

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner,

v.

ERICSSON AB,
Patent Owner.

Case No. IPR2025-00003

Patent No. 9,313,178

PETITIONER'S REPLY TO PATENT OWNER'S RESPONSE

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	ARGUMENT	3
A.	PO’s Multiple CEK Argument	3
1.	PO Mischaracterizes the Petition and Attacks Strawmen	5
2.	PO Attempts to Infuse a Specific Key Request Requirement to [1A]’s “Media Playback Request” Limitation	8
3.	PO Misreads Peterka	10
a.	PO Misreads Fig. 7B	11
b.	PO Misreads Peterka’s “Pull” Method for Keys	15
c.	PO Misreads Peterka’s Pay-By-Time Model	16
4.	PO Misreads Suzuki	17
B.	PO’s “Content Encryption Key Identifiers” Argument	20
C.	PO’s Rehashed “Detecting” Argument	23
D.	PO’s “Issuing Requests” Arguments	25

TABLE OF AUTHORITIES

CASES

<i>Continental Circuits LLC v. Intel Corp.</i> , 915 F. 3d 788 (Fed. Cir. 2019)	9, 21
--	-------

LISTING OF EXHIBITS

Exhibit	Description
1001	U.S. Patent No. 9,313,178 (“178 Patent”)
1002	Prosecution History of U.S. Patent No. 9,313,178 (“178 FH”)
1003	Declaration of Anthony J. Wechselberger (“Wechselberger”)
1004	Curriculum Vitae of Anthony J. Wechselberger (“Wechselberger-CV”)
1005	U.S. Patent Publication No. 2002/0172368 (“Peterka”)
1006	Certified Translation of Japanese Published Patent Application No. 2004-166153 (“Suzuki”)
1007	U.S. Patent Publication No. 2010/0235528 (“Bocharov”)
1008	U.S. Patent Publication No. 2008/0270308 (“Peterka308”)
1009	European Patent Publication No. 1,418,756 A2 (“Chen”)
1010	U.S. Patent Publication No. 2005/0138362 (“Kelly”)
1011	U.S. Patent No. 8,677,428 (“Lewis”)
1012	U.S. Patent No. 8,789,196 (“Pestoni”)
1013	International Publication No. WO 2010/107625 A2 (“Sood”)
1014	U.S. Patent No. 8,806,050 (“Chen050”)
1015	T. Stockhammer, “Dynamic adaptive streaming over HTTP: Standards and design principles,” in Proc. ACM MMSys’11, San Jose, CA, USA, Feb. 23–25, 2011, pp. 133–144 (“Stockhammer”)
1016	W. Van Lancker, D. Van Deursen, E. Mannens and R. Van de Walle, “HTTP adaptive streaming with Media Fragment URIs,” 2011 IEEE International Conference on Multimedia and Expo, Barcelona, Spain, 2011, pp. 1-6, doi: 10.1109/ICME.2011.6012149 (“Van Lancker”)
1017	3GPP TS 26.234 version 9.3.0 Release 9 (2010-06), Technical Specification, “Universal Mobile Telecommunications Systems (UMTS); LTE; Transparent end-to-end Packet Streaming Services (PSS); Protocols and codecs,” Reference RTS/TSGS-0426234v930 (June 2010) (“3GPP-AHS”)
1018	R. Pantos (Ed.), HTTP Live Streaming, draft-pantos-http-live-streaming-06 (Mar. 31, 2011) (“Pantos”)
1019	RFC793: Transmission Control Protocol (September 1981) (“RFC793”)
1020	RFC2616: Hypertext Transfer Protocol – HTTP/1.1 (June 1999) (“RFC2616”)

1021	The Secure Sockets Layer Protocol, V. 3.0, <i>available at</i> https://web.archive.org/web/19970614041044/http://home.netscape.com/eng/ssl3/ssl-toc.html and https://web.archive.org/web/19970617034012/http://home.netscape.com/eng/ssl3/3-SPEC.HTM#1 (“SSL-Standard”)
1022	U.S. Patent No. 8,756,673 (“Barrus”)
1023	Yuan et al., “Layered access control for MPEG-4 FGS video,” Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429), Barcelona, Spain, doi: 10.1109/ICIP.2003.1247012 (“Yuan”)
1024	U.S. Patent App. Pub. No. 2005/0060316 (“Kamath”)
1025	RESERVED
1026	<i>Apple Inc. v. Ericsson AB</i> , IPR2022-00618, Paper 2 (Feb. 25, 2022)
1027	<i>Apple Inc. v. Ericsson AB</i> , IPR2022-00618, Paper 9 (July 8, 2022)
1028	<i>Apple Inc. v. Ericsson AB</i> , IPR2022-00618, Paper 11 (Sept. 1, 2022)
1029	<i>Apple Inc. v. Ericsson AB</i> , IPR2022-00618, Paper 13 (Nov. 28, 2022)
1030	<i>Apple Inc. v. Ericsson AB</i> , IPR2022-00618, Paper 14 (Nov. 28, 2022)
1031	<i>Apple Inc. v. Ericsson AB</i> , IPR2022-00618, Paper 17 (Dec. 30, 2022)
1032	<i>In the Matter of Certain Electronic Devices, Including Mobile Phones, Tablets, Laptops, Components Thereof, and Products Containing the Same</i> , Inv. No. 337-TA-1376, Order No. 25 (Mar. 13, 2024)
1033	<i>In the Matter of Certain Electronic Devices, Including Mobile Phones, Tablets, Laptops, Components Thereof, and Products Containing the Same</i> , Inv. No. 337-TA-1376, Complaint, Exhibit 24 (October 12, 2023)
1034	<i>Telefonaktiebolaget LM Ericsson et al. v. Lenovo</i> , 5:23-cv-00570, Order Granting Motion to Stay (E.D.N.C. Feb. 9, 2024)
1035	USPTO Binding Interim Guidance Addressing PTAB’s Approach to Discretionary Denials, June 21, 2022, <i>available at</i> https://www.uspto.gov/sites/default/files/documents/interim_proc_discretionary_denials_aia_parallel_district_court_litigation_memo_20220621.pdf .
1036	<i>In the Matter of Certain Electronic Devices, Including Mobile Phones, Tablets, Laptops, Components Thereof, and Products Containing the Same</i> , Inv. No. 337-TA-1376, Final Initial Determination (Nov. 15, 2024)
1037	<i>In the Matter of Certain Electronic Devices, Including Mobile Phones, Tablets, Laptops, Components Thereof, and Products Containing the Same</i> , Inv. No. 337-TA-1376, Commission’s Determination Affirming ALJ’s Final Initial Determination (Jan. 16, 2025)

