UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

YEALINK (USA) NETWORK TECHNOLOGY CO., LTD., and
YEALINK NETWORK TECHNOLOGY CO., LTD.
Petitioners,

v.

BARCO N.V.
Patent Owner.

U.S. Patent No. 11,966,346
IPR2025-00597

REPLY DECLARATION OF
KEVIN C. ALMEROTH, PH.D.,
IN SUPPORT OF
PETITIONER'S REPLY

TABLE OF CONTENTS

I. INTRODUCTION 3
II. ADDITIONAL MATERIALS REVIEWED 4
III. SUMMARY OF OPINIONS 6
IV. SOFTWARE FUNCTIONALITY IS INDEPENDENT OF STORAGE MEDIUM 6
V. HOW OPERATING SYSTEMS INTERACT WITH DEVICES: THE ROLE OF DRIVERS 9
VI. DATA TRANSFORMATIONS ARE TYPE-SPECIFIC 12
VII. A POSA WOULD NOT HAVE BEEN DISCOURAGED BY DINKA's DISCLOSURES 16
VIII. A POSA WOULD HAVE FOUND IT OBVIOUS TO COMBINE THE PRIOR ART REFERENCES 19
A. Beel, Dinka, and Christison (Ground 1) 19
B. Beel's Base Unit Transmits Data to the Processing Device 21
C. Virtual Device Presentation in Beel's Peripheral-Less Embodiments 24
D. Kaplan, Van de Laar, and Christison (Ground 2) 27
IX. OBSERVATIONS REGARDING DR. BROGIOLI's TECHNICAL OPINIONS 32
X. ADDITIONAL TECHNICAL ANALYSIS FOR SPECIFIC CLAIM LIMITATIONS 39
A. Claims 21 and 22: "Presenting the First Display Device to the Processing Device" 39
B. Claim 6: "Decoding and Compositing or Decrypting the Video Data" 41
C. Ground 2: Additional Technical Analysis 44
D. Commercial Success: Analysis of Feature Attribution 52
XI. SECONDARY CONSIDERATIONS 55
XII. CONCLUSION 56

I, Kevin C. Almeroth Ph.D., declare as follows:

I. INTRODUCTION

1. I previously submitted a declaration (Ex-1002) in support of the Petition for Inter Partes Review of U.S. Patent No. 11,966,346 ("the '346 patent"). I have been asked to submit this Reply Declaration in further support of the Petition and in response to Patent Owner's Response ("POR"), the Declaration of Dr. Michael C. Brogioli (Ex. 2004), and certain deposition testimony given in this proceeding.

2. I am the same Kevin C. Almeroth identified in my prior Declaration (Ex-1002). My qualifications, professional background, and compensation terms remain as stated in that Declaration. I am a Professor Emeritus of Computer Science at the University of California, Santa Barbara. I am a Fellow of the Institute of Electrical and Electronics Engineers (IEEE) and a Member of the Association of Computing Machinery (ACM). My research expertise includes computer networks and protocols, wireless networking, multicast communication, large-scale multimedia systems, and mobile applications. My full curriculum vitae was submitted as Exhibit 1003.

3. For my efforts in connection with the preparation of this Reply Declaration, I have been compensated at my usual and customary rate for this type

of consulting activity. My compensation is in no way contingent on the outcome of these or any other proceedings.

4. In preparing this Reply Declaration, I have reviewed all materials identified in my prior Declaration, as well as Patent Owner's Response (Paper 19) ("POR"), the Board's Institution Decision (Paper 11) ("ID"), the Declaration of Dr. Michael C. Brogioli (Ex. 2004), the deposition transcripts of Dr. Michael C. Brogioli (Ex-1050 for the related '972 IPR and Ex-1060 for this proceeding), Petitioner's Reply, and the additional materials identified below.

5. My opinions contained in this Reply Declaration are based on the documents I reviewed, my knowledge and professional judgment, and my appreciation of how a person of ordinary skill in the art ("POSA") would have understood the state of the art, the teachings of the prior art, and the claim terms and specification of the '346 patent at the time of the alleged invention, which I have been asked to assume is around December 29, 2017.

II. ADDITIONAL MATERIALS REVIEWED

6. In addition to the materials identified in my prior Declaration (Ex-1002), I have reviewed the following:

Exhibit	Description
Ex. 2004	Declaration of Dr. Michael C. Brogioli
Ex. 2005	Curriculum Vitae of Dr. Michael C. Brogioli
Ex-1050	December 18, 2025 Deposition Transcript of Dr. Michael C. Brogioli for IPR2025-00491

Exhibit	Description
Ex-1051	Microsoft, "USB device drivers (Windows Drivers)," Microsoft Learn, https://learn.microsoft.com/en-us/windows-hardware/drivers/usbcon/ (accessed Mar. 2026)
Ex-1052	"Universal Serial Bus Specification, Revision 2.0," §§ 5.3, 9.1–9.2, April 27, 2000, http://www.poweredusb.org/pdf/usb20.pdf (accessed Mar. 2026)
Ex-1053	"Virtual Audio Device (Kernel Streaming)," Microsoft Learn, https://learn.microsoft.com/en-us/windows-hardware/drivers/audio/virtual-audio-devices (accessed Mar. 2026)
Ex-1054	"Windows Driver Model (WDM)," Microsoft Learn, https://learn.microsoft.com/en-us/windows-hardware/drivers/kernel/introduction-to-wdm (accessed Mar. 2026)
Ex-1055	"About Multimedia (Audio/Video)," Microsoft Developer Network Archive, https://learn.microsoft.com/en-us/windows/win32/multimedia/about-multimedia (accessed Mar. 2026)
Ex-1056	"Wireless USB Specification, Revision 1.1," Wireless USB Promoter Group, Sept. 2010, https://www.usb.org/document-library/wireless-usb-specification-rev-11 (accessed Mar. 2026)
Ex-1057	Salman A. Baset & Henning Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol," Columbia University, 2006, https://arxiv.org/abs/cs/0412017 (accessed Mar. 2026)
Ex-1058	"Wi-Fi Peer-to-Peer (P2P) Technical Specification, Version 1.7," Wi-Fi Alliance, October 2016
Ex-1059	"Overview of the Media Foundation Architecture," Microsoft Learn, https://learn.microsoft.com/en-us/windows/win32/medfound/about-topologies (accessed Mar. 2026)
Ex-1060	Brogioli Deposition Transcript
Ex-1061	ClickShare C-5 datasheet

7. I also reviewed any other materials I refer to in this Reply Declaration in support of my opinions.

III. SUMMARY OF OPINIONS

8. Having reviewed Patent Owner's Response and the Declaration of Dr. Brogioli, my opinions as expressed in my original Declaration (Ex-1002) remain unchanged. In this Reply Declaration, I focus on providing responsive technical explanations regarding the specific issues raised by Dr. Brogioli and Patent Owner. In particular, I address: (a) why software functionality does not change when copied from one storage medium to another; (b) why drivers are an inherent part of how operating systems interact with devices, including virtual devices; (c) how codecs operate as type-specific data transformations; (d) why Dinka's disclosures about computer-based VoIP would not have discouraged a POSA from applying those teachings; (e) how Van de Laar's flexible dockee architecture operates; and (f) specific concerns I have about the reliability of Dr. Brogioli's technical opinions.

IV. SOFTWARE FUNCTIONALITY IS INDEPENDENT OF STORAGE MEDIUM

9. A recurring theme in Dr. Brogioli's declaration is the contention that features Beel describes in its peripheral-device embodiments "cannot be attributed" to Beel's third and fourth embodiments, which do not require a peripheral device. Ex. 2004, ¶101. In my view, this position reflects a fundamental misunderstanding of how software works.

10. Beel's fourth embodiment is described as "similar to the third embodiment, with as only difference that the software is copied on the client operating device." Ex-1005, ¶247. The key phrase here is "only difference." As someone who has spent decades researching, building, and teaching about software systems, I can say with confidence that the behavior of a software program is determined by its code, not by whether those instructions were loaded from a USB stick, downloaded from a network, or pre-installed on a hard drive. This is one of the most basic principles of computer science. A word processing application does not lose the ability to format text because it was installed from a download rather than a disc. A video codec does not stop encoding video because its executable was copied from one directory to another.

