

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SLACK TECHNOLOGIES, LLC. and
SALESFORCE, INC.,
Petitioners,

v.

WRINKL, INC.,
Patent Owner

IPR2024-01490
U.S. Patent No. 11,973,731

**PETITIONERS' REPLY TO PATENT OWNER RESPONSE
FOR *INTER PARTES* REVIEW OF
U.S. PATENT NO. 11,973,731**

TABLE OF CONTENTS

	<u>Page(s)</u>
I. SUMMARY OF ARGUMENT.....	1
II. CLAIM CONSTRUCTION	1
III. IMPROPER COPYING ALLEGATIONS.....	1
IV. GROUND 1: KAKUTA ANTICIPATES CLAIMS 1–7, 9–12, 15–19, 21–22, 25–27, AND 30.....	2
A. Claim 1	2
1. 1[c]	3
2. 1[h]	4
3. 1[i]	5
4. 1[n] and 1[o].....	5
5. 1[p]	7
B. Claims 2–27.....	8
1. Claim 2.....	8
2. Claim 3.....	9
3. Claim 4.....	10
4. Claim 5.....	11
5. Claim 6.....	11
6. Claim 7.....	11
7. Claim 9.....	12
8. Claim 10.....	13
9. Claim 11	13

10.	Claim 12	13
11.	Claims 15–19, 21–22, and 25–27	14
C.	Claim 30	14
1.	30[pre]–30[b]	14
2.	30[c]	15
3.	30[d]	16
4.	30[e]	16
5.	30[f]	17
V.	GROUND 2: CLAIMS 1-7, 9-19, AND 21-30 ARE OBVIOUS OVER KAKUTA	17
A.	1[c], 15[b], and 25[b]	17
B.	1[d], 15[c], and 25[c]	18
C.	1[o], 15[n], and 25[n]	18
D.	1[i], 1[p], 3[a], 13[a], 15[h], 15[o], 17[a], 23[a], 25[h], 25[o], 27[a], and 28[a]	19
E.	Claim 4	19
F.	Claims 9 and 21	19
G.	Claim 12	20
H.	Claim 13	20
I.	Claim 14	21
J.	Claims 23–24 and 28–29	22
K.	Claim 30	22
VI.	GROUND 3: CLAIMS 1-7, 9-19, AND 21-29 ARE OBVIOUS OVER KAKUTA IN VIEW OF MATSUMOTO	23

A.	1[c], 15[b], and 25[b]	23
B.	Claims 13–14, 23–24, 28–29.....	23
C.	Remaining Claims	24
D.	Motivation to Combine	24
VII.	GROUND 4: CLAIMS 1-29 ARE OBVIOUS OVER KAKUTA IN VIEW OF MATSUMOTO AND CHANG	25
A.	Claims 1, 15, and 25	25
B.	Claim 4	26
C.	Claims 8 and 20	27
D.	Claim 12	28
E.	Remaining Claims	28
F.	Motivation to Combine	28
VIII.	GROUND 5: CLAIM 30 IS OBVIOUS OVER KAKUTA IN VIEW OF CHANG	29
A.	30[d]	29
IX.	CONCLUSION.....	30

TABLE OF AUTHORITIES

Page(s)

Cases

<i>Arendi S.A.R.L. v. Apple Inc.</i> , 832 F.3d 1355 (Fed. Cir. 2016)	19
<i>PAR Pharm., Inc. v. TWI Pharms., Inc.</i> , 773 F.3d 1186 (Fed. Cir. 2014)	18
<i>PharmaStem Therapeutics, Inc. v. ViaCell, Inc.</i> , 491 F.3d 1342 (Fed. Cir. 2007)	2
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005)	2, 8
<i>ResQNet.com, Inc. v. Lansa, Inc.</i> , 346 F.3d 1374 (Fed. Cir. 2003)	3
<i>Rosco, Inc. v. Mirror Lite Co.</i> , 304 F.3d 1373 (Fed. Cir. 2002)	18

EXHIBIT LIST

<i>Exhibit No.</i>	<i>Description</i>
Exh-1001	U.S. Patent No. 11,973,731 to Cohen
Exh-1002	File History for Application No. 18/535,572
Exh-1003	Declaration of Andrew Cockburn
Exh-1004	<i>Curriculum Vitae</i> of Andrew Cockburn
Exh-1005	U.S. Patent No. 6,630,944 (“Kakuta”)
Exh-1006	U.S. Patent No. 7,426,540 (“Matsumoto”)
Exh-1007	U.S. Patent App. Pub. No. 2013/0069969 (“Chang”)
Exh-1008	Return of Service in Related Action
Exh-1009	Complaint in Related Action
Exh-1010	Scheduling Order in Related Action
Exh-1011	Provisional Application No. 62/253,229
Exh-1012	John E. Grayson, Python and Tkinter Programming, (2000)
Exh-1013	Apple, Human Interface Guidelines: The Apple Desktop Interface (1987)
Exh-1014	Quentin Jones, et al. “Empirical Evidence of Information Overload Constraining Chat Channel Community Interactions.” In <i>Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work (CSCW ‘08)</i> , Association for Computing Machinery, New York, NY, USA, 323–332 (2008)
Exh-1015	Andrew Cockburn and Saul Greenberg, “Making Contact: Getting the Group Communicating with Groupware.” In <i>Proceedings of the conference on Organizational computing systems (COCS ‘93)</i> , Association for Computing Machinery, New York, NY, USA, 31–41 (1993)

