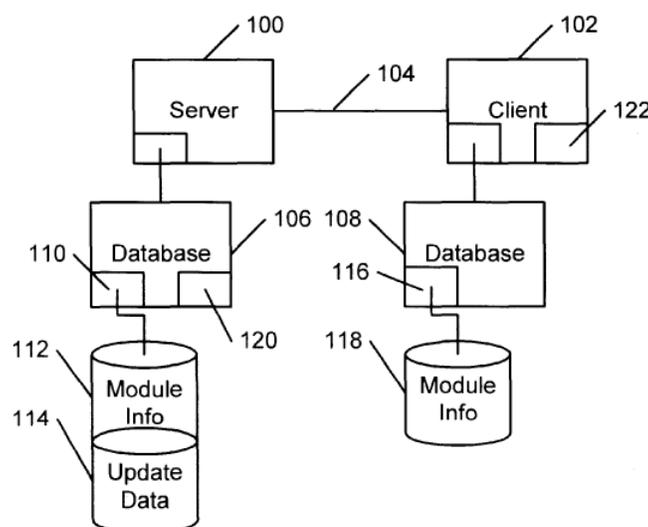


Figure 1 of Pedrizetti (above) depicts a basic configuration of a client and server. *Id.* at 2:55. As shown in Figure 1, server computer 100 is in communication with client computer 102 over a communications pathway 104. *Id.* at 2:57–58. Associated with both server 100 and client 102 are databases 106, 108, containing information regarding program modules that may be updated. *Id.* at 2:66–3:1. On the server 100 side, database 106 contains entries 110 regarding each updateable program module, where each entry contains data 112 including the module name and related tracking information. *Id.* at 3:6–9. On the client 102 side, database 108 corresponds to server database 106 in that it also contains entries 116 regarding each updateable program module, where each entry contains data 118 including the module name and related tracking information. *Id.* at 3:18–23. Client computer database 108 might only track entries for program modules corresponding to programs 122 already installed on client computer 102. *Id.* at 3:25–28.

2. *Analysis*

Claims 9 and 19 recite “wherein the reconfiguration request comprises a request for an upgrade of at least one of a software component and a hardware component of the electronic device.” Ex. 1001, 7:37–40, 8:42–45. Petitioner relies on Pedrizetti as teaching this limitation. In particular, Petitioner points out that Pedrizetti performs software updates as well as hardware updates. Pet. 67–68 (citing Ex. 1007, 1:41–65, 5:37–45, 5:60–65). Thus, Petitioner argues, it would have been obvious to include in Apfel’s request an upgrade request for an associated hardware component because “it is common for software and hardware components to work in a

complementary manner in electronic devices.” *Id.* at 69 (citing Villasenor Decl. ¶¶ 138–139). Other than to note that the arguments presented with respect to claims 1 and 11 are equally applicable to claims 9 and 19, by virtue of their dependence, Patent Owner does not address this ground in particular. Prelim. Resp. 32–33.

Having reviewed Petitioner’s arguments and evidence relied upon, we determine that Petitioner has demonstrated a reasonable likelihood of prevailing on its assertion that claims 9 and 19 would have been obvious over Apfel and Pedrizetti or Apfel, Lillich, Todd, and Pedrizetti.

F. Obviousness Grounds Based on Apfel and Lillich and Apfel and Todd

Petitioner argues that claims 1–3, 9–13, and 19–21 would have been obvious over Apfel and Lillich. Pet. 70–71. Petitioner also argues that claims 1, 3, 4, 6–11, 13, 14, and 16–21 would have been obvious over Apfel and Todd. *Id.* at 71–72. Petitioner presents these grounds as alternate grounds that broaden the claim limitations concerning the words “at least one of [a] or [b]” to require a comparison with *either* a list of known acceptable configurations *or* a list of known unacceptable configurations, but not to require the comparison with both lists. Patent Owner challenges these alternate grounds based on the same arguments discussed above, which, at this juncture, we have found unpersuasive. Prelim. Resp. 33–35.

Having determined that, for the above-discussed grounds, Petitioner has met the reasonable likelihood threshold, and considering that in those grounds we have applied a claim interpretation that is narrower than the interpretation presented in these grounds, we find that we do not need to

address these alternate grounds. Rather, we adopt, to the extent applicable here, our stated analysis of Apfel, Lillich, and Todd, and determine that these alternate grounds also present a reasonable likelihood of prevailing.

G. Patent Owner's Request for Discretionary Denial of the Petition

Patent Owner asks that the Board deny the Petition because “the Board previously denied institution in IPR2019-00056” (“IPR 56”) and because of the allegedly “weak justifications provided for presenting three grounds on each of the independent claims.” Prelim. Resp. 12. Patent Owner presents no other reason or analysis why we should exercise discretion. *Id.* Although Patent Owner cites only to *Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. 2131, 2140 (2016), we understand the request to invoke the discretionary denial under 35 U.S.C. § 314(a).