11. When Beel says the "only difference" is the location of the software, a POSA would read that as an express statement that every capability the software had in prior embodiments, *e.g.*, virtual device presentation, screen scraping, encoding, communication with the base node, is preserved. To conclude otherwise would require believing that the act of copying a program from a USB stick to a laptop's internal storage somehow strips that program of its functionality. I am not aware of any principle of computer science, operating system design, or software engineering that would support such a conclusion, particularly in the context of how Beel describes the functionality. See, e.g., "USB device drivers (Windows Drivers),"

Microsoft Learn (documenting that operating systems load and execute drivers in the same manner regardless of the installation source).

12. During his deposition, Dr. Brogioli did not offer any technical reason why the software would behave differently when stored on the processing device. Ex-1050. Indeed, in his more recent deposition in this proceeding, Dr. Brogioli confirmed that in Beel's third embodiment, "the execution of the software would ultimately be the execution of instructions on the Intel processor in the...Windows laptop." Ex-1060, 49:15-20. When asked whether the instructions "would be executed on the processor of the processing device," he answered: "As far as I recall, that sounds correct." Ex-1060, 49:21-50:1. He also acknowledged that in the fourth embodiment, there is "no plug-and-play port such as the USB port...required on the client operating device." Ex-1060, 45:6-46:1. These admissions confirm that the software executes on the processing device regardless of its original storage location. Dr. Brogioli's written position is not rooted in engineering analysis but in a textual argument about how Beel's specification is organized. I do not find that argument technically persuasive because I agree with Dr. Brogioli's admission concerning execution of software ultimately being done on the main processing device. Ex-1060, 49:15-20.

V. HOW OPERATING SYSTEMS INTERACT WITH DEVICES: THE ROLE OF DRIVERS

13. Dr. Brogioli opines that "operating a simulated peripheral does not necessarily require using a driver." Ex. 2004, ¶152. Patent Owner similarly argues that "neither the Petition, nor Dr. Almeroth, mention the word driver" in arguing the relevant limitation for Ground 2. POR at 39. I want to address this directly because it touches on a foundational aspect of computer architecture that I believe Dr. Brogioli has gotten wrong. Notably, when confronted with this opinion during his deposition, Dr. Brogioli could not "identify any real-world operating systems where a simulated peripheral is presented to an application without any driver involvement." Ex-1060, 149:18-24. He also admitted he has "no opinion" on whether operating systems present devices to applications without using drivers. Ex-1060, 78:1-12, 149:14-150:1.

14. In every major operating system I have worked with or studied, such as Windows, macOS, Linux, and others, the interaction between application software and hardware (or virtualized hardware) is mediated by drivers. This is not an optional design choice; it is how modern operating systems are architected. The operating system kernel does not allow application software to access devices directly. Instead, application software communicates with a driver, and the driver communicates with the device (or, in the case of a virtual device, simulates the device's responses). This

layered model is thoroughly documented in publicly available materials such as the Windows Driver Model (WDM) documentation on Microsoft Learn, which describes how all device interactions in Windows pass through a layered driver stack. Dr. Brogioli himself confirmed that pre-installed generic drivers for WiFi interfaces were well-known at Beel's priority date and that "the person of skill in the art would be aware of the use of these types of drivers." Ex-1060, 63:5-65:2.

15. When Van de Laar describes a "simulated webcam" that "would appear to the dockee as if it were a normal peripheral" (Ex-1007, ¶54), a POSA would understand that this appearance is achieved through a driver on the dockee's operating system. That is the only mechanism by which a device, real or virtual, can "appear" to the operating system and to applications. The word "driver" does not need to appear in Van de Laar for a POSA to understand this, any more than a patent describing a car needs to mention "internal combustion" for a POSA to understand how the engine works. It is a basic fact of the technology.

16. The same analysis applies to Christison's disclosure. When Christison describes wired USB devices appearing to a host computer "as if [they were] attached" (Ex-1011, 6:25-26), this presentation to the host involves the host's USB driver stack. As documented in the Universal Serial Bus Specification, Revision 2.0, §§ 9.1-9.2, whenever any USB device is presented to a host, the host enumerates the device and loads the appropriate driver. This happens automatically, whether the

device is physically wired or presented remotely through a wireless hub. There is no alternative pathway for device recognition in USB-based systems.

17. I want to provide additional technical context regarding why Christison's system involves drivers on the host computer, beyond what I stated above. In the USB standard architecture, when any device (physical or virtual, wired or wireless) is connected to a host, the host controller initiates an enumeration process. During enumeration, the host reads the device's descriptors, including the device descriptor (which identifies the device class), the configuration descriptor, and the endpoint descriptors. Based on this information, the operating system's Plug and Play manager loads the appropriate driver. This process is mandatory and automatic; there is no pathway in the USB architecture for a device to communicate with application software without going through this driver layer. Microsoft's documentation on USB endpoints explains that "[e]ndpoint is hardware on the device" and a "pipe is software on the host side," and that "[a] pipe talks to an endpoint on a device." Ex-1010. The pipe is a software abstraction through which applications communicate with the device and is created when the driver configures the endpoint. Without a driver, there is no pipe, and without a pipe, there is no communication. When Christison describes presenting wired USB devices as "native" wireless USB devices (Ex-1011, 6:25-26), a POSA would have understood that this presentation occurs through exactly this mechanism: the host's USB driver

stack enumerates the device, loads the appropriate class driver (or a custom driver), and creates the pipes through which data flows. Dr. Brogioli's suggestion that virtual device presentation can occur without driver involvement ignores this fundamental aspect of USB architecture.

18. The '346 patent itself acknowledges this reality. It states that "a driver is standard for the operating system and can drive a standard class of peripheral devices." Ex-1001, 8:53-67. In my view, this confirms what any POSA would already know: drivers are part of the standard operating system infrastructure for device interaction.

VI. DATA TRANSFORMATIONS ARE TYPE-SPECIFIC

19. Dr. Brogioli takes the position that Beel and Dinka do not teach data transformations "based on a type of functional data." Ex. 2004, ¶¶110-112. Patent Owner has also attempted to use my deposition statement that "data is data" to undermine my opinions on this point. POR at 24-25. I want to clarify the technical concepts involved.

20. In my field, we draw a clear distinction between data transport and data transformation. Data transport refers to moving data from one point to another, *e.g.*, over a network, across a bus, through a cable. At most of the lower layers of the protocol stack, including the transport layer, "data is data." The communication medium transmits bits without regard to whether those bits represent audio, video,

or anything else. That is what I meant during my deposition, and it is a standard concept in networking and telecommunications.

21. Data transformation is different. It refers to the processing operations applied to data before or after transport, such as encoding, decoding, compression, decompression, encryption, decryption, mixing, compositing. These operations are fundamentally type-specific. A video codec like H.264 exploits spatial and temporal redundancy in image frames; it uses motion estimation, discrete cosine transforms, and inter-frame prediction. An audio codec like Ogg/Vorbis or Opus operates in the frequency domain, using psychoacoustic models to discard audio information that human hearing cannot perceive. These are completely different algorithms solving completely different problems, each tailored to the characteristics of the data type they process. You cannot meaningfully apply a video codec to audio data or vice versa.

22. Beel expressly uses type-specific transformations. For video, Beel describes: "The video signal is then encoded in a video encoder 3, packetized in a video packetizer 4." Ex-1005, ¶320. For audio, Beel describes: "Audio is...optionally encoded into for instance Ogg/Vorbis." Ex-1005, ¶314. The fact that Beel routes video data through a video encoder and audio data through an audio codec is, in my view, a textbook example of performing transformations "based on a type of functional data." Dr. Brogioli confirmed this understanding during his deposition,

acknowledging that Beel's "video encoder can encode video data using a video codec, and audio data using an audio codec" and that "a client PC...could run an audio encoder" to "encode a .wav file to mp3." Ex-1060, 77:5-25. The Windows multimedia architecture works the same way: as documented in Microsoft's documentation, the operating system maintains separate processing pipelines and codecs for audio and video data, with distinct codec chains for each type. Exs-1055, -1059.

23. Dinka similarly employs a dedicated "voice engine" for encoding speech signals (Ex-1006, 9:32-42) and separate video codecs for visual data (Ex-1006, 9:45-50). This is consistent with how Skype itself operated during the relevant time period. As documented in Baset & Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol," Skype employed codecs such as iSAC and iLBC for audio data and VP7/H.264 for video data. Ex-1057. The use of different codecs for different data types was standard practice in unified communications systems well before 2017.