<i>Exhibit No.</i>	<i>Description</i>
Exh-1016	Thomas F. Stafford and Dennis Gonier, <i>What Americans Like About Being Online</i> , 47 Communications of the ACM, 107, 107–12 (2004)
Exh-1017	Petitioners' Reasons for Parallel Petitions and Petition Ranking
Exh-1018	'731 Patent Claim Elements
Exh-1019	Declaration of Christopher Kao In Support of Petitioners' Unopposed Motion for Admission Pro Hac Vice of Christopher Kao
Exh-1020	Declaration of Brock Weber In Support of Petitioners' Unopposed Motion for Admission Pro Hac Vice of Brock Weber
Exh-1021	Reply Declaration of Andrew Cockburn For <i>Inter Partes</i> Review of U.S. Patent No. 11,973,731
Exh-1022	Transcript of Deposition of Kevin C. Almeroth, October 28, 2025
Exh-1023	<i>mIRC User Guide</i> (Sept. 9, 2016), available at https://web.archive.org/web/20160909085546/https://www.mirc.com/help/mirc.pdf#expand .
Exh-1024	Antony Fountain, Jeremy Huxtable, Paula Ferguson & Dan Heller, <i>Motif Programming Manual, Open Source Edition</i> , Vol. 6A (Dec. 2001), available at https://www.ist.co.uk/motif/download/6A/6A_book.pdf
Exh-1025	John E. Grayson, <i>Python and Tkinter Programming</i> 16, 154–55, 365–66 (2000).
Exh-1026	Ben Shneiderman, <i>The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations</i> (Dep't of Computer Sci., Human-Computer Interaction Lab & Inst. for Sys. Rsch., Univ. of Md., College Park, 1996), available at https://doi.ieeecomputersociety.org/10.1109/VL.1996.545307 .
Exh-1027	Jakob Nielsen, <i>Usability Engineering</i> , ch. 5 (Morgan Kaufmann Publishers Inc. 1994), available at https://dl.acm.org/doi/10.5555/2821575 .

<i>Exhibit No.</i>	<i>Description</i>
Exh-1028	<i>New Ways to Interact with Your Tablet PC</i> (Aug. 25, 2004), available at https://web.archive.org/web/20051221130655/http://www.microsoft.com/windowsxp/tabletpc/evaluation/overviews/pcinteract.mspx

I. SUMMARY OF ARGUMENT

The Petition demonstrated that Claims 1–30 of U.S. Patent No. 11,973,731 (“’731 Patent”) are unpatentable. Patent Owner’s (“PO”) Response (Paper 16, “POR”) fails to rebut this, improperly importing additional limitations from the specification, applying claim constructions that are contrary to the terms’ plain meaning, and mischaracterizing the prior art.

Kakuta alone—or with Matsumoto and Chang—discloses or renders obvious every limitation. Each reference addresses the same problem identified in the ’731 Patent—improving clarity and navigation in multi-party messaging. PO’s alleged distinctions reflect only routine, predictable GUI refinements a PHOSITA would have implemented using known techniques. The record confirms all claims are unpatentable.

II. CLAIM CONSTRUCTION

PO asserts that no claim constructions are necessary. POR, 18. Petitioners agree.

III. IMPROPER COPYING ALLEGATIONS

PO’s allegations of “copying” or “admissions” by Petitioners are improper, irrelevant, and unsupported. POR, 3–4.

The cited blog posts and press materials (Exs-2001–2002) are merely marketing statements, not technical evidence or admissions tied to the ’731 Patent’s

claims. *See PharmaStem Therapeutics, Inc. v. ViaCell, Inc.*, 491 F.3d 1342, 1351 (Fed. Cir. 2007). They reflect routine product development and design in a crowded field predating the '731 Patent. *See, e.g.*, Exhs-1005–1007.

Even if Slack acknowledged that user-friendly threads were “nontrivial,” this reflects independent development using different design choices, not copying or non-obviousness. Indeed, PO’s own evidence shows Slack began developing its *Threads* feature as early as August 19, 2014 (Exh-2003), well before Wrinkl’s October 14, 2016 priority date. IPR2024-01489, Paper 16, 12.

Regardless of PO’s disputed priority date (*see* IPR2024-01489), the '731 Patent was filed December 11, 2023, and issued April 30, 2024—years after Slack’s January 2017 *Threads* launch. Slack’s threaded messaging was independently developed, and PO offers no evidence of access, copying, or nexus to the challenged claims. *See* Exh-1022, 49:1–50:10.

IV. GROUND 1: KAKUTA ANTICIPATES CLAIMS 1–7, 9–12, 15–19, 21–22, 25–27, AND 30

A. Claim 1

PO distorts the claim language by importing unwarranted limitations from exemplary embodiments. POR, 19–36. Yet, the invention is defined by the claims, not by specific embodiments. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1323 (Fed. Cir. 2005); *see* Exh-1022, 60:15–25. PO also cites prosecution remarks from

unrelated applications, which cannot redefine the claims. *ResQNet.com, Inc. v. Lansa, Inc.*, 346 F.3d 1374, 1382 (Fed. Cir. 2003).

1. 1[c]

PO incorrectly argues that Kakuta lacks a first pane because Figure 8 allegedly shows only one channel (“#CH1”) rather than multiple identifiers. POR, 19–23. A PHOSITA would recognize Kakuta’s channel identifier as a drop-down menu displaying multiple channel options. Exh-1003, ¶¶57–60. IRC systems like Kakuta support multiple channels, and Kakuta teaches that creating multiple channels “according to topic is easy.” Exh-1005, 1:57–58; Exh-2003, ¶61.

PO attempts to recast the channel identifier as a static label or “title field,” but a PHOSITA would recognize otherwise. Exh-1021, ¶21. The arrow box next to “#CH1” in Figure 8 is a standard drop-down indicator, which a PHOSITA would recognize as a conventional GUI control for accessing multiple channel identifiers. Exh-1003, ¶59.

PO claims a drop-down menu is not a “pane,” but a PHOSITA would know “menu panes” were well known and within that scope. *See* Exh-1024, 38; Exh-1022 70:2–4. IRC clients like Kakuta accommodated varying channel lists with scrolling or drop-down menus. *See* Exh-1023, 8; Exh-1021, ¶¶14–15. Nothing in the claims excludes a drop-down pane, and PO cites no evidence to preclude it. Kakuta

expressly discloses channel identifiers for multiple user groups across channels. Exh-1005, 2:4–9.