The Board’s precedential decision in *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 (PTAB Sept. 6, 2017) (designated precedential in relevant part) set forth a series of factors to be considered by the Board in evaluating whether to exercise discretion under 35 U.S.C. § 314(a) to deny a petition that challenges a patent that was previously challenged before the Board. *General Plastic*, Paper 19 at 15–16. These factors are:

1. whether the same petitioner previously filed a petition directed to the same claims of the same patent;
2. whether at the time of filing of the first petition the petitioner knew of the prior art asserted in the second petition or should have known of it;
3. whether at the time of filing of the second petition the petitioner already received the patent owner’s preliminary

response to the first petition or received the Board's decision on whether to institute review in the first petition;

4. the length of time that elapsed between the time the petitioner learned of the prior art asserted in the second petition and the filing of the second petition;

5. whether the petitioner provides adequate explanation for the time elapsed between the filings of multiple petitions directed to the same claims of the same patent;

6. the finite resources of the Board; and

7. the requirement under 35 U.S.C. § 316(a)(11) to issue a final determination not later than 1 year after the date on which the Director notices institution of review.

Id. These factors are “a non-exhaustive list,” and “additional factors may arise in other cases for consideration, where appropriate.” *Id.* at 16, 18.

The “follow-on petitions” addressed in *General Plastic* were filed by the same petitioner, after a first set of petitions against the same patents was denied on the merits. *General Plastic*, Paper 19 at 2–3. The Petition here is not a “follow-on petition,” as that term is used in *General Plastic*; there was no prior petition challenging the '088 patent filed by Petitioner, Microsoft. The prior petition challenging the '088 patent was filed by Apple, not Microsoft. And there is no evidence of cooperation between Microsoft and Apple in either the underlying litigation or the instant proceeding. Furthermore, the serial-filing concerns addressed in *General Plastic* are not involved in this proceeding because Patent Owner did not file suit against Microsoft (May 20, 2019) until *after* the Board issued the Decision Denying Institution in IPR 56 (April 29, 2019). Pet. 6 (citing Exs. 1008, 1012). Thus, the record shows that Microsoft had no need to be aware of IPR 56 or

Patent Owner's patent infringement suits involving the patent-at-issue until it was served with the complaint (May 22, 2019),³ at which time Microsoft alleges diligence in preparing and filing the instant Petition (October 11, 2019). *Id.* This factor alone weighs strongly in favor of not exercising discretion.

Nevertheless, we have considered the other *General Plastic* factors and determine that either they are not applicable or do not favor denial of the Petition. The challenges in IPR 56 rely on different prior art than the Petition here. Moreover, the three challenges against each independent claim are based on combinations of the same three prior art references, but based on an alternative claim construction positions, and thus, we find that the Board's resources are not unduly taxed by the Petition.

In sum, the prior petition in IPR 56 was independently filed by a different party having no overlap or cooperation with the Petitioner here, and there is no showing of undue delay or abusive filing of the Petition here that would warrant denial of the Petition.

III. CONCLUSION

We have determined that Petitioner demonstrated a reasonable likelihood of prevailing on its assertion that independent claims 1, 11, and 21 would have been obvious over Apfel alone or the combination of Apfel, Lillich, and Todd applying, for purposes of this Decision, the narrow claim construction of "at least one of [a] or [b]." *See* Section II.B. None of Patent

³ *See* also Exhibit 1015.

Owner’s arguments challenging institution were persuasive. And we declined to exercise discretion to deny the Petition under 35 U.S.C. § 314(a).

Therefore, we institute trial on all challenged grounds and all claims as asserted (listed below). *See SAS Inst., Inc. v. Iancu*, 138 S. Ct. 1348, 1355 (2018).

Claims Challenged	35 U.S.C. §	References
1–4, 6–14, 16–21	§ 103	Apfel, Lillich, Todd
9, 19	§ 103	Apfel, Lillich, Todd, Pedrizetti
1–3, 9–13, 19–21	§ 103	Apfel, Lillich
1, 3, 4, 6–11, 13, 14, 16–21	§ 103	Apfel, Todd

Although we have adopted certain claim constructions for purposes of this Decision, our determination is preliminary in nature—not a final determination of the construction of any claim term or phrase. Further, our determination in this Decision is not a final determination on the patentability of any challenged claims and, thus, leaves undecided any factual issues necessary to determine whether sufficient evidence supports Petitioner’s contentions by a preponderance of the evidence in the final written decision. *See TriVascular, Inc. v. Samuels*, 812 F.3d 1056, 1068 (Fed. Cir. 2016) (noting that “there is a significant difference between a petitioner’s burden to establish a ‘reasonable likelihood of success’ at institution, and actually proving invalidity by a preponderance of the evidence at trial”) (quoting 35 U.S.C. § 314(a) and comparing § 316(e)).

IV. ORDER

Accordingly, it is hereby:

ORDERED that pursuant to 35 U.S.C. § 314(a) the Petition is granted;
and

FURTHER ORDERED that pursuant to 35 U.S.C. § 314(a), *inter partes* review of the '088 patent is hereby instituted with trial commencing on the entry date of this decision, and pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4, notice is hereby given of the institution of review.

IPR2020-00023
Patent 6,467,088 B1

PETITIONER:

Derrick W. Toddy (Lead Counsel)
Andrew M. Mason
John M. Lunsford
Todd M. Siegel
KLARQUIST SPARKMAN, LLP
derrick.toddy@klarquist.com
andrew.mason@klarquist.com
john.lunsford@klarquist.com
todd.siegel@klarquist.com

PATENT OWNER:

Ryan Loveless (Lead Counsel)
Brett Mangrum
James Ethereidge
Jeffrey Huang
ETHERIDGE LAW GROUP
ryan@etheridgelaw.com
brett@etheridgelaw.com
jim@etheridgelaw.com
jeff@etheridgelaw.com