24. Dr. Brogioli argues that "while different types of data can be encoded differently, they can also be encoded the same way." Ex. 2004, ¶111. While it is theoretically possible to apply a generic compression algorithm (such as the ones used in the "gzip" utility) to any data stream, that is not how multimedia systems work, and it is not what the prior art teaches. In practice, multimedia systems use

type-specific codecs because generic compression cannot achieve the compression ratios, latency requirements, or perceptual quality needed for real-time audio and video communication. A POSA would understand this distinction immediately. Significantly, during his deposition, Dr. Brogioli himself confirmed that Beel teaches compression and scaling operations for bandwidth management, testifying that Beel describes situations where "you may want to compress the media content that's being sent to them or scale it" and that this involves "changing a resolution." Ex-1050, 91:16-93:8. This admission undercuts his written opinion that Beel does not teach type-specific data transformations.

25. I also note that during prosecution, Barco itself described the "data transformations" limitation as referring to transformations "related to the functional data of the functional device, e.g., the images for a camera, the audio data for a microphone, etc., in which the transformations can include encoding/decoding, mixing, (de)multiplexing." Ex-1004, 1038. This description matches precisely what Beel and Dinka disclose.

26. I want to explain why the claim limitation requiring data transformations "based on a type of functional data" is satisfied by the use of type-specific codecs from a systems architecture perspective. In multimedia processing pipelines, the selection of a codec is determined by the data type. This is not merely a matter of choosing different software modules, it reflects fundamentally different

mathematical and algorithmic approaches to data compression. Video codecs like H.264 operate on frames as two-dimensional arrays of pixels, exploiting spatial correlation (adjacent pixels tend to have similar values) and temporal correlation (successive frames tend to be similar). They use techniques such as block-based motion compensation, discrete cosine transforms on 8x8 or 16x16 pixel blocks, and variable-length entropy coding optimized for image statistics. Audio codecs like Ogg/Vorbis or AAC, by contrast, transform the signal into the frequency domain using modified discrete cosine transforms, then apply psychoacoustic models to determine which frequency components can be discarded without perceptible quality loss. The perceptual models for audio are based entirely on human hearing characteristics—critical bands, temporal masking, simultaneous masking—concepts that have no analog in video processing. A POSA would understand that applying a video codec to an audio stream would produce meaningless output, and vice versa. The reason Beel and Dinka route video through video codecs and audio through audio codecs is not incidental, it is the only technically coherent approach. This is performing transformations "based on" the type of data.

VII. A POSA WOULD NOT HAVE BEEN DISCOURAGED BY DINKA's DISCLOSURES

27. Dr. Brogioli contends that Dinka "teaches away" from using laptops for VoIP because Dinka "discourages use of laptops to host the virtual meeting and

instead advocates for participating in packet-based communication from a television or set-top box." Ex. 2004, ¶¶80-87. Based on my experience in this field, I disagree.

28. Dinka itself discloses a system built around "a plurality of computer terminals 102" participating in VoIP calls. Ex-1006, 5:45-48. It describes each terminal as being "installed with a communication client application 110" enabling "VoIP calls according to peer-to-peer principles." Ex-1006, 6:7-8, 6:23-26. And it acknowledges that VoIP is "most commonly accessed using a personal computer." Ex-1006, 2:1-4. In my view, a reference that describes, in detail, a system for computer-based VoIP communication cannot reasonably be understood as discouraging that very approach.

29. What Dinka actually discusses is a practical observation that some users may find it cumbersome to install and configure VoIP software on a laptop. Ex-1006, 2:4-8. I can confirm from personal experience that this was a common observation during the relevant time period; it was one reason why videoconferencing equipment manufacturers developed simplified interfaces. But noting a practical inconvenience is very different from saying the technology does not work or should not be used. By 2017, Skype had hundreds of millions of users running the application on personal computers. See Baset & Schulzrinne (documenting Skype's widespread PC-based deployment). I personally used Skype and similar VoIP applications on my laptop regularly during this time period, as did

many of my colleagues in academia and industry. The notion that a POSA would have read Dinka and concluded that laptop-based VoIP was disfavored or should be avoided does not comport with reality.

30. Dr. Brogioli's testimony during his deposition is relevant here. When asked whether Dinka contemplates a plurality of computer terminals communicating via Skype, Dr. Brogioli stated he did not "recall discussing that in my declaration" and had "no opinion" beyond what was in his declaration. Ex-1050, 85:2-86:9. When asked whether Dinka is limited to communication between a computer terminal and television sets, he again stated he did "not recall talking about that aspect of Dinka." Ex-1050, 86:14-19. In his more recent deposition, when confronted with Dinka's disclosure of "a plurality of computer terminals 102" connected for VoIP communication (Ex-1006, 5:45-46, FIG. 1), Dr. Brogioli acknowledged Dinka "talks about...computers connected...over a packet-based computer network such as the internet" for voice and video calls. Ex-1060, 102:8-24. He also admitted that he does not "recall specifically that Dinka is making the statement that a VoIP call would be nonfunctional between two laptops." Ex-1060, 99:14-17. In my view, it is difficult to understand a rational technical basis for suggesting that Dinka discourages laptop-based VoIP when the expert offering that opinion has no view on whether Dinka actually contemplates laptop-based VoIP and concedes that Dinka discloses computer-based VoIP.

VIII. A POSA WOULD HAVE FOUND IT OBVIOUS TO COMBINE THE PRIOR ART REFERENCES

A. Beel, Dinka, and Christison (Ground 1)

31. Dr. Brogioli opines that "a POSITA would not be motivated to combine Beel, Dinka, Christison." Ex. 2004, ¶74. In my view, the combination would have been straightforward for a POSA, and I want to explain why from a technical perspective.

32. Beel and Dinka address the same fundamental engineering problem: enabling multimedia communication between participants using networked devices equipped with cameras, microphones, and displays. Beel focuses on the meeting-room environment with a base node connecting functional devices. Ex-1005, ¶¶85-89, 120-122. Dinka focuses on peer-to-peer VoIP communication using an application like Skype. A POSA working on conferencing systems in 2017 would have been intimately familiar with both types of systems and would have recognized that running a Skype-type application on a processing device connected to Beel's base node is exactly the kind of integration that was happening across the industry during this time period. The '346 patent itself identifies Skype as a known "tool[] for unified communications" in its background section. Ex-1001, 1:20-25.

33. Dr. Brogioli suggests that because Dinka is assigned to Skype and Beel is assigned to Barco, a POSA would not think to combine them. Ex. 2004, ¶93. In

my experience, engineers working on conferencing systems do not limit their thinking to solutions developed by a single company. The entire premise of the '346 patent is running third-party unified communications software on a processing device connected to a base unit with functional devices and involves integrating technology from different sources. A POSA would think nothing of considering how a Skype-type application could leverage hardware from a different vendor. Moreover, Dr. Brogioli's requirement for what's necessary for combinability would undercut basic tenets of an obviousness analysis.

34. Regarding Christison, the concept of presenting remote devices as if they were locally connected was well established by 2017. The Wireless USB Specification (Rev. 1.1, Sept. 2010) (Ex-) defined the protocols for wireless USB device presentation, and technologies like Wi-Fi Direct (documented by the Wi-Fi Alliance) provided the underlying wireless connectivity frameworks. Christison's use of a Proxy WUSB Hub to present wired USB devices as native wireless devices represents one specific implementation of this broadly understood concept. In my view, a POSA looking to present Beel's functional devices as virtual devices on a processing device would naturally have considered techniques like those described in Christison.

B. Beel's Base Unit Transmits Data to the Processing Device

35. Dr. Brogioli argues that Beel does not disclose data transmission from the base unit to the processing device. Ex. 2004, ¶¶76, 104-106. From a technical standpoint, I find this argument difficult to reconcile with what Beel actually describes.

36. Beel describes "microphone or microphones 38 that can be used to transfer audio, e.g. to the processing devices 31." Ex-1005, ¶120. The microphones are functional devices connected to the base node. For audio data from those microphones to reach the processing devices, it must pass through the base node's wireless connection. Beel's description of the base node having a "transceiver" (Ex-1005, ¶129) confirms this; a transceiver, by definition, both transmits and receives. Similarly, Beel's software on the processing device includes "a third software code portion for receiving media content from the network and for displaying the media content on the display." Ex-1005, ¶71. The processing device has software expressly designed to receive media content, which in the context of Beel's system includes content from functional devices connected through the base node.