The additional “Channel” control does not alter this analysis. Graphical interfaces often provide redundant controls to perform the same action. *See* Exh-1023, 8; Exh-1021, ¶¶19–20; *see* Exh-1022, 71:11–73:7. Kakuta itself includes up/down arrows and a right-side drop-down icon, each permitting channel navigation within the same window. *See* Exh-1005, Fig. 8; Exh-1021, ¶¶20–22.

2. 1[h]

PO argues that Kakuta lacks a “first user-selectable control” because its “Create” control is not tied to a specific message. POR, 23–28.

Kakuta’s toolbar includes selectable controls—including “Create”—that act on selected messages. Exh-1005, 8:30–47. A PHOSITA would understand that selecting messages and pressing “Create” links control to those communication(s)—precisely the claimed functional association. Exh-1021, ¶26.

PO’s contention that the toolbar is merely “generic” overlooks Kakuta’s disclosure. Kakuta shows the toolbar used after message selection. Exh-1005, Fig. 8. Highlighting a message and pressing “Create,” generates a thread from that message, demonstrating contextual association. *Id.*, 8:38–47; Exh-1021, ¶27. Kakuta discloses multiple thread-creation methods, not just sequences PO cites. Exh-1021, ¶27.

PO's argument that "association" requires some specific visual association is unfounded. The claim does not require that, and selecting or highlighting messages provides the necessary visual and functional association regardless. *Id.*, ¶28.

3. 1[i]

PO argues Kakuta lacks this feature because it uses multiple steps instead of a single-step process. POR, 28–30.

Claim 1 requires only that the third pane display "in response to" selecting the first control—not instantaneous or single-step action (i.e., no confirmation or dialog window). Exh-1021, ¶¶29–31. A PHOSITA understands that confirmation prompts are routine and do not interrupt the causal link between selecting "Create" and displaying the new pane; "in response to" simply means the pane appears as a result of that selection. *Id.*

Even if immediacy were required, Kakuta teaches that pressing "Create" after selecting a message opens thread window 307 ("third pane") with the message input field displayed as a direct result. *Id.*; Exh-1005, 8:30–47.

4. 1[n] and 1[o]

PO wrongly argues that Kakuta lacks an "anchor message," arguing that the top-most message in Kakuta's thread window is not system-designated as "special." POR, 30–32.

Kakuta teaches that selecting messages and activating “Create” carries them into a new thread window, with the first selected message serving as the “anchor message” that links the subsidiary thread to the original communication. Exh-1005, 8:38–42. A PHOSITA would understand that Kakuta anchors a subsidiary thread to a prior message by carrying the selected message into the new thread. Exh-1021, ¶¶32–33.

PO dismisses this as merely the “top-most” message, but the first message selected in Kakuta functions as the anchor by appearing at the head of the thread window, framing the conversation. Exh-1005, Fig. 7; Exh-1003, ¶85. The claim requires no additional labeling or formatting.

Kakuta shows the “anchor message” identified by message and thread IDs (e.g., messages 1, 5, and 6 under “Th1”), and PO’s objections about visibility or timing are irrelevant because the claims require no sequence and the first message’s placement visually indicates the anchor. Exh-1005, Fig. 2, 4:57–65.

PO ignores that Kakuta’s thread titles and message IDs inherently provide an “indication” that the selected message is the anchor, and a PHOSITA would understand that pressing “Display” to open a thread necessarily identifies that corresponding message as the anchor. *See id.*, 6:23–34, 9:23–31; Exh-1003, ¶¶114–16.

PO contends that Kakuta lacks a “second user-selectable control,” but Kakuta’s “Display” satisfies this element because, after a message is selected, pressing “Display” opens the corresponding thread window; its floating toolbar placement does not affect its association with the selected message, and the claims require no adjacency. Exh-1005, 9:23–31; Exh-1021, ¶40.

PO misconstrues the phrase “display a second user-selectable control *in said second pane*,” which modifies the message, not the control; the claim requires the control to be associated with that message, not embedded within the pane’s pixel boundaries. *See* Exh-1022, 90:7–92:25. But Kakuta meets this even under PO’s view because its moveable toolbar appears within the pane’s boundaries. *See* Exh-1005, Figs. 5, 8; Exh-1021, ¶41.

The “Display” control identifies the first message as the thread’s anchor, as the “Program” option corresponds to the initial “programming” message; contrary to PO’s claim, the control operates on the user-selected thread linked to that anchor message. Exh-1021, ¶¶42–43.

5. 1[p]

PO wrongly argues that Kakuta’s “Display” control fails to teach the claim element because it opens a thread-selection window instead of displaying part of the thread. POR, 34–36.

Kakuta teaches that activating “Display” opens a thread window showing the anchor message and subsidiary conversation, and a PHOSITA would understand that selecting a thread title is merely an intermediate step in displaying the thread in the third pane. Exh-1005, 9:23–31; Exh-1003, ¶¶108–10; Section IV.A.3.

PO's suggestion that thread titles like “Program” are too “cryptic” is misplaced as Kakuta links threads to message IDs so that selecting a thread reliably displays its anchor message at the top of the thread window. Exh-1005, 6:23–34; Exh-1003, ¶108–10.

B. Claims 2–27

PO presents no separate arguments for Claims 15 and 25, which are anticipated. Section IV.A.

1. Claim 2

PO misreads Kakuta, claiming it “teaches away” by displaying messages in only one window rather than both. POR, 36–38. Kakuta expressly teaches that messages appear in both the chat and thread windows. Exh-1005, 15:11–22 (“messages . . . displayed in thread window 307 are also displayed in chat window 303.”); *id.*, Cls. 15, 17. A PHOSITA would view this duplication as a routine design choice, and nothing in Kakuta limits display to one window. Exh-1021, ¶¶44–46. PO cannot rely on isolated embodiments while ignoring Kakuta's full disclosure. *Phillips*, 415 F.3d at 1323.