1038	Declaration of Turhan F. Sarwar in Support of Admission <i>Pro Hac Vice</i>
1039	Declaration of Elizabeth A. DiMarco in Support of Admission <i>Pro Hac Vice</i>
1040	Deposition Transcript of John Villasenor Ph.D. (August 21, 2025) (“Villasenor-Dep.”)
1041	Supplemental Declaration of Anthony J. Wechselberger (“Wechselberger-Supp.”)
1042	U.S. Patent No. 7,392,381 (“Traw”)
1043	USPTO Memorandum Re: PTAB Consideration of Prior Findings of Fact and Conclusions of Law (September 16, 2025)

I. INTRODUCTION

Independent claims 1 and 16 of the '178 Patent (EX-1001) recite content encryption keys (CEKs) for decryption of secured content. The recited CEKs are acquired at two stages: (1) in response to a media playback request, when playback is first being initiated (limitations [1A]-[1B]), and (2) during continued playback in response to ongoing requests issued to a license server (limitation [1E]).

To meet these features, the Petition (Paper 3, "Pet.") relied on Peterka's disclosure of a system in which a client (1) acquires secure content using "a pay-by-time purchase mechanism," and (2) acquires content keys to decrypt that content during playback using "a 'pull' key distribution method where clients monitor key expiration times to request new keys before they expire." Pet., 18. Petitioner explained how such a system rendered obvious most of the Challenged Claims (in Ground 1), including independent claims 1 and 16 (Pet., 19-37, 48-50).

Among Petitioner's showings was a detailed explanation of two different ways in which Peterka's client detected key rotation boundaries—(1) by monitoring key expiration times (to know when to request new keys) and (2) by monitoring key change signals in the content (to know when to apply new keys). Pet., 31-34.

In response, PO offers four flawed arguments:

First, using Peterka's disclosures regarding purchase mechanisms ("pay-per-view," "subscription") that Petitioner *never relied on* and the "provisioning" of

other key types (“free preview key,” “group key”) that Petitioner also *never relied on*, PO alleges that multiple CEKs would not have been provided in response to one client-initiated request for program content in Petitioner’s Peterka-based combinations. POR, 6-37. As discussed *infra* § II.A, PO’s argument is nonresponsive. The Petition demonstrated that Peterka *expressly* discloses sending multiple content keys in response to a client-initiated playback request (*e.g.*, Peterka, [0080]-[0082]) *and* that this was desirable/obvious. Pet., 19-22, 25-26. The ITC agreed that Peterka met the multiple-CEK requirement. EX-1036, 58-59; EX-1043.

Second, PO alleges that Petitioner’s Peterka-based combinations do not meet the claims’ “content encryption key identifiers” (“CEK IDs”) requirement because, according to PO, CEK IDs must be specifically “assigned” to the CEK and must be “unique” amongst other CEKs stored in some unclaimed “history.” POR, 37-43. As discussed *infra* § II.B, this is a flawed attempt to import unrecited requirements into the claims that are in any event met by Grounds 1-2 and 4-5.

Third, PO rehashes its failed POPR argument alleging that one of Peterka’s two CEK rotation boundary detection methods—monitoring when CEKs expire—does not meet the claims’ “detecting” limitation because “detecting” *requires* “checking for an explicit key-change notification.” POR, 43-51. As discussed *infra* § II.C, the Board twice rejected this argument, and absent evidence that “detecting” is narrower than its plain and ordinary meaning, should do so again.

Fourth, PO alleges that the second of Peterka’s two CEK rotation boundary detection methods—monitoring the content stream for a signal informing the client that a new key needs to be used for decryption—would not also meet the claims’ “issuing requests” limitation because the claims purportedly *require* the “detecting” step to occur before the “issuing requests” step. POR, 52-55. As discussed *infra* § II.D, PO argued the opposite position before the ITC, *which agreed* with PO that no such order was required. EX-1032, 21-25. The ITC also *agreed* with Petitioner’s showing (undisputed by PO) that a client using both Peterka’s second CEK rotation boundary detection method (Peterka, [0116]-[0121]) and Peterka’s “pull” method for key distribution (Peterka, [0105]-[0106], [0110]) meets the “detect[ing]” and “issu[ing] requests” requirements. EX-1036, 56, 59. None of PO’s arguments justify reaching a different decision. EX-1043.