37. I want to address Patent Owner's apparent position that Beel's microphones 38 are limited to lapel microphones that connect directly to the processing device. Dr. Brogioli acknowledged that Beel's paragraph 120 "doesn't use the 'lapel' description" and that Beel "is also referring to it simply as

'microphone' in the context of paragraph 120." Ex-1060, 24:14-25. He further admitted that microphones 38 "probably could be a microphone that sits on a desk or something like that." Ex-1060, 26:5-7. When asked whether the microphone could be "integrated into the ceiling" of a conference room, Dr. Brogioli conceded he had not "performed an analysis of the different form factors the microphone could or could not be in the context of Beel" and stated he had "no opinion one way or the other whether the microphone 38 could be integrated into the ceiling." Ex-1060, 26:8-27:7. He also acknowledged that "ceiling microphones were known in the art" at the time of Beel's 2012 filing. Ex-1060, 28:7-11.

38. In my opinion, Beel's own disclosure supports a broader reading. Paragraph 118 describes "audio equipment 46" that "may be provided (e.g., via telephone) which allows other members from the meeting to call in from remote destinations." Ex-1005, ¶118. Figure 1A depicts audio equipment 46 at the center of the conference table, surrounded by a ring of dots that appear visually similar to the microphones 38. Dr. Brogioli confirmed during his deposition that these dots "are visually similar" to the microphones. Ex-1060, 30:1-4. The description of audio equipment 46 as including telephone functionality is consistent with the audio equipment having integrated microphones. More significantly, Beel's paragraph 120 describes microphones 38 transferring audio "to the processing devices 31 and to loudspeakers (not shown) attached to the base node 36." Ex-1005, ¶120. The fact

that microphone audio can be transferred to loudspeakers "attached to the base node" demonstrates that the microphones can communicate through the base node, not merely directly to the processing device. A ceiling microphone in a conference room, connected via network infrastructure to the base node, would satisfy this disclosure. Dr. Brogioli's insistence that the microphones must connect "directly" to the processing device adds a limitation that does not appear in Beel's text and ignores the express disclosure that audio can be transferred to devices attached to the base node.

39. Even if one were to read Beel narrowly on this point, Dinka's bidirectional communication architecture provides the missing piece. Dinka's system encodes, decodes, and (de)multiplexes data in both directions. Ex-1006, Abstract, 8:7-51, 9:32-50. In the combination, data from functional devices connected to the base unit would flow to the processing device through the bidirectional communication channel, exactly as the claims describe.

40. I also note that Beel's description of its system is not limited to simple screen sharing. Beel expressly describes "electronic meeting systems," "web conferencing systems," and "data conferencing" for "screen sharing and voice conferencing" so that "a screen, keyboard, mouse, camera, etc. can be shared." Ex-1005, ¶¶85, 87-88. Beel further explains that "the present invention provides an electronic meeting tool for communicating arbitrary media content between different

users 37 (with their own processing devices 31, e.g. PC, mobile phone, or tablet) and one display or projector or multiple displays or projectors 44 in the meeting room 30." Ex-1005, ¶122. This broader context confirms that Beel's system was designed for the same bidirectional collaborative communication that characterizes modern conferencing systems. Dr. Brogioli's focus on isolated embodiments ignores this overarching purpose.

C. Virtual Device Presentation in Beel's Peripheral-Less Embodiments

41. I addressed the general principle that software functionality is independent of storage medium in Section IV above. Here, I want to add a specific technical point about virtual device presentation.

42. Beel describes software that causes a "functional device" to appear to the operating system "as a composite device comprising for instance a (virtual) audio speaker device." Ex-1005, ¶317. This virtual device presentation is accomplished through software running on the processing device that interacts with the operating system's device management framework, using pre-installed generic drivers such as the USB Audio Class (UAC) driver. Ex-1005, ¶¶43, 312, 319. The peripheral device in Beel's earlier embodiments is merely the delivery mechanism for this software; the actual virtual device presentation is performed by code executing on the processing device, interacting with the processing device's operating system. Once

the software is loaded and launched on the processing device (Ex-1005, ¶208), the peripheral device's involvement in virtual device presentation is complete. Notably, Dr. Brogioli confirmed this understanding during his deposition, testifying that Beel describes "capturing audio on the peripheral device and then that peripheral device preferably acts as a composite device comprising, for instance, a virtual audio speaker." Ex-1050, 117:14-118:8.

43. When the fourth embodiment teaches that this software can be "copied on the client operating device" (Ex-1005, ¶247), a POSA would understand that the virtual device presentation capability travels with the software. Dr. Brogioli acknowledged this during his deposition, confirming that the portable application "goes on the client operating device" in the fourth embodiment. Ex-1060, 45:6-47:4. He could not identify any technical reason why the executable codes in Beel paragraphs 71-72 "could not be implemented using software on the client device, as explained in the fourth embodiment." Ex-1060, 34:8-35:23. See also "Virtual Audio Device (Kernel Streaming)," Microsoft Learn (describing how virtual audio devices are implemented entirely through software drivers interacting with the operating system's audio stack, with no dependency on any particular physical hardware).

44. I also want to address the dependent claims regarding simulated functional devices. Claims 2 and 12 recite a "simulated speakerphone," and claims 3 and 13 recite a "simulated videocam." Beel expressly discloses these features. For

the simulated speakerphone, Beel describes a "(virtual) audio speaker device" that is presented to the processing device's operating system through "pre-installed generic drivers." Ex-1005, ¶317. Beel also describes "microphones 38" connected to the base node that transfer audio to processing devices. Ex-1005, ¶¶120, 317. The combination of a virtual speaker and microphone functionality constitutes a speakerphone—a device that provides bidirectional audio communication. A POSA would understand that presenting these as virtual devices through the processing device's operating system creates a "simulated speakerphone" as claimed.

45. Similarly, for the simulated videocam, Beel's base node includes "cameras 39, 40, 41" that capture video data and are linked to the base node. Ex-1005, ¶¶119-121. When video data from these cameras is transmitted to the processing device and presented to the operating system as a virtual camera device, this constitutes a "simulated videocam." The same software architecture that presents virtual audio devices can present virtual video devices—this is standard operating system functionality. Furthermore, in the combination with Dinka, bidirectional video communication is expressly taught. Dinka discloses that users can "establish a voice or video call over the Internet" with data flowing in both directions, including video data from webcams. Ex-1006, 6:12-13, 6:34-37. A POSA would have understood that the combination naturally enables simulated videocam functionality for unified communications calls.

D. Kaplan, Van de Laar, and Christison (Ground 2)

46. Dr. Brogioli argues that Petitioner "indiscriminately" relies on features of both primary and secondary dockees from Van de Laar. Ex. 2004, ¶¶126-134. In my view, this criticism misunderstands Van de Laar's architecture, which is intentionally flexible. During his deposition, Dr. Brogioli confirmed that dockees "can switch roles between primary and secondary and vice versa." Ex-1060, 85:17-92:24. This admission directly undercuts his written opinion that primary and secondary dockees have rigid, non-overlapping capabilities.

47. Van de Laar expressly teaches that dockees are not locked into a single role: "Each dockee can therefore operate in the capacity as a primary, secondary, or both, by having any combination of write and read permissions." Ex-1007, ¶93. The system "allows for more than one primary dockee to be connected to the wireless docking host (WDH) at the same time, and that each primary dockee can both send AV data to the WDH and receive AV data from the WDH." Ex-1007, ¶¶59-60. Van de Laar even describes "transforming a secondary device into a primary dockee device, or a primary dockee device into a secondary device." Ex-1007, ¶60. This is a system designed for fluid role assignment, not rigid categories.

48. From a networking perspective, this flexibility is not surprising. Wi-Fi Direct, which underlies many wireless docking protocols, similarly allows devices to negotiate roles dynamically. See "Wi-Fi Direct," Wi-Fi Alliance. In the proposed

combination, a processing device can exercise both primary dockee capabilities (running Skype, sending data) and secondary dockee capabilities (receiving simulated peripherals from the WDH) simultaneously. Van de Laar's architecture expressly accommodates this.