PO misreads Figure 14, as S15 merely shows logic for deciding where to render a message at a given moment—it is not a ban against dual display. Exh-1021, ¶49; Exh-1005, 10:65–11:3. A PHOSITA would understand it as conditional routing, not a restriction against simultaneous display. Exh-1021, ¶49.

Moreover, as described, Kakuta's administration module can send messages to both the main chat and thread window modules "in accordance with user instructions." *Id.*, ¶50; Exh-1005, 7:48–60.

2. Claim 3

PO wrongly asserts Kakuta lacks a "third user-selectable control" (POR, 40–43), but the "Link" control performs this function by retrieving and displayed linked communications in a new subsidiary window. Exh-1005, 8:59–9:4, 13:22–30; Figs. 10, 17.

PO misreads Kakuta in asserting "Link" is "not associated with particular messages;" the claim requires no positional association. Kakuta's "Link" control is functionally associated with thread messages in the main chat window and links them for display in a new thread window. Exh-1005, 8:59–9:4; Exh-1021, ¶53–57.

The "Link Process Subroutine" confirms that selecting "Link" triggers acquiring messages (S374), opening a new thread window (S375), and displaying the rearranged messages there (S376), thus displaying part of the linked thread in response to user selection. Exh-1005, Fig. 17, 13:22–30; Exh-1021, ¶56.

PO also argues that “Link” is unassociated with messages due to its floating toolbar placement, but Kakuta states that toolbar commands, including “Link,” operate on chat messages “in accordance with user instructions.” Exh-1005, 7:31–34. When a user highlights messages and activates “Link,” the system merges and redisplayes them in a new window; its function, not location, establishes the required association. Exh-1021, ¶53–57.

Kakuta further shows that thread window 307 displays copied or quoted messages from the prior conversation—“additional messaging communications” linkable through “Link”—selecting “Link” while messages are visible in the main pane displays those same communications in the thread window. *Id.*; Exh-1005, 8:30–37; Figs. 6–7.

PO’s claim that “Link” is merely a “generic list opener” is incorrect; Kakuta’s display process retrieves and renders the thread message content in a new window, not just a list. Exh-1005, 14:46–15:5; Exh-1021, ¶56.

3. Claim 4

PO claims Kakuta lacks a third control displayed “proximate” to an additional message (POR, 44–48), but a PHOSITA would understand “Link” on floating toolbar 305 can be docked or positioned near messages, and that the precise pixel distance or static placement is immaterial. *See* Exh-1005, 8:30–58, Figs. 4–7; Exh-1021, ¶¶59–60; *see* Exh-1022, 107:8–12.

4. Claim 5

PO reiterates its Claim 2 argument, but Kakuta anticipates Claim 5. Section IV.B.1; Exh-1003, ¶¶127–28. A PHOSITA would view these as subsidiary communications from user selection, and Kakuta confirms flexible display by time or user sequence. Exh-1021, ¶62; Exh-1005, 9:1–4.

5. Claim 6

PO repeats its Claim 3 arguments, but Kakuta anticipates Claim 6. Section IV.B.2. Kakuta discloses input field 309 in thread window 307 operable to receive messages. Exh-1021, ¶ 64.

6. Claim 7

PO misinterprets the claim and Kakuta. POR, 49–50. Claim 7 requires only displaying identifying information (e.g., the anchor's author), not a separate label; Kakuta shows each message with the author's name in both windows, which a PHOSITA would recognize as marking the anchor message. Exh-1005, Fig. 7, 8:30–42; Exh-1021, ¶66; *see* Exh-1022, 107:16–108:2.

The claim limits neither form nor placement, only requiring information “associated with said anchor message.” Kakuta discloses message-identifying information, and the '731 Patent's specification provides no further restriction; a PHOSITA would understand this includes Kakuta's displayed usernames. Exh-1005, 3:39–47; Exh-1021, ¶66.

PO misreads “displaying ... in association with said additional messaging communications.” POR, 50–51. The claim requires only that anchor identifying information appear with additional messages, which Kakuta satisfies by showing the author's name in the same chat window. Exh-1005, 7:19–22, 8:43–47; Exh-1021, ¶¶66.

7. Claim 9

PO misunderstands both the claim and Kakuta in asserting that it lacks a “fourth user-selectable control.” Kakuta. POR, 51–52. Claim 9 requires a control based on identifying information that, when selected, displays the third thread pane.

Kakuta teaches that users can select messages or their identifying information to create a new thread, and following IRC conventions, a PHOSITA would recognize that clicking an author's name functions as a control to open a new thread. Exh-1005, 3:43–47, 1:32–42; Exh-1021, ¶¶67–69.

Pressing “Create” performs the same function, operating on selected messages or identifying information: “a thread is created by selecting messages as a whole, or words included in messages.” Exh-1005, 8:45–47, Fig. 8; Exh-1021, ¶70.

PO's assertion that the “Create” control or user-name selection is “not a control” contradicts Kakuta and standard practice. Kakuta's “creation means” links message selection or message-identifying information to thread creation, confirming these identifiers operate as controls; under IRC conventions, selecting a username

or message (e.g., double/triple-click) alone or with “Create,” likewise triggers third pane thread display. *See* Exh-1005, 3:43–47; 8:45–47, Fig. 8; Exh-1021, ¶71; Exh-1023, 9.

The claims do not require identifying information to appear only after sidebar creation. Kakuta shows the same identifying data (e.g., username) in both the main chat and thread windows, displaying it with additional messaging communications. Exh-1005, 10:65–11:3, 15:11–22, Fig. 7; Exh-1021, ¶67. Even if “Create” precedes the sidebar field, Kakuta still discloses this behavior, as selected messages and their identifying information can appear in both panes depending on user settings. *Id.*

8. Claim 10

PO relies on its Claim 1 arguments, but Kakuta anticipates Claim 1. Section IV.A. Kakuta's thread window includes its own input field, shown with the main chat field, operable simultaneously. Exh-1005, Fig. 7.