II. ARGUMENT

A. PO’s Multiple CEK Argument

Petitioner showed that limitation [1A]’s “initiating a media playback request” requirement is met when Peterka’s client requests access to media content, e.g., to an ongoing show/event. Pet. 19-20 (citing Peterka, [0080]-[0082], [0091]-[0092], [0141]-[0143]; EX-1001, 2:15-21, 9:49-52). Among Peterka’s supporting disclosures were *express* descriptions of client-initiated “requests *for a program*” (Peterka, [0091]-[0092]), “client...*request[s]*” for “*program content*” (Peterka,

[0141]-[0143]), and client-initiated key “*negotiations*” resulting in the *receipt of multicast content* (Peterka, [0080]-[0082]).¹

Petitioner also showed that limitation [1A]’s “receiving a playback request response” requirement is met because “Peterka teaches distributing multiple keys (e.g., current and next) to a client when it negotiates access to media content.” Pet., 20-22. Among Peterka’s supporting disclosures were (1) Peterka’s express description of a “key management approach” where clients are given keys to both the current and next segments of media to mitigate server load at segment boundaries (Peterka, [0080]) and (2) an illustrated example (*id.*, FIG. 7B) corroborating that this key management approach—sending multiple keys to a client in response to a client’s content access request—was applicable and *desirable* in scenarios where many such requests from many clients were expected at segment boundaries. Wechselberger, ¶¶77-79. Relying on Peterka’s description of individual keys being sent in an entitlement control message (ECM) or entitlement management message (EMM)—Peterka, [0104]-[0106]—the Petition explained that the ECMs/EMMs sent in response to a particular media playback request by the client collectively constitute a “playback request response” received by the client, as recited in [1A]. Pet., 22; Wechselberger, ¶¶80-81.

¹ Emphasis added unless otherwise specified.

The parties agree the claims do not require multiple keys transmitted in a “single message” (e.g., a single ECM or EMM). Pet., 22; POR, 32 (“Patent Owner has not relied on such an interpretation of ‘response.’”). Instead, PO alleges—based on purchasing mechanisms and key types in Peterka that Petitioner *did not rely upon*—that “Peterka’s ‘pull’ embodiment does not teach providing multiple ‘content keys’ in response to one client-initiated request for program content.” POR, 6-37. PO’s arguments fail for numerous reasons discussed below, and provide no basis to depart from the ITC’s prior adjudication that Peterka teaches a playback request response with multiple CEKs. EX-1036, 58-59; EX-1043.

1. PO Mischaracterizes the Petition and Attacks Strawmen

The POR offers **32** pages of argument regarding whether Peterka teaches multiple CEKs in a playback request response—**2** of which mischaracterize the Petition’s theories (POR, 7-9), and **20** of which (POR, 9-23, 26-27, 32-34) are devoted to attacking arguments that Petitioner did not make.

For example, PO asserts that “the Petition relies exclusively on Peterka’s ‘pull’ model for” limitation [1A]’s “‘initiating a media playback request’ requirement.” POR, 7 (citing Pet., 12, 18). This is wrong. The Petition argued that “initiating a media playback request” is met when Peterka’s client *requests access to media content*. Pet. 19-20 (citing Peterka, [0080]-[0082], [0091]-[0092], [0141]-[0143]; EX-1001, 2:15-21, 9:49-52). Peterka’s disclosures regarding requesting

access to *media content* are distinct from Peterka’s “pull” model because Peterka’s “pull” model refers to *key* distribution (Peterka, [0105]). Neither Peterka nor the Petition uses the word “pull” for *content*. PO’s arguments focusing on Peterka’s pull model for acquiring keys are thus nonresponsive to Petitioner’s mapping of [1A]’s “media playback request.”

PO also argues that Peterka’s *pay-per-view* (PPV) and *subscription* purchase mechanisms do not include providing clients multiple content keys. POR, 10-14, 32-34. Here again, PO’s arguments are nonresponsive. The Petition unambiguously relied on Peterka’s *pay-by-time* (PBT) purchase mechanism, *and did not* rely on PPV or subscription. Pet., 18. Unlike PPV and subscription, Peterka’s PBT mechanism involves rotating content keys during a program to encrypt/decrypt its different segments. Peterka, [0080]-[0082]. PO’s attempts to distract the Board from Peterka’s key rotation teachings by focusing on irrelevant Peterka mechanisms *that work differently* should be rejected.

PO also argues that Peterka’s teachings regarding certain “*free preview key*” implementations of Peterka’s “free preview period” (POR, 15-17) and “*group key*” implementations of Peterka’s “initial viewing period” only involve keys provided to a client “ahead of time,” allegedly before any media playback request, during “*provisioning*” (POR, 17-23, 26-27). These arguments focus on key types and concepts that Petitioner *did not rely on*. As PO (and PO’s expert) acknowledged, the

Petition *did not rely on* the “free preview key” or “group key.” POR, 8; Villasenor-Dec. (EX-2010), ¶¶33-34, 40, 58, 65; Villasenor-Dep. (EX-1040), 51:5-59:18, 65:6-13. The Petition instead relied on Peterka’s teachings regarding Peterka’s “*content keys*.” Pet., 23-25. Thus, PO’s arguments about *other* key types are irrelevant to the Petition’s grounds.

The Petition also did not rely on any implementations of Peterka where “content keys” are “provisioned,” including because Peterka *never even discloses* such an option. Peterka, [0104] (Table 1); *see infra* § II.A.3.a. PO’s arguments regarding provisioning are likewise irrelevant.

PO also ignores the import (and misinterprets Petitioner’s use) of Fig. 7B. According to PO, Petitioner “relies on Figures 7A and 7B *as disclosing* the multiple ‘content encryption keys’ requirement of the challenged claims.” POR, 27. While Fig. 7B certainly *does* disclose distributing multiple CEKs in response to each request for program content that occurs during Fig. 7B’s “free preview” (CK0, CK1) and “initial key distribution” periods (CK1, CK2) (Pet., 21),² PO fails to address Petitioner’s actual argument. The Petition used Fig. 7B to “*confirm[]*” that the “key management approach” of “distributing multiple keys (e.g., current and next) to a client when it negotiates access to media content” was both “applicable” and

² PO disputes this with a tortured reading of Fig. 7B, discussed *infra* § II.A.3.a.