49. For Claims 9 and 18, which require "the virtual functional device presented to the unified communication," Van de Laar's architecture directly satisfies this limitation. When a dockee receives a simulated peripheral from the WDH, such as a simulated webcam (Ex-1007, ¶54), that simulated peripheral appears to the dockee's operating system as a standard device through the driver layer. Applications running on the dockee, including unified communications applications like Skype, can then access this virtual device through standard APIs. From Skype's perspective, the simulated webcam is indistinguishable from a physical webcam; Skype simply requests video data from what it believes is a camera device, and the operating system routes that request through the appropriate driver. This is how virtual device presentation works in practice—the virtual device is "presented to" applications by appearing in the operating system's device enumeration, where applications like Skype can discover and use it. Van de Laar's teaching that a dockee can both run Skype (Ex-1007, ¶128) and receive simulated peripherals (Ex-1007, ¶¶54, 93) on the same device directly renders obvious a virtual functional device presented to the unified communication application.

50. Regarding the "walled garden" argument, Patent Owner's characterization of Kaplan as requiring pre-paired transmitters and receivers (POR at 31-32), I note that Kaplan describes pre-pairing as occurring "in a particular embodiment." Ex-1008, ¶18. In my experience reading technical disclosures, the phrase "in a particular embodiment" signals that the described feature is one option among several. A POSA would understand that Kaplan's broader teaching of wireless USB transmission between a processing device and a receiver is not limited to configurations that use pre-pairing. During his deposition, Dr. Brogioli confirmed he has "no opinion one way or the other whether Kaplan includes embodiments that do not require a 'walled garden' approach." Ex-1060, 113:6-114:5. His lack of opinion on this point undercuts Patent Owner's assertion that Kaplan is limited to a "walled garden" configuration.

51. I also want to address Kaplan's use of standard operating systems and drivers. Kaplan discloses that the processing device runs "standard operating systems including Windows®." Ex-1008, ¶¶47-49. Standard operating systems like Windows necessarily use drivers to communicate with peripheral devices, whether those peripherals are physical or virtual. This is not a feature that needs to be expressly recited in a patent disclosure; it is inherent to how operating systems function. When Kaplan describes automatic software installation and USB connectivity (Ex-1008, ¶¶17, 45, 47-49), a POSA would understand that the

Windows operating system loads appropriate drivers to interface with the connected devices. Kaplan also acknowledges that "compression/decompression algorithms (codecs) are available on the computer" and that "resident software on the computer" determines codec selection. Ex-1008, ¶48. This confirms that the processing device running Windows performs data transformations through its standard software infrastructure, which necessarily includes the driver layer.

52. I want to elaborate on why Van de Laar's flexible architecture supports the proposed combination. During his deposition, Dr. Brogioli confirmed that dockees "can switch roles between primary and secondary and vice versa." Ex-1060, 85:17-92:24. He also acknowledged Van de Laar's paragraph 59, which states that "[o]ptionally, the docking processor is arranged for docking multiple primary dockee devices." When confronted with this disclosure, Dr. Brogioli could not dispute that Van de Laar contemplates multiple devices both sending and receiving data simultaneously. This is exactly the architecture the proposed combination requires: a processing device operating as a primary dockee (with control privileges and Skype functionality) while simultaneously receiving simulated peripherals (acting as a secondary dockee for those particular data streams). There is no technical barrier to this dual functionality—Van de Laar's system was designed for precisely this type of flexibility.

53. Regarding the claim limitation that "the processing device [is] configured to host the unified communication between two or more other processing devices," I want to clarify the technical architecture. Van de Laar discloses that primary dockees run unified communications applications like Skype locally on the processing device. Ex-1007, ¶128. In this architecture, the WDH facilitates data transmission between the dockee and functional devices (such as cameras and microphones), but it does not "host" the unified communication session. The UC application runs on the dockee's processor, manages the call state, handles codec selection, and coordinates with remote participants. The WDH is a pass-through for functional device data, not a call controller. A POSA would understand that "hosting" a unified communication refers to running the UC application that manages the session—exactly what Van de Laar's primary dockee does when it runs Skype.

54. I also want to address the claim limitation regarding data transformations being performed by "client software" on the processing device. Patent Owner argues that in Van de Laar and Kaplan, scaling and transcoding occur at the WDH rather than on the processing device. This argument reflects a misunderstanding of how multimedia systems operate in practice. While the WDH may perform certain transformations for bandwidth management (Ex-1007, ¶126), the processing device running a unified communications application like Skype

obviously performs its own data transformations. When Skype or any similar VoIP application receives encoded audio or video data, it must decode that data before rendering it to the user—this decoding is a data transformation performed by the client software. Similarly, when Skype captures audio from a virtual microphone or video from a virtual camera, the application encodes that data before transmission—another transformation. These transformations are fundamental to how unified communications applications function. A POSA would understand that regardless of what processing the WDH performs, the dockee device running Skype will perform additional type-specific transformations because that is how UC applications work. Kaplan expressly acknowledges this reality by disclosing that "compression/decompression algorithms (codecs) are available on the computer" and that resident software on the computer determines which codecs to use. Ex-1008, ¶48.

IX. OBSERVATIONS REGARDING DR. BROGIOLI's TECHNICAL OPINIONS

55. I want to share several observations about Dr. Brogioli's technical analysis that have informed my own evaluation of his opinions.

56. During his deposition, Dr. Brogioli exhibited a pattern of declining to engage with technical questions that fell outside the precise scope of his written declaration. When asked about basic aspects of the prior art references he had opined

on, he repeatedly answered that he had "no opinion" or did "not recall discussing" the topic. Ex-1050, 85:2-19, 97:1-11. In my experience, a technical expert with genuine familiarity in a field would be able to discuss fundamental concepts in that field even if those specific concepts were not addressed in a written report.

57. I was also struck by an inconsistency in Dr. Brogioli's testimony regarding endpoints. He testified in a related proceeding that a mass storage device was not an "endpoint" because the patent at issue did not use that word, but he also testified that a mass storage device is a type of endpoint. Ex-1050, 41:17-42:13, 43:22-45:6; Ex-1060, 106:18-107:9, 108:6-109:6. In my view, the answer to whether a mass storage device is an endpoint should not depend on which patent is being discussed—it is a technical question with a technical answer.

58. Most significantly, Dr. Brogioli's assertion that "operating a simulated peripheral does not necessarily require using a driver" (Ex. 2004, ¶152) is, in my view, inconsistent with how operating systems have functioned for decades. As I explained in Section V, drivers are the mechanism through which operating systems interact with all devices, whether physical or virtual. I am not aware of any modern operating system that presents a device to application software without a driver in the communication path. This is thoroughly documented in standard references such as the USB 2.0 specification and Microsoft's WDM documentation, and it is a concept taught in introductory operating systems courses. Dr. Brogioli is taking

positions that are clearly contradicted by published and well-understood technical principles.

59. Dr. Brogioli's deposition also revealed significant admissions that support Petitioner's positions. When questioned about Beel's paragraph 254, which describes "[f]urther compression and/or scaling of the arbitrary media content to allow use of low bandwidth connection," Dr. Brogioli confirmed that this describes data transformations: "It looks like it's saying there may be somebody—a client operating device from 253 that may be on a low bandwidth connection so they can't get as many bits effectively at a certain rate, for example, and so you may want to compress the media content that's being sent to them or scale it—probably scale it down, I think it's referring to." Ex-1050, 91:16-93:11. When asked whether "scaling is a form of compression," Dr. Brogioli agreed it was "compression and scaling or scaling" and confirmed that "changing a resolution" is an example of such scaling. Ex-1050, 92:9-94:2. These admissions directly support Petitioner's position that Beel teaches data transformations for accommodating different bandwidth requirements.

60. Dr. Brogioli further confirmed Beel's disclosure of virtual device presentation when questioned about paragraph 317. He testified that Beel "talks about scraping audio, like the prior art. And capturing audio on the peripheral device and then that peripheral device preferably acts as a composite device comprising, for

instance, a virtual audio speaker." Ex-1050, 116:12-118:8 He also confirmed the data flow: "The peripheral device captures the audio stream with the device driver and then that is streamed—or the peripheral streams the audio to the base unit." Ex-1050, 117:1-118:8. These admissions align with my analysis of how Beel implements virtual device presentation through software executing on the processing device.

61. Additionally, Dr. Brogioli confirmed that unified communication between processing devices occurs through client software, as the claims require. When asked about the communication between processing devices 160-3 and 160-4 through "unified communication software," Dr. Brogioli agreed: "That looks like what the first and second sentence of this paragraph in Column 18 seem to be referring to." Ex-1050, 68:7-69:21. This testimony supports the combination of Beel and Dinka for teaching unified communication functionality.