9. Claim 11

PO relies on its Claim 1 arguments, but Kakuta teaches that thread messages may be sent to fewer than all channel members. Exh-1005, 3:56–63, 10:20–27.

10. Claim 12

PO misreads Kakuta and a PHOSITA's understanding in claiming it lacks “a request to permit a non-member user to receive messages.” POR, 52–53. Kakuta describes administrative functions managing participation and visibility for public

and private threads. Exh-1005, 3:56–63, 11:6–10, Fig. 6; Exh-1003, ¶70. Consistent with the '731 Patent's disclosure that non-members may be notified via email, a PHOSITA would understand Kakuta's web-based chat system—as inherently enabling communication to non-members. Exh-1001, 16:8–20, Exh-1005, 11:34–36, Fig. 7; Exh-1021, ¶75. A PHOSITA, familiar with IRC invite commands (e.g., “/invite”), would view Kakuta's control modules as managing non-member add/remove requests; accessing a private thread via “Display” thus constitutes a request to permit a non-member to receive messages. Exh-1023, 211–12; Exh-1021, ¶74.

11. Claims 15–19, 21–22, and 25–27

PO offers no new arguments, and for the reasons discussed above (Sections IV.A–B.10), Kakuta anticipates them.

C. Claim 30

PO wrongly argues Kakuta does not anticipate Claim 30 because “Display” does not “create” a thread and “Program” is not a valid summary indicator. POR, 53–54.

1. 30[pre]–30[b]

PO repeats its failed Claim 1 arguments. Section IV.A.

2. 30[c]

PO wrongly claims “Display” cannot satisfy this limitation because it does not literally “create” a new thread (POR, 53), but the claim requires only “receiving a thread request,” not instantaneous thread creation. A PHOSITA would understand “create,” to include instantiating a thread process—e.g., an action that generates or displays a thread window for composing or exchanging messages. Exh-1021, ¶80.

Kakuta discloses when selecting “Display,” the system opens a thread selection window, (Exh-1005, 9:23–31, Fig. 12) and, after selecting a title, invokes creation module 21 to execute the display process subroutine. *Id.*, Fig. 19, 14:46–50. Steps S431–S435 depict a multi-step creation process in which the system retrieves thread identifiers, acquires and reconstructs message data, and renders the subsidiary thread window, thereby instantiating a thread environment for continued interaction. Exh-1005, 14:51–58; Exh-1021, ¶81.

Even under a narrow reading of “create,” Kakuta satisfies this limitation. Figure 19 shows the system retrieving and populating messages (“acquires messages in the selected threads...”) and transmitting them for display, thereby functionally recreating the thread and initiating a new instance where additional messages can be composed via input field 309 instructions to the I/O module. Exh-1005, 14:64–15:9, Fig. 19 (S433–S434); Exh-1021, ¶82.

3. 30[d]

PO wrongly asserts that Kakuta's "Program" label is not a thread summary indicator or rendered proximate to a first messaging. POR, 54. A PHOSITA would understand "Program" and "Meeting" as thread summary indicators—titles identifying and summarizing the thread's subject, and the claim imposes no format requirement (e.g., textual, graphical)—only that it be renderable within the messaging pane proximate to the first message. Exh-1003, ¶¶192–94.

Kakuta's "Program" label satisfies the claim. *Id.* Kakuta illustrates the "Display" control presenting thread summary indicators like "Program," with the moveable thread selection window can be rendered relative to the main chat. Exh-1005, Fig. 12. Even if not fixed beside the first message, the movable window enables the "Program" label to appear near the relevant communication, consistent with standard GUI design. *Id.*; Exh-1021, ¶84.

4. 30[e]

PO repeats its 30[d] argument, but Kakuta's thread titles (e.g., "Program") appear beside the messaging pane, identify threads anchored to main chat messages, and when selected (i.e., input), retrieve and display the thread. Exh-1005, 9:23–31, Figs. 12, 19 (S432–S434); Exh-1003, ¶¶195–96.

5. 30[f]

PO incorrectly argues that the channel name field in Figure 7 is not a “list pane” (POR, 54–55), but Kakuta’s channel selection mechanism lists multiple channel identifiers and serves that function. Section IV.A.1. A PHOSITA would recognize it as a list pane displayable concurrently with the chat and thread panes. Exh-1005, Figs. 7, 12; Exh-1003, ¶¶197–99.

V. GROUND 2: CLAIMS 1-7, 9-19, AND 21-30 ARE OBVIOUS OVER KAKUTA

PO’s claim that Petitioners “bootstrap[]” their anticipation arguments is unfounded. Expert testimony shows any undisclosed feature would be obvious to a PHOSITA, supported by specific teachings and rationale beyond “common sense.”

A. 1[c], 15[b], and 25[b]

PO wrongly argues Kakuta displays only one channel identifier and that Petitioners “assume” a drop-down list (POR, 56–58), but a PHOSITA would recognize Figure 8’s channel identifier element as a ComboBox-type control listing multiple channels, consistent with GUI frameworks. Exh-1003, ¶¶201–07; Exh-1012, 52–53.

Even if Kakuta does not expressly show multiple channel identifiers, implementing a drop-down or list-based menu would have been an obvious, routine design choice to handle many IRC channels, as such were standard in chat GUIs.

Exh-1021, ¶¶85–87. ComboBoxes “occupy little space, even though they may give the user access to an unlimited number of selectable values.” Exh-1025, 153.

PO's reliance on *Rosco, Inc. v. Mirror Lite Co.*, is misplaced, as Petitioners rely on Kakuta's express teachings, obvious GUI design principles, and expert opinion on what a PHOSITA would glean from those, not solely inherency. 304 F.3d 1373, 1380 (Fed. Cir. 2002). Kakuta's channel identifier and a standard ComboBox—sharing identical visual features—would make their functional equivalence apparent to a PHOSITA. Exh-1021, ¶90.