“*desirable* in any distribution scheme in which key requests from many clients are expected at segment boundaries.” Pet., 20-22 (citing Wechselberger, ¶79: “Providing a client with a content key for the next segment in response to a client’s request to begin playback during the current segment as taught in Peterka was desirable (and indeed obvious), since this would predictably reduce the volume of requests that the server needed to respond to ahead of the upcoming segment boundary and help reduce disruptions in content distribution due to latency caused by any network congestion that arises from many clients using the network to transmit requests.”). None of PO’s arguments about what Peterka discloses rebut Petitioner’s showing that providing multiple CEKs in response to a client request in Peterka’s PBT context was *desirable/obvious*.

PO’s devotion of nearly two-thirds of its argument regarding multiple CEKs to arguments Petitioner never presented speaks volumes about PO’s inability to respond to the Petition’s actual trial grounds—which stand unrebutted.

2. PO Attempts to Infuse a Specific Key Request Requirement to [1A]’s “Media Playback Request” Limitation

PO repeatedly assumes [1A]’s “media playback request” limitation requires a specific request for a content key and attempts to distinguish Peterka on this basis. *E.g.*, POR, 16 (attempting to distinguish Peterka because “[t]o access the free preview, Peterka teaches that the client receives any necessary content key *without making any ‘request’ for that content key.*”), 21 (“*At this stage there has been no*

client-initiated content key request, and thus there has been no ‘pull.’”); see also *id.*, 16 (“*unrequested content key...*”), 17 (alleging a “*first unrequested content key*” and alleging that “Peterka *provides the content key...without request*”), 19 (“...Peterka also gives clients the *content key...without request*”), 20-21 (“*Content Key 1 (CK1)*...is provided to Peterka’s clients *without request*”), 22 (“*unrequested content key...*”).

Even if PO’s characterizations of Peterka were accurate—they are not (see *infra* § II.A.3)—PO’s suggestion that limitation [1A] requires a specific request for a content key should be rejected.

PO does not allege lexicography or disavowal. *Continental Circuits LLC v. Intel Corp.*, 915 F. 3d 788, 796-97 (Fed. Cir. 2019).

Limitation [1A] only requires a client to “initiat[e] a media playback request and receiv[e] a playback request response.” Limitation [1A] ***does not require*** the media playback request ***to include any specific data***, e.g., a specific request for a CEK. As PO’s own expert noted, while the media playback request “certainly ***can***” include such a specific request, all that is required is “a request for you to be playing back the media.” Villasenor-Dep., 39:22-40:9, 41:6-16.

None of the other limitations in the Challenged Claims require limitation [1A]’s “media playback request” to include a specific request for CEKs. Limitation [1B] requires [1A]’s playback request ***response***—i.e., “what comes back as a result

of the request”—to include content encryption keys, but does not specify what “forms” [1A]’s media playback *request* must take. Villasenor-Dep., 36:24-37:10, 39:22-40:9, 40:18-25. And Limitation [1E] separately requires a client to issue specific “requests” to a license server to retrieve a second content encryption key. Limitation [1E]’s “requests” are not the same as [1A]’s “media playback request.”

As the Petition explained, the claim encompasses precisely what is taught in Peterka: a client that initiates a media playback request and receives multiple content keys in response, and then pulls additional content keys from a license server during playback. Pet., 19-22, 27-28, 34-36. Because the “initiating a media playback request” requirement in limitation [1A] *does not require* any explicit format or “request” for a content key, PO’s argument that Peterka can be distinguished because it does not disclose a media playback request that includes a specific request for a content key should be rejected.

3. PO Misreads Peterka

As discussed *supra* §§ II.A.1-II.A.2, PO’s arguments either mischaracterize and ignore Petitioner’s actual arguments, or argue about requirements not recited in the claims. PO’s arguments also *separately* fail because PO’s allegedly “close analysis of Peterka’s teachings” (POR, 9)—which PO uses to suggest that Peterka does not teach providing clients with more than one content key in response to a request for access to media content—is deeply flawed. As discussed below, PO

misunderstands Fig. 7B’s teachings, Peterka’s “pull” embodiment for key distribution, and Peterka’s pay-by-time purchasing model.

a. PO Misreads Fig. 7B

PO’s argument that Fig. 7B and its “associated teachings cannot demonstrate unpatentability” is based on a fatal misreading of Fig. 7B. POR, 27.

PO repeatedly argues that Peterka’s client in Fig. 7B receives content keys CK0 and CK1 *without any request*. E.g., POR, 1 (“...Peterka expressly teaches that the first two content keys (CK0 and CK1) in Figure 7B are provided *without any request at all*... Content Key 2 (CK2) is the one and *only* key that clients obtain through request...” (emphasis original), 26 (“In Fig. 7B, neither the first ECM carrying Content Key 0 (CK0) encrypted under a free preview key (FPK), nor the second ECM carrying Content Key 1 (CK1) encrypted under a group key (GK), was provided in response to any *request*.... Rather,...both content keys were provided ‘ahead of time,’ before any client-initiated request for content.... The first *requested* content key in Figure 7B is Content Key 2 (CK2).”) (emphasis original). PO is simply wrong.

None of the purportedly “extensive teachings” (POR, 26) that PO relies upon to allege that Fig. 7B’s CK0 and CK1 are exclusively “provision[ed]” “ahead of time” actually relate to CK0 or CK1. Instead, Peterka’s teachings regarding providing keys “ahead of time” (e.g., during “provisioning”) exclusively relate to

Peterka’s “*free preview keys*” (Peterka, [0083], [0085], [0103], [0104]) and “*group keys*” (Peterka, [0102], [0104], [0127]-[0129])—which Petitioner did not rely upon (*see supra* § II.A.1).