62. Finally, Dr. Brogioli's lack of familiarity with the accused products undermines Patent Owner's secondary considerations arguments. When asked whether he had performed a "side-by-side analysis" of the ClickShare button in Beel's Figure 10 versus the modern conference version, Dr. Brogioli admitted: "I don't recall performing a side-by-side analysis of those different devices." Ex-1050, 143:17-144:10. He also could not identify the version of ClickShare button depicted in his declaration or explain whether the conference version includes the same

internal components. Ex-1050, 141:11-144:19. Without this analysis, Patent Owner cannot establish a nexus between any alleged commercial success and the specific claims of the ’346 patent.

63. Dr. Brogioli’s more recent deposition (Ex-1060) in this ’346 IPR proceeding revealed additional significant admissions. When asked about Beel’s fourth embodiment, Dr. Brogioli acknowledged that the portable application “goes on the client operating device” and confirmed there is “no plug-and-play port such as the USB port...required on the client operating device.” Ex-1060, 45:6-46:1. He further confirmed that in Beel’s third embodiment, “the execution of the software would ultimately be the execution of instructions on the Intel processor in the...Windows laptop,” and when asked whether the instructions “would be executed on the processor of the processing device,” he answered: “As far as I recall, that sounds correct.” Ex-1060, 49:15-50:1. These admissions technically contradict his written opinion that peripheral-device features cannot be attributed to peripheral-less embodiments. Ex. 2004, ¶¶99-103.

64. Dr. Brogioli also exhibited the same pattern in the more recent deposition. When asked whether Figure 7 of Beel illustrates a peripheral device—even though Beel paragraph 100 explicitly states “Figure 7 shows a peripheral device”—Dr. Brogioli could not give a yes-or-no answer despite extended questioning. Ex-1060, 38:18-44:9. He also maintained contradictory positions on

endpoints: he testified that mass storage devices are endpoints under the USB specification (Ex-1060, 97:10-98:1, 114:17-22), yet insisted Beel's mass storage device is not an endpoint because "Beel does not mention endpoints at all." Ex-1060, 106:2-108:25. This double standard was further exposed when he testified that ClickShare's webcam is an endpoint "exposed via a USB" (Ex-1060, 146:1-3), but could not explain why Beel's webcam would not also be an endpoint—even though ClickShare also does not use the word "endpoint." Ex-1060, 146:15-147:12.

65. Critically, Dr. Brogioli confirmed key technical teachings that support Petitioner's combination. He acknowledged that Beel's "video encoder can encode video data using a video codec, and audio data using an audio codec" and that "a client PC...could run an audio encoder" to "encode a .wav file to mp3." Ex-1060, 77:5-25. He also confirmed that pre-installed generic drivers for WiFi interfaces were well-known at Beel's priority date and that "the person of skill in the art would be aware of the use of these types of drivers." Ex-1060, 63:5-64:19. When confronted with Dinka's disclosure of "a plurality of computer terminals 102" connected for VoIP communication, Dr. Brogioli acknowledged Dinka "talks about...computers connected...over a packet-based computer network such as the internet" for voice and video calls. Ex-1060, 102:4-24.

66. Perhaps most difficult to reconcile with applying the knowledge of a POSA, he admitted having "no opinion" on fundamental technical questions that any

POSA in this field, let alone an expert in the field, should be able to address. He testified he had no opinion on: (1) whether operating systems present devices to applications without using drivers (Ex-1060, 78:1-12, 149:14-150:1); (2) what "translating" means in the context of the '346 patent (Ex-1060, 125:6-25); (3) what "interpreting" or "encoded" mean (Ex-1060, 126:1-16, 132:5-133:20); (4) whether endpoints must be either fixed or configurable (Ex-1060, 115:16-116:23); and (5) what technical challenges a POSA would face using a WiFi interface (Ex-1060, 60:1-62:5). When confronted with his opinion that "operating a simulated peripheral does not necessarily require using a driver," he could not "identify any real-world operating systems where a simulated peripheral is presented to an application without any driver involvement." Ex-1060, 149:18-150:1.

67. Finally, Dr. Brogioli testified that he could not recall ever using a dongle to share content in a conference room before this litigation. Ex-1050. The technologies at issue in this proceeding, *e.g.*, wireless conferencing systems, virtual device presentation, and unified communications applications, are technologies I have used, researched, and taught about throughout my career, as documented in my curriculum vitae (Ex-1003) and original Declaration (Ex-1002, ¶¶5-30).

X. ADDITIONAL TECHNICAL ANALYSIS FOR SPECIFIC CLAIM LIMITATIONS

A. Claims 21 and 22: "Presenting the First Display Device to the Processing Device"

68. I understand that Patent Owner argues the Petition improperly relies on peripheral device embodiments for Claims 21 and 22, which recite "presenting the first display device to the processing device." Patent Owner contends that peripheral device features cannot be applied to the peripheral-less embodiments claimed in the '346 Patent. In my view, this argument fails for several technical reasons.

69. First, as I explained in Section IV, software functionality is independent of storage medium. When Beel's fourth embodiment teaches that software is "copied on the client operating device" with that being the "only difference" from prior embodiments, a POSA would understand that all software capabilities—including those related to presenting display devices—are preserved. The '346 patent specification itself describes multiple embodiments where client software presents functional devices to the processing device without requiring a peripheral device. In Figures 12-14, the specification describes scenarios where "processing device 160' has no first peripheral device 130" and "client software 70' can be adapted to interact directly with base unit 100 over a wireless connection 127'." The specification further explains that "the client software 70' is adapted to provide any simulations, or translations as required to present the display device 126 to the processing unit

160' and to allow interaction therewith." This is precisely what Claims 21 and 22 require: presenting the first display device (display 126) to the processing device through client software functionality.

70. Second, Beel's disclosure supports this same architecture. Beel paragraphs 71-72 describe "executable software code...stored for execution on the processing device." The actual software execution—including presenting devices to the operating system—occurs on the processing device, not on the peripheral device. The peripheral device is merely a storage and transport mechanism for the software. Once the software is loaded and launched on the processing device, the peripheral device's role in device presentation is complete. Beel's description of its third embodiment, which uses a "regular USB memory stick" with "no input device, visual indicator, or transmitter," confirms that the system can operate without the peripheral device's active participation after software loading. In this embodiment, "the system needs to use the transmitter/receiver from the client processing device" for wireless communication, but the software executing on the processing device still performs its functions.

71. Third, the combination with Dinka reinforces bidirectional communication enabling display data to flow to the processing device. Dinka describes bidirectional video communication where users can "establish a voice or video call over the Internet" with data flowing in both directions. In the combination,

display data from the base unit's display (the "first display device") flows to the processing device through this bidirectional communication channel. The processing device's client software then presents this display data to applications running on the processing device. This is not merely a "cursory" response to Patent Owner's argument—it is a complete technical explanation of how the combination satisfies the claim limitation.

72. Fourth, Patent Owner's characterization of the Reply's response as merely a "cursory" cross-reference to Section II.A is inaccurate and misleading. The technical explanation provided encompasses multiple interconnected disclosures: (a) Beel's express teaching that software functionality is preserved when copied between storage media (paragraph 247); (b) the '346 patent's own description of peripheral-less embodiments in Figures 12-14; (c) Dinka's bidirectional communication architecture; and (d) Dr. Brogioli's deposition admissions confirming that software execution occurs on the processing device regardless of the software's original storage location. This multi-faceted technical analysis directly addresses the specific limitation in Claims 21 and 22.

B. Claim 6: "Decoding and Compositing or Decrypting the Video Data"

73. Claim 6 recites that "the data transformations of the video data comprises decoding and compositing or decrypting the video data." Patent Owner

argues that Beel does not disclose these specific video data transformations and improperly relies on peripheral device communications. In my view, Beel's disclosure of video processing satisfies this limitation.