B. 1[d], 15[c], and 25[c]

Kakuta anticipates or renders obvious these limitations. Section IV.A.1. Drop-down menus are inherent in IRC systems and would have been an obvious, routine design for channel switching. Exh-1021, ¶¶91–92.

C. 1[o], 15[n], and 25[n]

Kakuta anticipates or renders obvious these claims. Section IV.A.4. A PHOSITA would find it obvious to add an indicator linking a message to its thread, improving navigation and conversation. Exh-1003, ¶¶212–14.

PO's cited cases are inapposite. *PAR Pharm., Inc. v. TWI Pharms., Inc.*, limits inherency without evidence, but Petitioners identified Kakuta's teachings and explained that a PHOSITA would add an anchor-message indicator as a predictable design enhancement. 773 F.3d 1186, 1195 (Fed. Cir. 2014); Paper 1 (“Pet.”), 85–

86. *Arendi S.A.R.L. v. Apple Inc.*, bars substituting “common sense” for analysis, but Petitioners provided expert evidence and rationale showing the obvious benefit of a visible control linking messages to threads. 832 F.3d 1355, 1362 (Fed. Cir. 2016); Pet., 85–86.

D. 1[i], 1[p], 3[a], 13[a], 15[h], 15[o], 17[a], 23[a], 25[h], 25[o], 27[a], and 28[a]

Kakuta anticipates or renders obvious these limitations. Sections IV.A.3, IV.A.5, IV.B.2, V.H. A PHOSITA would view Kakuta's confirmation step as an optional safeguard, and omitting it for efficiency to directly display the new thread window would be an obvious GUI refinement. Exh-1005, 8:52–56; Exh-1021, ¶¶96–105.

E. Claim 4

Kakuta anticipates or renders obvious this claim. Section IV.B.3. GUI design conventions favored contextual controls near relevant content, making it obvious to relocate or dock Kakuta's toolbar beside a message. Exh-1021, ¶108.

F. Claims 9 and 21

Kakuta anticipates or renders obvious this claim. Sections IV.B.7, IV.B.11. Incorporating identifying information as a separate control would be an obvious design choice, preserving context and continuity. Exh-1021, ¶110–13.

G. Claim 12

Kakuta discloses or renders obvious this claim as its control modules manage user join, leave, and participation changes, inherently handling such requests like standard IRC “/invite” functions. Section IV.B.10; Exh-1021, ¶¶110–13; Exh-1023, 8–9, 211.

H. Claim 13

PO claims Kakuta lacks a “fifth user-selectable control” for displaying messages from multiple channels or a “specific type” (POR, 65), but Kakuta’s toolbar includes distinct controls with different thread or messaging functions. Exh-1005, 8:38–42; Fig. 4. The claims do not limit the control number or order; numbering simply distinguishes functionality. Exh-1003, ¶¶227–30.

The “Link” control uniquely opens a thread list (e.g., “Program”) and links or displays them in a new thread window, and a PHOSITA would find it obvious to extend this to filtering or cross-channel display. Exh-1005, 8:59–9:4; Fig. 10; Exh-1003, ¶¶228–30.

Message filtering and selective viewing were routine in chat and IRC systems, allowing users to filter by topic, participant, or keyword. Exh-1012, 52–53; Exh-1023, 23–24, 114. Kakuta’s administration table already supports such filtering, and a PHOSITA would find it predictable to adapt the “Link” or “Display” functions to

query messages by specific type or across threads. Exh-1005, 6:22–33; Exh-1003, ¶230.

PO wrongly asserts Kakuta must explicitly disclose filtering by “specific type.” The claim only requires a control be “operable to . . . display” such communications—not that the original system must already implement every filter category. Applying known filtering techniques to Kakuta’s modular, multi-thread structure would have been a predictable improvement. Exh-1021, ¶115.

Dr. Almeroth (Exh-2003, ¶¶182–85) concedes Kakuta’s toolbar controls operate independently and the common use of filtering, and the absence of an explicit “specific type” filter does not preclude obviousness, as such filters were an expected design enhancement. Exh-1021, ¶117–18.

I. Claim 14

PO misreads both the claim and Kakuta in asserting it lacks a mechanism to display “a class of messaging communications ... generated by a particular user.” POR, 66.

Kakuta teaches creating threads from selected messages or words and a PHOSITA would understand that selecting one user’s messages inherently forms a thread containing that user’s messages—a class of communications generated by that user. Exh-1005, 3:43–47, 8:45–47, Fig. 7; Exh-1021, ¶120. Highlighting a

participant's messages and activates "Create" or "Link" yields a thread containing only those messages. *Id.*

Kakuta's administration table stores message-identifying information, nicknames, and thread identifiers enabling filtering by sender. Exh-1005, 6:14–33. A PHOSITA would find it obvious to extend this for user-based grouping, as IRC clients routinely supported "view by user" or "filter by participant" features. Exh-1026, 337.

Nothing in the claims limit the "class" to an entire channel or require a novel interface element. Kakuta's "Link" and "Display" already filter and aggregate thread content, and adding an option to view all messages authored by a selected user would have been an obvious refinement. Exh-1021, ¶124.

J. Claims 23–24 and 28–29

PO merely reiterates prior arguments, but Kakuta teaches or renders obvious each limitation. Sections V.H–I; Exh-1021, ¶125.

K. Claim 30

PO contends a PHOSITA would not merge the "Display" and thread-selection steps (POR, 67) but combining them—clicking a thread title to open it—is a routine usability improvement, not an inventive step. Exh-1021, ¶126–27.

PO's reliance on Kakuta's "straightforward" multi-step display process misplaced. POR, 68. Petitioners argue not complexity but that a PHOSITA would

find it obvious to streamline the process—combining the “Display” control and selection list into one step for efficiency. Pet., 98; Exh-1021, ¶¶127–28.