Peterka never once discusses “provisioning” keys “ahead of time” in the context CK0 and CK1. *See* Peterka, [0122]-[0125], [0130]. Peterka does not even identify “provisioning” as an option for “content keys.” Peterka, [0104] (Table 1) (listing “provisioned” for only FPK and GK); *see also id.*, [0083] (discussing Fig. 3 and provisioning for a “free preview”), [0085] (discussing Fig. 4 and provisioning a “free preview key”), [0102]-[0103] (discussing provisioning a “group key” and “free preview key”), [0127]-[0129] (discussing provisioning a “group key” in the context of Figs. 15, 16A-16B).

Fig. 7B itself also contradicts PO’s premise that CK0 and CK1 were provided to the client prior to any client requests for the program. In Peterka’s own words, Fig. 7B “show[s] exemplary distribution of cryptographic keys during portions of a program.” Peterka, [0021]. Fig. 7B is reproduced below.

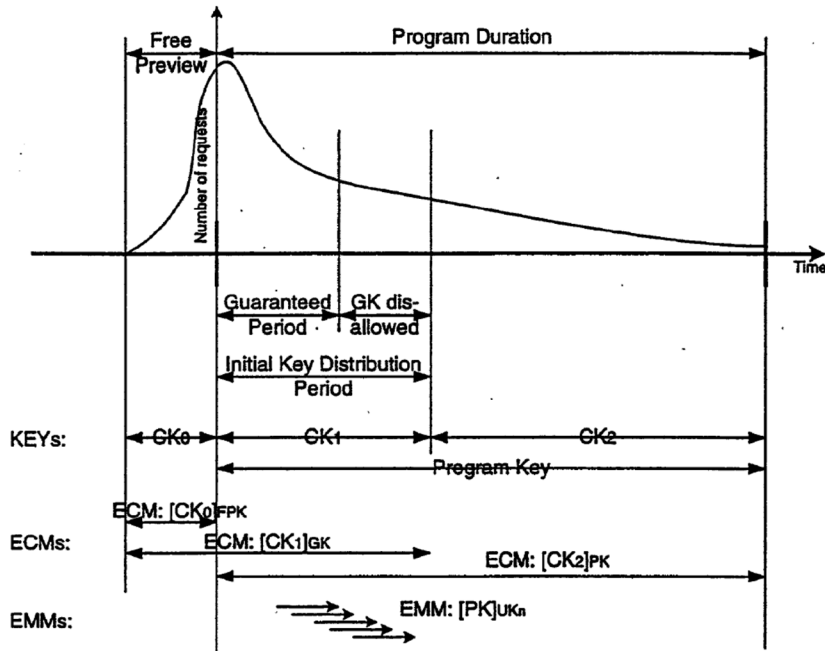


Fig. 7B

The top half of Fig. 7B shows the “number of requests” for a program that Peterka’s system receives during portions of a program, plotted against time. Peterka, [0091] (describing the same plot in Fig. 7A: “FIG. 7A illustrates a plot showing the number of requests for a program versus the program duration.”); Villasenor-Dep., 104:8-13 (comparing Fig. 7B with 7A: “I think it’s intended to show the same general sort of behavior.”). Below the “number of requests” plot, Fig. 7B shows the ECMs that the system “convey[s]” to clients during the same program portions. Peterka, [0122].

Fig. 7B thus shows, at a given time, the number of requests for a program (above the time axis), and the ECMs / content keys that would be sent in response (below the time axis). Pet., 21; Wechselberger, ¶79. For example, during the “free

preview period,” clients are sent an ECM containing CK0 (encrypted under a free preview key FPK) and an ECM containing CK1 (encrypted under a group key GK). Pet., 21; Wechselberger, ¶79; *see also* Peterka, [0122].

As can be seen above, Fig. 7B ***does not show*** any distribution of keys that occurs ***before*** clients begin requesting access to a program, because—as PO’s own expert acknowledged—the only time in Fig. 7B where there have been no client requests to access a program is before the beginning of Fig. 7B’s free preview period (left side of the graph). Villasenor-Dep., 105:18-106:7.

The fact that Fig. 7B does not teach content keys (CK0, CK1) provisioned to clients ***before*** any request to access the program (as PO argues) is confirmed by Peterka’s description in the paragraphs directly after paragraph [0122]. In paragraphs [0123]-[0124], Peterka describes ***alternatives*** to the embodiment shown in Fig. 7B that do not use CK0 and CK1 and instead uses the FPK and GK. Peterka, [0123] (“***Alternatively***, CK0 does not need to be distributed inside an ECM. The value of ***FPK already possessed by the plurality of clients*** may be taken as CK0.”), [0124] (“***Alternatively***, CK1 does not need to be distributed inside an ECM. The value of ***GK already possessed by the plurality of clients*** may be taken as CK1.”). As discussed, Peterka only references provisioning for the FPK and GK, which the Petition ***did not rely upon***.

PO's reading of Fig. 7B contradicts Peterka and is unsupported. As the Petition established, Fig. 7B teaches distribution of keys occurring in response to client requests, and this *confirms* the applicability and desirability of distributing multiple keys in response to a client request for media. Pet., 20-22; Wechselberger, ¶¶78-79. To the extent PO's Fig. 7B argument is responsive, it is based on a false premise: CK0 and CK1 are *not* exclusively provisioned to clients ahead of time, as PO argues.

b. PO Misreads Peterka's "Pull" Method for Keys

PO's argument that Peterka does not disclose providing multiple content keys in response to a client-initiated request to access content is *also* based on repeatedly conflating Peterka's client-initiated requests to access content (under a PBT model) with Peterka's "pull" requests for keys. *E.g.*, POR, 1 (referencing "client-initiated 'pull' request *for content*."), 11 (referencing "'pull' request *for pay-per-view content*"), 12 (referencing "client-initiated 'pull' request *for content*"); *see also* POR, 7 ("...relies exclusively on Peterka's 'pull' model for key requests for the 'initiating a media playback request' requirement."), 9 ("Peterka's '*pull*' *embodiment does not teach* providing multiple 'content keys' in response *to one client-initiated request for program content*."). This is an independent reason that PO's argument should be rejected.