74. Beel expressly describes video decoding and compositing operations. Figure 11 of Beel illustrates a base unit 33 that receives encoded video data from the processing device, and the base unit includes a "video decode" block 26 and a "compositor" block 29. The specification explains that after video data is transmitted through the communications network, "the incoming stream is read from the communications interface such as a WiFi access point 22, unpacked in an unpacker 25, decoded in a decoder 26 and then inserted into a suitable composition such as an OpenGL based composition in the compositor 29 for display on a central display device." This is precisely "decoding and compositing" the video data. Dr. Brogioli confirmed this understanding during his deposition, testifying that Beel paragraph 322 "describes the reading from the communications interface, such as the WiFi access point 22, unpacked in an unpacker 25, and decoded in decoder 26. And then there's the composition that happens after that." He further acknowledged that "Beel is describing a decoding process in that 26 that may be the inverse of the encoding process."

75. Regarding decryption, Beel paragraph 155 describes "Eighth code for providing means or a method step for receiving, decrypting, and decoding incoming

arbitrary media content." Beel's software on the processing device includes "code for providing a means for receiving, decrypting and decoding incoming arbitrary media content"—which encompasses video data. The '346 patent's own file history confirms that decryption is a data transformation within the scope of this claim limitation. During prosecution, Barco stated that "transformations can include encoding, decoding, mixing, and demultiplexing." Decryption is the inverse operation of encryption, just as decoding is the inverse of encoding. A POSA would understand that decryption falls within the same category of transformations.

76. As for Patent Owner's argument about "directional flow" of video data, I addressed this concern in Section VIII.B. Beel's base node receives video from functional devices like cameras and transmits that video to processing devices through its transceiver—this is bidirectional data flow. The video received by the processing device can then be decoded, composited, or decrypted by client software running on that device. Claim 6 does not require the video data to originate from any particular source; it requires only that video data received from the base unit undergoes the specified transformations. The combination of Beel and Dinka satisfies this requirement by establishing bidirectional video communication and implementing type-specific video transformations.

77. Moreover, Claim 6 specifically recites data transformations comprising "decoding and compositing or decrypting" as alternatives connected by "or," a

POSA would understand that the claim is satisfied by any of these operations, not necessarily all of them simultaneously. Beel expressly discloses both decoding and compositing operations performed on video data at the base unit. The video decoder 26 performs the decoding operation, transforming encoded video (such as H.264-encoded data) back into a displayable format. The compositor 29 performs the compositing operation, combining multiple video sources into a unified display output suitable for the central display device. These are textbook examples of "data transformations" of video data. Dr. Brogioli's deposition testimony confirms this understanding, he acknowledged that Beel describes "a decoding process in that 26 that may be the inverse of the encoding process" and that "there's the composition that happens after that." Ex-1060, 73:1-9. Patent Owner's argument regarding the "directional flow" of video data conflates where transformations occur with what transformations occur; the claim language does not specify directionality requirements beyond what the combination satisfies.

C. Ground 2: Additional Technical Analysis

78. Patent Owner raises several specific arguments regarding Ground 2 (Kaplan-Van de Laar-Christison) that I want to address in detail: (1) that neither Kaplan nor Van de Laar mention the word "driver"; (2) that the Petition fails to elect a specific transmission technique; and (3) that Skype and Miracast are incompatible protocols that would not be implemented together.

79. Regarding the "driver" argument, I addressed this extensively in Section V, but I want to add specific analysis for the Ground 2 references. Patent Owner's argument that "neither Kaplan nor Van de Laar mention the word 'driver'" fundamentally misunderstands how prior art disclosures work. A prior art reference need not explicitly use every technical term to disclose a concept that a POSA would understand as inherent or obvious. The '346 patent itself acknowledges that "a driver is standard for the operating system and can drive a standard class of peripheral devices." Ex-1001, 8:54-56. When Kaplan discloses that the processing device runs Windows operating systems, a POSA would immediately understand that Windows uses drivers to interface with peripheral devices—this is how Windows has operated since before the relevant time period. Ex-1008, ¶47. Dr. Brogioli confirmed this understanding during his deposition, testifying that "modern versions of Windows include drivers" and that a driver can be "used for the OS to drive a standard class of peripheral devices." Ex-1060, 152:4-15. When Van de Laar describes "simulated peripheral" devices appearing to dockees, a POSA would understand that this appearance is accomplished through the dockee's driver layer—because that is the only mechanism through which devices, whether physical or virtual, can be presented to an operating system. Ex-1007, ¶¶93-94.

80. Regarding the transmission technique argument, Patent Owner contends that the Petition fails to articulate whether the combination uses Kaplan's

peripheral USB transmitter or Van de Laar's integrated Wi-Fi transmitter. In my view, this argument misunderstands how obviousness combinations work. The Petition does not require a single, fixed implementation. A POSA would have understood that either transmission approach could be used in the combination, depending on the particular implementation requirements. Beel itself teaches multiple transmission approaches: some embodiments use a peripheral device with an integrated transmitter (the "dongle" embodiments), while others use the processing device's built-in wireless interface (the third and fourth embodiments). Ex-1005, ¶¶241-247. Dr. Brogioli confirmed during his deposition that Beel teaches "two embodiments, one using the WiFi interface of the dongle to communicate with the base unit. And then second, using the WiFi interface of the processing device to communicate with the base unit." Ex-1060, 61:1-17. He also acknowledged that pre-installed generic drivers for WiFi interfaces were well-known at Beel's priority date. Ex-1060, 63:5-64:2. A POSA would have recognized that Van de Laar's integrated Wi-Fi approach and Kaplan's peripheral transmitter approach are both viable implementations of wireless communication between a processing device and a host/receiver. The choice between them is a design decision based on factors like device portability, cost, and complexity—not a fundamental technical barrier that would render the combination "unworkable."

81. To be clear, the Petition's obviousness analysis does not depend on selecting one transmission technique to the exclusion of another. I understand a POSA is not an automaton but rather a person of ordinary creativity who can combine known elements according to known methods to yield predictable results. Both Kaplan's peripheral-based wireless transmission and Van de Laar's integrated Wi-Fi transmission were known techniques for accomplishing wireless communication between a processing device and a display host. A POSA seeking to implement the claimed functionality would have understood that either approach, or a combination, could achieve the desired result. The choice between them involves routine design considerations such as whether the processing device has built-in Wi-Fi capability, whether the user prefers a dongle-based solution for portability across different devices, or whether bandwidth requirements favor one implementation over another. None of these considerations would have posed any technical barrier to a POSA. Patent Owner's demand that Petitioner "elect" a single transmission technique imports a limitation into the obviousness analysis that has no basis in law or fact.

82. Regarding the Skype and Miracast protocol compatibility argument, Patent Owner argues that Van de Laar's use of Skype is an alternative to using Miracast—not a simultaneous use—and that "Skype and Miracast are different protocols" that would not be implemented together. From a technical perspective,

this argument reflects a misunderstanding of how unified communications systems operate. Skype and Miracast serve different functions in a wireless docking system. Miracast is a display mirroring protocol, it transmits video and audio content from a source device to a display. Skype is an application-layer communication protocol, it handles voice and video calling, presence information, and messaging. These protocols can and do coexist on the same device because they operate at different layers of the communications stack and serve different purposes.

83. Van de Laar paragraph 128 describes using "WSB Audio, Wi-Fi Direct Play Service, VoIP, XMPP Skype, DLNA or other audio streaming mechanism." Ex-1007, ¶128. The phrase "or other audio streaming mechanism" indicates these are options for audio transmission, but it does not mean a dockee device cannot support multiple protocols simultaneously. Modern computing devices routinely run multiple network protocols concurrently; a laptop can simultaneously use Wi-Fi for internet access, Bluetooth for audio streaming, and a wired USB connection for data transfer. Similarly, a dockee device running Skype can simultaneously use Miracast for display mirroring to a meeting room display. The claim limitation does not require Skype and Miracast to be used "simultaneously" in the sense of transmitting the same data through both protocols; it requires only that the processing device can host a unified communication (using Skype) while also presenting virtual functional

devices to that communication (which can be accomplished through the docking protocol, whether Miracast-based or otherwise).

84. I want to provide additional technical context regarding why Skype and Miracast can coexist on a single device. From a networking architecture perspective, Skype operates primarily at the application layer (Layer 7 of the OSI model), using TCP and UDP at the transport layer for voice and video communication with remote endpoints. Miracast, by contrast, operates as a peer-to-peer Wi-Fi display technology using Wi-Fi Direct for local device-to-display communication. These protocols do not compete for the same resources in a way that would create technical incompatibility. A laptop running Windows can simultaneously (1) maintain a Skype video call over its internet connection, (2) mirror its display to a Miracast-compatible television, and (3) stream audio to Bluetooth headphones—this is standard functionality that millions of users employ daily. Van de Laar's architecture contemplates precisely this type of multi-protocol operation, with the WDH managing different peripheral connections while dockees run UC applications locally. The Columbia University analysis of Skype's protocol architecture confirms that Skype's design allows it to operate concurrently with other network protocols. Ex-1057. Patent Owner's suggestion that Skype and Miracast represent mutually exclusive alternatives ignores how modern operating systems and applications actually function.