VI. GROUND 3: CLAIMS 1-7, 9-19, AND 21-29 ARE OBVIOUS OVER KAKUTA IN VIEW OF MATSUMOTO

A. 1[c], 15[b], and 25[b]

PO wrongly claims Matsumoto's “channel window 36” is not a “channel selection pane” (POR, 69–70), but it clearly lists multiple active channels (e.g., “#CHOCOA”) for navigation, serving the same function as the claimed pane. Exh-1006, 9:31–46; Fig. 3; Exh-1021, ¶¶129–30.

A PHOSITA would find integrating Matsumoto's channel list into Kakuta's IRC interface straightforward and beneficial, as both organize chats by channel. Exh-1005, 6:28–33, Fig. 8; Exh-1021, ¶¶129–30. Adding Matsumoto's scrollable list would let users view and switch channels without leaving the main window, improving usability. *Id.*

Contrary to PO's claim of “no utility,” a channel pane would advance Kakuta's goal of simplifying multi-channel viewing and management, consistent with common IRC designs of the time. Exh-1023, 8; Exh-1021, ¶131

B. Claims 13–14, 23–24, 28–29

PO wrongly claims Matsumoto lacks additional user-selectable controls (POR, 70–71), but Matsumoto teaches a “Correlate” control that filters or forwards messages across channels by type, keyword, or author—e.g., messages starting with

“#CUSTOMER” or from “user-A.” Exh-1006, 2:65–3:4, 3:51–63, 16:51–65; Exh-1003, ¶¶249–52.

A PHOSITA would find integrating Matsumoto's “Correlate” into Kakuta's toolbar straightforward, as both systems employ user-initiated commands. Exh-1005, 8:38–42, Fig. 4. Adding keyword- or user-based filtering would naturally extend Kakuta's architecture, and Matsumoto's condition tables show that messages from multiple channels can be displayed or transferred to one destination, producing the claimed filtering and cross-thread display. Exh-1006, 16:51–65; Exh-1003, ¶¶252–53.

C. Remaining Claims

PO raises no distinct arguments for the remaining claims, which are obvious. Section VI.A–B.

D. Motivation to Combine

PO's incorrectly claims a PHOSITA would lack motivation to combine Matsumoto with Kakuta. POR, 70. Both concern IRC-based chat systems aimed at improving navigation and managing simultaneous conversations. Exh-1005, 1:6–23; Exh-1006, 1:52–54; Exh-1003, ¶¶247–48. While PO fixates on Matsumoto's main channel window 31, a PHOSITA would view its other elements—like channel window 36 (even if for sub-channels)—as complementary efficient display features. Exh-1021, ¶¶137–38. Integrating channel window 36 into Kakuta's thread-based

layout would be a natural extension from single- to multi-channel threading, as both use modular IRC-style GUIs; adding a scrollable channel list would enhance navigation, and nothing teaches away. *Id.*; Exh-1023, 8.

VII. GROUND 4: CLAIMS 1-29 ARE OBVIOUS OVER KAKUTA IN VIEW OF MATSUMOTO AND CHANG

A. Claims 1, 15, and 25

PO's reading of Chang is overly narrow, claiming it lacks a "second user-selectable control" and applies only to "private chat areas." POR, 73–76. Chang teaches that pressing a participant's identifier or specific message triggers related menu actions, like initiating a private chat. Exh-1007, ¶[0175] ("by pressing identification information UI2 of a specific another party or a message[]"). Likewise, pressing the surrounding message area causes a contextual menu with options to display or hide the private chat. Exh-1007, ¶[0270]; Exh-1003, ¶¶266–68. Thus, touching the message or its display area triggers the control, and a PHOSITA would recognize that selecting a message's visual area—by touch or click—targets that message, as in Kakuta. Exh-1003, ¶269.

PO's claim that the private chat "is already displayed" misreads Chang. POR, 75. Touching the message region opens a menu for that message, allowing the user to toggle visibility or enter the private chat, and this menu interaction constitutes the claimed "receiving a request" to display the thread. Exh-1007, ¶¶[0270], [0296],

[0299], [0318]; Exh-1003, ¶¶269–70. The private area's prior existence does not negate this; the control governs its display state, like Kakuta's "Display" controlling thread windows. Exh-1021, ¶143.

PO wrongly claims Chang's private chat region is not a "third pane," but areas A21 and A22 are visually and functionally distinct regions for private threads, separate from group chat area (A1), displaying concurrently displayed chat areas. Exh-1007, Fig. 29(a), ¶¶[0269]–[0270].

Chang satisfies the "anchor message" association: the first message within private chat region (A21) initiates the thread and remains displayed when opened, and a PHOSITA would understand that touching this message, invoking the menu, identifies it as the anchor. Exh-1003, ¶268.

B. Claim 4

PO improperly argues that Chang lacks a control "displayed proximate to an additional messaging communication." (POR, 79), but Kakuta (Section IV.B.3) and Chang disclose controls functionally and contextually tied to specific messages, and their placement is a routine GUI design choice. Exh-1021, ¶149.

Chang teaches long-pressing a message to open a contextual menu over it, offering options to open or manage related private chats. Exh-1007, ¶[0270], Fig. 29. Chang describes proximity-based touch sensing triggering controls and visual indicators "corresponding to the sensed proximity touch action and proximity

touch position.” *Id.*, ¶[0071]. A PHOSITA would have understood this as disclosing a control displayed proximate to the acted-upon communication. Exh-1021, ¶150.

Both Kakuta (toolbar) and Chang (menu) let users select messages to trigger a control tied to a related thread; Kakuta uses a floating toolbar, while Chang uses a pop-up menu appearing beside the selected message. Exh-1003, ¶¶273–74. Relocating the Kakuta’s control to a contextual menu would be a routine design optimization. Exh-1021, ¶152.

C. Claims 8 and 20

PO inappropriately argues that Chang’s menus are not message-specific and lack an indicator linking a message to a thread. POR, 80. Both Kakuta and Chang provide message-specific cues indicating thread association, satisfying Claims 8 and 20, which require only a visual or contextual signal (e.g., shading, highlighting, symbol, or grouping) of an attached conversation. Exh-1003, ¶¶275–79; Section VII.A.