As the Petition explained in detail, Peterka discloses both receiving keys in response to a request to access media content (Pet., 19-22, 27-28) and then “pulling” keys on an ongoing basis during playback (Pet., 34-36). In Peterka and in the Petition, reference to the “pull” model refers to **key** distribution (Peterka, [0105])—which, e.g., occurs **after** a request to access media content has been initiated and the first set of content keys have already been negotiated/received (Peterka, [0080]). Neither Peterka nor the Petition ever uses the word “pull” for **content**.

PO’s conflation of these two concepts echoes PO’s repeated attempts to add unrecited requirements to limitation [1A]—which does not require a specific request for a content key (*see supra* § II.A.2)—and underlies PO’s strained conclusion that Peterka’s client never receives more than one content key when negotiating access to a media stream. *Cf.* Peterka, [0080]-[0082]. This misunderstanding infects PO’s entire multiple CEK argument, and for this separate reason, the argument fails.

c. PO Misreads Peterka’s Pay-By-Time Model

PO also argues that “giv[ing] certain clients access to additional future content that they did not purchase” “would defeat the purpose of Peterka’s pay-by-time” model. POR, 33-34. This argument is illogical.

First, PO’s argument ignores Peterka’s express teachings. Peterka undisputedly teaches that in the PBT purchasing model, “the client pays for the time duration of the content **that was actually watched**,” instead of what keys the client

receives. Peterka, [0053]; POR, 14. Peterka also undisputedly teaches, *in a PBT context*, providing a client the key for “the *next* segment”—i.e., for content not yet watched and not yet billed and paid-for—in order to help mitigate large server load. Peterka, [0080]. Peterka also confirms that the PBT model can work regardless of which keys are distributed to the client by expressly teaching a server that bills clients *based on the time* a client joins and leaves a multicast: “The caching server will record the *time* when each client joins and leaves the multicast *for billing purposes*.” Peterka, [0082]. These teachings all plainly point to a system in which clients can receive keys for segments that the client may not consume.

Second, PO *itself* argues that providing a client a key for a future, as-yet-unpurchased segment in the PBT scheme—to “avoid service interruptions”—“is consistent with the rest of Peterka’s teachings.” POR, 27-28.

PO’s suggestion that sending multiple keys to a PBT client somehow defeats the “purpose” of Peterka’s PBT model is both inconsistent with Peterka’s teachings and PO’s own admissions. This argument fails and should be rejected.

4. PO Misreads Suzuki

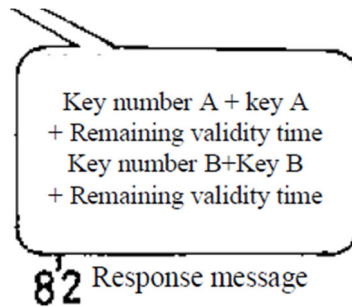
In Ground 1, the Petition cited Suzuki to show that multiple keys were conventionally transmitted in one message in response to a client request, and to support the argument that it was obvious to implement Peterka to return multiple keys in response to a client request. Pet., 22; Suzuki, [0076]-[0077], Figs. 5-6. In

response, PO says “Suzuki’s paragraphs 76 and 77 do not teach transmitting multiple ‘content encryption keys,’ they only disclose providing ‘key information relating to the key currently in use (key A) and the next key to be used (key B).” POR, 36 (citing Villasenor-Dec., ¶78). Suzuki’s multiple-key teachings had previously been *undisputed*. EX-1036, 49-50. PO now misreads Suzuki.

First, Suzuki unambiguously uses the term “key information” to mean *both* “the key” itself “and associated information” such as key number and remaining validity time. *E.g.*, Suzuki, [0012] (“key information (*the key and associated information*)”); *see also id.*, [0050], [0119].

Second, Suzuki, [0076]-[0077] disclose sending, in response to a client request, a single “response message 82 having key information relating to the key currently in use (key A) and the next key to be used (key B).” *See also* Wechselberger, ¶81.

Third, notwithstanding PO’s failure to acknowledge it, the Petition cited to Suzuki, [0076]-[0077] and Figs. 5-6. Suzuki, [0077] itself also references Fig. 5. Fig. 5 *expressly shows* multiple keys (Key A, Key B) transmitted in a single response message 82, along with the keys’ key numbers and remaining validity times. Villasenor-Dec., 146:14-19. Fig. 5 is excerpted below.



See also Suzuki, [0072] (“FIG. 5 is a diagram specifically showing...the contents of each packet/request/message. In FIG. 5, the key currently in use is indicated as key A, the next key to be used is indicated as key B, and their key numbers are indicated as ‘key number A’ and ‘key number B.’”). Fig. 6 shows the same information in an operation sequence. Suzuki, Fig. 6, [0073].

PO’s expert had no credible explanation for PO’s misunderstanding, suggesting (wrongly) that Petitioner failed to cite relevant portions of Suzuki and that these teachings were inconsistent with Suzuki’s other teachings. Villasenor-Dep., 143:14-148:22. PO’s expert is wrong. Petitioner cited Figs. 5-6. Pet., 22; Wechselberger, ¶81. And Suzuki’s above teachings (including in Figs. 5-6) are consistent with Suzuki’s other teachings. *E.g.*, Suzuki, [0057] (discussing multiple-key response), [0060] (similar). Had PO or PO’s expert carefully “reviewed the entirety of Suzuki” (Villasenor-Dep., 148:16-17), there would be no dispute about Suzuki’s teachings.