85. Regarding the WDH "hosting" argument, Patent Owner contends that in Van de Laar, the WDH (wireless docking host) hosts the unified communication call rather than the processing device, because the WDH receives data from the primary dockee, mixes audio, and broadcasts to secondary dockees. I addressed this point in Section VIII.D, but I want to elaborate on the technical architecture. In Van de Laar's system, the WDH acts as a hub for data distribution, but it does not "host" unified communications in the sense the claims require. When a primary dockee runs Skype locally, the Skype application executes on the dockee's processor, manages call state, handles codec selection and negotiation, and coordinates with remote Skype users through Skype's peer-to-peer or server-based infrastructure. The WDH facilitates local data routing—for example, mixing audio from multiple sources or distributing video to secondary dockees—but this is fundamentally different from hosting the unified communication session. Ex-1007, ¶¶96-97 (describing WDH creating and providing video and/or audio streams to secondary dockees).

86. An analogy may be helpful: consider a conference call using a traditional PBX phone system. The PBX routes calls between internal extensions and to external lines, but we would not say the PBX "hosts" a particular phone call—the participants and their phones host the call by maintaining the conversation. Similarly, Van de Laar's WDH routes data between devices in the wireless docking system, but the dockee running Skype hosts the unified communication session. Van

de Laar paragraph 128 confirms this architecture by describing how the WDH "enable[s] primary dockees to provide output to the WDH and/or its audio peripherals through an audio stream" using protocols including "VoIP" and "XMPP Skype." Ex-1007, ¶128. The Skype session originates from and is controlled by the primary dockee; the WDH merely facilitates the routing of audio and video data to local peripherals and other dockees. Dr. Brogioli confirmed during his deposition that primary dockees "have control" while secondary dockees "just receive the rendered AV data." Ex-1050, 101:12-22. This distinction supports the conclusion that the processing device (operating as a primary dockee) hosts the unified communication, with the WDH serving as a routing and distribution layer.

87. The technical architecture of unified communications further supports this understanding. When Skype or similar VoIP applications establish a call, the application manages several critical functions: (1) session initiation and teardown using protocols like SIP or proprietary signaling; (2) codec negotiation to determine audio and video encoding parameters; (3) NAT traversal to establish peer-to-peer or relayed connections; (4) quality-of-service monitoring and adaptation; and (5) user interface for call management. These functions execute on the processing device's CPU and require the processing device to maintain the call state. The WDH in Van de Laar performs none of these functions—it acts as an audio/video routing hub for local peripherals. Van de Laar expressly describes dockees running "applications"

locally, including "a docking application" that interfaces with the WDH peripherals. Ex-1007, ¶¶120-122, 128. When a primary dockee runs Skype, the Skype application is the "host" of the unified communication session in every meaningful technical sense: it initiates the call, maintains the connection, processes the signaling, and terminates the session. The WDH merely provides access to higher-quality microphones and speakers than those built into the dockee device—exactly what the claimed invention contemplates.

D. Commercial Success: Analysis of Feature Attribution

88. Patent Owner presents evidence of commercial success of ClickShare products. Petitioner has argued that ClickShare includes significant unclaimed features, including Teams and Zoom support. I want to provide additional technical context regarding which features may drive commercial success in this product category.

89. In my experience with wireless conferencing systems, several factors influence purchasing decisions beyond the basic claimed features. First, integration with popular collaboration platforms like Microsoft Teams and Zoom has become a critical feature in recent years. Organizations adopting wireless conferencing systems often prioritize compatibility with their existing unified communications infrastructure. ClickShare's Teams and Zoom integration represents substantial engineering effort beyond the basic wireless presentation functionality. Ex-1061.

Second, security features such as encryption, authentication, and compliance certifications are important to enterprise customers. Third, management capabilities including cloud-based device management, firmware updates, and usage analytics add value beyond the core functionality. Fourth, user experience factors such as button design, setup simplicity, and visual feedback influence adoption within organizations. None of these features are recited in the challenged claims.

90. From a technical standpoint, I would note that the core features claimed in the ’346 patent—wireless communication between a processing device and a base unit, presenting virtual functional devices to the operating system, and performing data transformations based on data type—were each individually well known before the claimed priority date, as I have explained throughout this declaration. The fact that a product implementing these features achieves commercial success does not necessarily establish that the success results from the specific combination claimed in the patent, rather than from unclaimed improvements, marketing, brand recognition, or other factors. Without a rigorous technical analysis identifying which specific features drive customer purchasing decisions—which Dr. Brogioli acknowledged he did not perform—I do not believe the commercial success evidence establishes a nexus to the specific claims at issue. Dr. Brogioli confirmed during his deposition that he had not performed a “side-by-side analysis” comparing

ClickShare's claimed features to its unclaimed features, nor could he identify which features specifically drive customer purchases.

91. I want to elaborate on the lack of nexus evidence with specific examples. Modern ClickShare products, as documented in Barco's own marketing materials, prominently feature integration with Microsoft Teams and Zoom—functionality that is not recited in the challenged claims. During Dr. Brogioli's deposition, he acknowledged that he had not performed any analysis of "what additional features might be driving the commercial success of the Barco ClickShare base unit." Ex-1060, 143:1-5. He further confirmed that he had not done "a breakdown of features driving commercial success" for either the '346 or '347 patents. Ex-1060, 143:5-9. When shown a ClickShare product document referencing "smooth experience with Microsoft Teams Resume," Dr. Brogioli admitted he had not analyzed what that functionality entails or whether it relates to the claimed features. Ex-1060, 141:16-142:8. Without this analysis, Patent Owner cannot establish that the commercial success of ClickShare products results from the specific claimed combination rather than from unclaimed features like Teams/Zoom integration, cloud management capabilities, or standalone operation modes. The Supreme Court has made clear that commercial success evidence requires a nexus to the claimed features, not merely to a product that happens to practice the claims among many other features. Petitioner's identification of significant unclaimed

features—which Dr. Brogioli did not analyze—shifts any burden back to Patent Owner to establish which features actually drive purchasing decisions.

XI. SECONDARY CONSIDERATIONS

92. Section X.D contains additional opinions as it relates to secondary considerations. In addition to those opinions, I offer more opinions here.

93. Patent Owner points to a Crestron license agreement and alleged commercial success of ClickShare products as evidence of non-obviousness. POR at 58-62. I am not offering a legal opinion on the weight of this evidence, but I can offer relevant technical observations.

94. From a technical standpoint, Barco's ClickShare product line encompasses a wide range of features that go well beyond the specific claims of the '346 patent. Modern ClickShare products support integration with platforms like Microsoft Teams and Zoom, can operate with or without dongles, provide cloud management capabilities, and include numerous other sophisticated features. In my view, it is difficult to attribute the commercial performance of a product with this many features to any single patent's claims, particularly when the core underlying technologies being wireless communication between a processing device and a base unit with connected functional devices, virtual device presentation, and type-specific data transformations, were, as I have explained, well understood and widely implemented before the claimed priority date.

95. Regarding the Crestron license, I understand it covers multiple patents from multiple patent families and includes international rights. Without a way to isolate what portion of the license value, if any, is attributable to the specific claims of the ’346 patent, I do not believe the license tells us much about whether those specific claims would have been obvious to a POSA.

XII. CONCLUSION

96. For the reasons set forth above and in my original Declaration (Ex-1002), it is my view that all challenged claims of the ’346 patent would have been obvious to a POSA at the time of the alleged invention. After carefully reviewing Dr. Brogioli’s declaration and deposition testimony, I find that his technical arguments do not withstand scrutiny. The core technologies at issue, *e.g.*, running software on a processing device, presenting virtual devices to an operating system through drivers, performing type-specific data transformations, and communicating wirelessly between a processing device and a base unit with connected functional devices, were each individually well known and routinely combined in conferencing systems before December 2017.

97. I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the

like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Date: March 31, 2026

By: Kevin C Almeroth
Kevin C. Almeroth, Ph.D.