Kakuta distinguishes thread-related messages display behavior (Exh-1005, 8:45–9:31, Figs. 7, 12), which a PHOSITA would recognize as a contextual thread indicator. Chang makes this explicit by shading private chat areas, visually separating private chats (“indicator identifying”). Exh-1007, ¶¶[0270]–[0271], Fig. 29; Exh-1003, ¶278. A PHOSITA would see such cues as thread indicators and adding them to Kakuta routine. Exh-1003, ¶¶279–80.

D. Claim 12

PO wrongly claims Chang lacks a “requests to permit a non-member user to receive messages.” POR, 81. Chang teaches invitation functionality—dragging and dropping a user into an existing chat to join and receive messages. Exh-1007, ¶¶[0276]–[0285], Figs. 31–33; Exh-1003, ¶284. Combined with the thread access control features of Kakuta and Matsumoto, it would be obvious to process such requests and enable message access. *Id.*

E. Remaining Claims

PO offers no distinct arguments for the remaining claims, which are obvious. Section VII.A–D.

F. Motivation to Combine

PO wrongly claims the Kakuta-Matsumoto-Chang combination is hindsight-driven (POR, 85–86). All three address usability in real-time chat: Kakuta enhances IRC with threaded conversations (Exh-1005, 1:6–23), Matsumoto enables multi-channel viewing (Exh-1006, 1:47–2:15), and Chang provides message-specific controls (Exh-1007, ¶¶[0270]–[0271]). They aim to simplify navigation and organization in multi-party messaging. Exh-1003, ¶¶291–99.

A PHOSITA would have been motivated to combine these features to improve efficiency and clarity: Matsumoto's channel window extends Kakuta's threads to multi-channel management, and Chang's contextual controls provide intuitive

thread access. Exh-1003, ¶¶295–97. Their modular, pane-based designs align with IRC architectures. Exh-1021, ¶¶167–69.

PO incorrectly argues Chang is inapplicable because it uses mobile interfaces. A PHOSITA would equate touch with mouse or keyboard input, making adaptation routine. Exh-1003, ¶¶297–98. Similarly, Matsumoto and Kakuta are IRC-based and fully compatible, focusing on display management. *Id.*

None of the references “teach away,” and a PHOSITA would view combining Matsumoto’s multi-channel management, Kakuta’s thread clarity, and Chang’s simplified interaction as an obvious improvement in navigation and usability. Exh-1021, ¶174.

VIII. GROUND 5: CLAIM 30 IS OBVIOUS OVER KAKUTA IN VIEW OF CHANG

A. 30[d]

PO misreads Kakuta and Chang in claiming the latter lacks a message-specific “thread summary indicator” and that the combination would merely list all threads. POR, 82–84. Kakuta discloses a functional summary thread indicator and Chang, adds message-level association by letting users tap or long-press a message to open a related chat, inherently serving as a proximate indicator. Sections IV.C.3, V.K, VII.B; Exh-2021, ¶¶176–181

IX. CONCLUSION

For these reasons and those in the Petition, Claims 1–30 of the '731 Patent are unpatentable.

Dated: November 5, 2025

Respectfully submitted,

/Robert C.F. Pérez/

Robert C.F. Pérez (Reg. No. 39,328)

**PILLSBURY WINTHROP SHAW
PITTMAN LLP**

7900 Tysons One Place, Suite 500

Tysons, VA 22102

Telephone: 703.770.7759

Facsimile: 703.770.7901

Email: robert.perez@pillsburylaw.com

Christopher Kao (*pro hac vice*)

Brock S. Weber (*pro hac vice*)

**PILLSBURY WINTHROP SHAW
PITTMAN LLP**

Four Embarcadero Center, 22nd Floor

San Francisco, CA 94111

Telephone: 415.983.1000

christopher.kao@pillsburylaw.com

brock.weber@pillsburylaw.com.

Counsel for Petitioners

Slack Technologies, LLC and

Salesforce, Inc.

CERTIFICATE OF COMPLIANCE

1. The undersigned certifies that this brief complies with the type volume limitations of 37 CFR § 42.24(a)(1)(i). This brief contains 5598 words (excluding the table of contents, the table of authorities, mandatory notices under 37 CFR § 42.8, the certificate of service, certificate of compliance, and appendix of exhibits), as calculated by the “Word Count” feature of Microsoft Word 360, the word processing program used to create it.

2. The undersigned further certifies that this brief complies with the typeface requirements of 37 CFR § 42.6(a)(2)(ii) and typestyle requirements of 37 CFR § 42.6(a)(2)(iii). This brief has been prepared in a proportionally spaced typeface using Microsoft Word in Times New Roman 14-point font

Dated: November 5, 2025

Respectfully submitted,

/Robert C.F. Pérez/

Robert C.F. Pérez (Reg. No. 39,328)

Christopher Kao (*Pro hac vice*)

Brock S. Weber (*Pro hac vice*)

Counsel for Petitioners

Slack Technologies, LLC and

Salesforce, Inc.

CERTIFICATE OF SERVICE

The undersigned hereby certifies that a true copy of the foregoing
**PETITIONERS' REPLY TO PATENT OWNER RESPONSE FOR INTER
PARTES REVIEW OF U.S. PATENT NO. 11,973,7311** and supporting
materials are being served by electronic mail on the following attorneys of record:

LEAD COUNSEL	BACK-UP COUNSEL
Steven R. Katz (Reg. No. 43706) FISH & RICHARSON 60 South Sixth Street, Suite 3200 Minneapolis, MN 55402 Telephone: 617-542-5070 Facsimile: 877-769-7945 Email: katz@fr.com	Lawrence K. Kolodney (Reg. No. 43,807) FISH & RICHARSON 60 South Sixth Street, Suite 3200 Minneapolis, MN 55402 Telephone: 617-542-5070 Facsimile: 877-769-7945 Email: kolodney@fr.com

/Robert C.F. Pérez/
Robert C.F. Pérez