PO’s only other argument regarding Suzuki is about whether Suzuki motivated a POSA to provide Peterka’s clients with unpurchased content for

segments. POR, 36-37. This is irrelevant. Petitioner cited Suzuki to show multiple-key responses were conventional. And as discussed *supra* § II.A.3.c, PO’s “unpurchased content” premise misreads Peterka.

PO’s errors leave Petitioner’s use of Suzuki in Ground 1 to help demonstrate obviousness un rebutted, and are separately fatal to PO’s multiple CEK argument.

B. PO’s “Content Encryption Key Identifiers” Argument

PO’s second argument is that Petitioner’s Peterka-based grounds do not meet the “content encryption key identifiers” term because it “imposes two” “requirements”—(1) that the identifier is “*unique*” in the context of “*all*” content encryption keys maintained in a “*history*” and (2) that the identifier must be “*assigned*” to the content encryption key *itself*.” POR, 37-43. These purported “requirements” appear *nowhere in the challenged claims*. And PO agreed the claims included no such requirements before the ITC. EX-1032, 15. PO’s attempt to import unrecited requirements should be rejected:

First, the claims do not recite any requirement that CEK IDs be “assigned” to a CEK, or that they be sufficiently “unique” to distinguish between keys stored in a “history.” The claims recite no “history.”

Second, PO does not establish any basis to depart from the plain and ordinary meaning of “content encryption key identifier.” PO does not allege lexicography or disavowal—let alone establish either with the relevant legal analysis. The *one*

sentence in the specification that PO alleges “imposes two requirements for [CEK IDs]” (EX-1001, 3:28-32) neither defines the term nor contains “expressions of manifest exclusion or restriction.” *Continental Circuits*, 915 F.3d at 796-97. Instead, that sentence describes how *one* embodiment of a server-side “*workflow manager*” works—an entity “distinct from the client” that is undisputedly unclaimed. Villaseñor-Dep., 132:17-133:4.

Third, PO has no response to Petitioner’s showing that using an ECM’s “sequence number” or “time stamp” as a CEK identifier closely tracks the ’178 specification’s teaching of using “content encryption key identifiers” “selected based on a segment number of the prepared content”—Pet., 25 (citing EX-1001, 3:34-35, 4:8-10; *see also id.*, 6:1-3, 6:17-19)—and cites no legal basis to construe the term inconsistently with the specification.

Fourth, even if the claims included PO’s “requirements,” PO’s argument fails. Peterka’s “sequence number”/“time stamp” are each undisputedly “unique” enough (Pet., 25) to distinguish amongst the client’s stored keys (a “history”). Petitioner also showed there were *numerous* reasons to implement Peterka’s client to receive and use identifiers that were (1) assigned specifically to keys and (2) sufficiently “unique” to distinguish amongst keys in a “history”—and numerous reasons to combine Peterka with other references disclosing these unrecited features. Pet., 26 (citing Suzuki, Pestoni, and Wechselberger, ¶¶93-96); Pet., 51-55 (Peterka-

Suzuki, citing Wechselberger, ¶¶161-172); Pet., 66-68 (Peterka-Suzuki-Bocharov, citing Wechselberger, ¶¶201-207); Pet., 69-73 (Peterka-Bocharov-Peterka308-Chen, citing Wechselberger, ¶¶208-221); *see also* Wechselberger-Supp. (EX-1041), ¶¶274-277 (discussing key uniqueness); Wechselberger-Tr. (EX-2011), 39:7-40:13.

PO's argument fails to credibly rebut Petitioner's obviousness arguments. Regarding the obviousness of using unique CEK IDs assigned specifically to keys (in view of Suzuki and Pestoni, Pet., 26), PO's only response is to acknowledge that there is no meaningful difference between using Peterka's sequence numbers/time stamps as CEK IDs for individual keys and specifically assigning CEK IDs to individual keys. POR, 41-42. Moreover, PO does not dispute that a POSA would have been motivated to use assigned CEK IDs in the "*alternative*" scenario contemplated by the Petition (Pet., 26), where Peterka's sequence numbers/time stamps are not used as CEK IDs.

Regarding Ground 5, PO refers back to its flawed argument that Peterka does not teach sending clients more than one content key—which fails (*see supra* § II.A)—and completely ignores Petitioner's discussion of multi-bit signals embedded in Peterka's streaming media packets. Pet., 69; Peterka, [0117]-[0118].

PO's second argument should be rejected.

C. PO's Rehashed "Detecting" Argument

Petitioner showed that limitation [1D]'s "detecting" requirement was met by Peterka's client in two different ways: (1) by monitoring key expiration times so that the client can *acquire* new content keys before current content keys expire (Pet., 31-32, citing Peterka, [0105]-[0106]); and (2) by monitoring the content stream for signals that inform a client when to begin *utilizing* a given content key for decryption (Pet., 33, citing Peterka, [0116]-[0121]). Petitioner explained why Peterka's "signaling" teachings regarding key utilization were fully compatible with Peterka's pull model for acquiring keys, including by citing to Peterka's disclosure of sending clients information for *both* detection operations in one key distribution message—key lifetime (for monitoring key expirations) *and* a key parity bit (for comparing against key change signals in the stream). Pet., 34; Peterka, [0121].

PO's third argument rehashes PO's recurring "detecting" argument, which contends that Peterka's disclosures regarding monitoring key expiration times does not meet the "detecting" limitation. POR, 43-51. Petitioner addressed a similar attempt to narrow the "detecting" limitation prior to institution. Pet., 31-33; POPR Reply (Paper 10), 1-3. The Board correctly rejected PO's previous attempts to narrow "detecting" twice—in this proceeding *and* in earlier IPR proceeding IPR2022-00618, brought by Apple. Institution, 16-24; EX-1028, 20-23.

PO brings little new to the table. POR, 43-48. PO *again* tortures Figure 3 and its accompanying disclosures, which expressly describe a process for “*detecting*” CEK rotation boundaries (EX-1001, 9:47-49) that *includes* checking for whether the current key will expire. EX-1001, 11:3-8, 11:38-41, Fig. 3. PO *again* identifies nothing in the intrinsic record that justifies narrowing “detecting” from its ordinary meaning to exclude one of the detecting types in Fig. 3.

PO also mischaracterizes the Petition to try distinguishing Peterka, arguing that merely “identifying a key’s expiration time alone” cannot be “detecting” a key rotation boundary. POR, 49. But Petitioner never equated “identifying a key’s expiration time alone” with “detecting.” Instead, Petitioner showed how “*keep[ing] track of*”/“*monitor[ing]*” key expiration times meets the “detecting” described in the ’178 Patent. Pet., 31-33.

PO finally tries to argue that “detecting” requires a three-step process that includes “checking” for both an express notification and an expiration time, analyzing the resulting information, and determining the appropriate rotation boundary in view of the results.” POR, 49. Much like the POPR’s “partly hidden”/“not clear” argument, PO’s new “analyzing”/“determining” argument bears no relation to the claims, finds no support in the intrinsic record, and departs wildly from PO’s previous allegations that detecting a key rotation boundary required nothing more than “encountering” a header in a media stream. POPR Reply, 1-3;

EX-1033, 14. This is another attempt to flee the intrinsic record. PO provides no reason to depart from the Board’s well-reasoned analysis at Institution. Institution, 20-24.

D. PO’s “Issuing Requests” Arguments

PO’s fourth argument focuses on the second of Peterka’s two CEK rotation boundary detection methods—monitoring the content stream for a signal informing the client that a new key needs to be used for decryption. POR, 52-55. PO’s argument is that Peterka’s signaling method would not *also* meet the “issuing requests” limitation [1E]/[16F] because there is “no mechanism” in Peterka for a “client to reliably request and obtain the necessary ‘new content key’ *before* the client has received a signal to change keys.” POR, 52-53. PO also rehashes its POPR argument that Peterka’s key change signaling method is somehow incompatible with its pull key distribution method. POR, 54-55. As discussed below, both arguments contradict positions PO took during the ITC litigation, and in any event fail.

PO’s argument that “Peterka’s ‘pull’ model...provides no mechanism for a client to reliably request and obtain the necessary ‘new content key’ *before* the client has received a signal to change keys” (POR, 53) contradicts PO’s ITC argument that [1D]/[16E] and [1E]/[16F] did not need to occur in the recited order. Pet., 34 n.5. The ITC *agreed with PO’s* pre-POR position, finding that neither the claims nor the specification supported requiring the steps to occur in order. EX-1032, 21-25; EX-

1043. None of PO’s arguments justify departing from the ITC’s prior finding. EX-1043. And, as Petitioner explained, even if that step order were required, Peterka’s detection of content encryption boundaries by monitoring the expiration of keys satisfies that order. Pet., 34 n.5.

PO’s argument that Peterka’s “pull” model for key distribution is “incompatible with Peterka’s signaled key change method” (POR, 55; *see* POPR, 23) was abandoned by PO during the ITC litigation. POPR Reply, 5; EX-1036, 56. Regardless, as the Petition explained, the two processes are compatible. PO conflates two separate, orthogonal concepts disclosed in Peterka—one which allows a client to *acquire* new keys before they are needed for decryption, and one which allows a client to *apply* keys to decrypt a stream of content at the correct time. Peterka teaches a client that does both. Pet., 11-12 (citing Wechselberger, ¶49); Pet., 33-34 (citing Wechselberger, ¶¶115-117). And Peterka expressly describes sending clients information for *both* detection operations in one key distribution message—i.e., key lifetime (for monitoring key expirations) and a key parity bit (for comparing against key change signals in the stream). Peterka, [0121] (cited at Pet., 34).

The Board recognized the parties’ dispute regarding the compatibility of Peterka’s pull model and key-change-signaling and preliminarily agreed with Petitioner for Institution. Institution, 26-27. PO relegates this argument to a handful of paragraphs without bringing anything meaningfully new or explaining why PO

reverses itself on an issue that it conceded before the ITC. EX-1036, 56. PO misreads Peterka. The Board should find—as did the ITC—that Peterka teaches a client that pulls keys before they are needed *and* detects key change signals. *Id.*; EX-1043.

Respectfully submitted,

Date: October 3, 2025

/Turhan Sarwar/

Turhan F. Sarwar, admitted *pro hac vice*
WOLF, GREENFIELD & SACKS, P.C.

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24, the undersigned certifies that the foregoing **PETITIONER'S REPLY TO PATENT OWNER'S RESPONSE** contains 5,599 words excluding; a table of contents, a table of authorities, a certificate of service or word count, or appendix of exhibits or claim listing. Petitioner has relied on the word count feature of the word processing system used to create this paper in making this certification.

Date: October 3, 2025

/Dara Del Rosario/
Dara Del Rosario
Paralegal
WOLF, GREENFIELD & SACKS, P.C.

CERTIFICATE OF SERVICE UNDER 37 C.F.R. § 42.6 (e)(4)

I certify that on October 3, 2025, a copy of the foregoing document, including any exhibits filed therewith, is being served via electronic mail, as previously consented to by Petitioner, upon the following:

Peter C. Knops
Kayvan B. Noroozi

peter@noroozipc.com
kayvan@noroozipc.com

Date: October 3, 2025

/Dara Del Rosario/
Dara Del Rosario
Paralegal
WOLF, GREENFIELD & SACKS, P.C.