UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GENIUS SPORTS LTD.,
Petitioner

v.

SPORTSCASTR INC.
(d/b/a PANDA Interactive),
Patent Owner.

IPR2024-01305, -01307, -01308, and -01309
U.S. Patent Nos. 10,805,687 and 10,425,697

SUPPLEMENTAL DECLARATION OF DR. STEPHEN B. WICKER UNDER 37 CFR 1.68 IN SUPPORT OF PETITIONER'S REPLY FOR *INTER PARTES* REVIEW OF U.S. PATENT NOS. 10,805,687 AND 10,425,697

I, Dr. Stephen B. Wicker, declare as follows:

I. INTRODUCTION

1. My name is Stephen B. Wicker, Ph.D. I am Professor Emeritus of Electrical and Computer Engineering at Cornell University in Ithaca, New York. I have been retained on behalf of Genius Sports Ltd. to provide my opinions regarding the patentability of three patents involved in Inter Partes Reviews:

IPR2024-01305: involving claims 1-9 of USP 10,805,687
IPR2024-01307: involving claims 19-25 and 27-30 of USP 10,805,687
IPR2024-01308: involving claims 1-9 of USP 10,425,697
IPR2024-01309: involving claims 19-25 and 27-30 of USP 10,425,697
IPR2024-01310: involving claims 1-7 and 24-34 of USP 11,039,218
IPR2024-01311: involving claims 8-23 of USP 11,039,218

2. Below I provide my supplemental opinions responsive to the institution decisions, patent owner replies, and the supplemental declarations of Dr. Avi Rubin (Ex.1041) provided on behalf of Patent Owner in IPR2024-01305, -01307, -01308, and -01309. In preparing this declaration I have reviewed each of these documents, the documents cited therein, and the deposition transcript of Dr. Rubin, Ex.1063. My supplemental opinions directed to IPR2024-01310 and -01311 are provided in a separate declaration.

II. SPIVEY DISCLOSES THE CLAIMED EVENT SOCKETS

A. Spivey Discloses an MQSD-Client Connection

3. Patent Owner argues that Spivey does not disclose or suggest that its Message Queue Server ("MQSD") / topic queues are capable of handling communication to individual client devices because Spivey's user interfaces "all pertain to accessing the live data access server." Patent Owner Response, Paper 14 ("POR"), 66-67. That is incorrect. Spivey does not require the live data access server ("LDAS") to serve as an intermediary between the MQSD and the Client. As I explained in my first declaration, in Spivey the MQSD may be "accessible to clients over the network directly." *See* Ex.1001, ¶354.

4. Spivey states that the "MQS communication component 308 may also receive requests to subscribe to topics in a topic set 330, and may push data from topic queues 332-1, 332-n to the subscribers of the topics corresponding to those topic queues" (Ex.1017, ¶46), indicating that the MQSD can deliver data directly to subscribers without the necessity of an LDAS. Furthermore, Spivey clarifies that the system "may include a[n] [LDAS]" (*id.*, ¶38), not that it must, underscoring the optional nature of the LDAS structural component. As discussed below with reference to Fig. 22 of Spivey, the functionality of the LDAS can be provided by a server functioning as both the MQSD and the LDAS.

5. Spivey notes that the time from data receipt at the MQSD to delivery to the client device can be as little as 100 milliseconds or less (*id.,*, ¶96). Spivey's explicit focus on low-latency design further supports my conclusion that the MQSD alone is sufficient to achieve the real-time performance goals of Spivey, rendering the LDAS an optional, non-essential component. The MQSD receives live data from various sources, immediately sorts it into topic queues based on topic labels, and pushes the data payloads to subscribers without requiring intermediary storage or additional processing layers. *See id.,*, ¶46. The use of the term "may" in reference to the LDAS and its role as a "public interface" (*id.,* ¶38) further emphasizes that the LDAS is not essential to the core low-latency data flow, but rather serves as an optional enhancement for client access or security.

6. The architectural flexibility of Spivey is exemplified in the distributed system embodiment depicted in Figure 22, where the server device 2250 is explicitly described as being capable of implementing the functionality of the live data access server (LDAS) 130, 1430, the message queue server 120, 1420, and/or the aggregator host server 1450 (*see id.,* ¶158). This design enables a single server to consolidate the roles of both the LDAS and MQSD, thereby supporting configurations in which client devices communicate directly with the MQSD without the need for a separate LDAS component. Such an arrangement underscores the modularity and adaptability of the system, allowing for streamlined deployments

that can optimize for performance, resource utilization, or security requirements as needed.

7. A POSA would have readily understood from Spivey that the message MQSD is configurable with a communication component that functions as both an interface and a subscription component, enabling direct connections between the MQSD and client devices. Spivey explicitly states that the MQSD "may include a message queue server (MQS) communication component 308. MQS communication component 308 may include hardware and/or software that allows MQS device 310 to transmit and receive electronic data, for example live data, with other devices, such as live data access server 130" and further notes that "MQS communication component 308 may also receive requests to subscribe to topics in a topic set 330, and may push data from topic queues 332-1, 332-n to the subscribers of the topics corresponding to those topic queues." *Id.,* ¶46.

8. Spivey's system is described as "maintain[ing] a full-duplex connection, such as a websocket connection with a client device 110," which "typically comprises a two-way connection between two entities, where each of the two entities can send and receive data simultaneously" (*id.*, ¶61). This architecture, where the MQSD manages topic subscriptions and pushes data directly to clients, is further reinforced by the depiction of the MQSD as the central hub for both receiving live data and distributing it to subscribers. *Id.*, ¶46 and Fig. 3. These disclosures

make clear that the MQSD's communication component is configurable to act as both the interface for direct data exchange and the subscription manager, allowing for direct, low-latency MQSD-client connections.

9. Furthermore, the claim set presented in Spivey confirms that Spivey contemplated direct MQSD Client connections. For example, Spivey's method claim 7 requires a message queue server and client device, but not an LDAS. *Id.,* Cl. 7. Claim 7 recites a "connection" between the "requested topic at a [MQSD]" and the "client device."

10. Overall, I believe that Patent Owner's arguments adopt an overly restrictive read of Spivey, effectively requiring the literal and complete incorporation of Spivey's architecture into Ellis, rather than considering what the combined teachings would have reasonably suggested to a POSA. Rather than evaluating how the core concepts and mechanisms disclosed in Spivey—such as real-time data distribution via message queue servers—would have been adapted or integrated in Ellis, Patent Owner insists that only the precise, unaltered components and message flows of Spivey should be considered. By focusing narrowly on whether Spivey's LDAS is a mandatory or optional component, Patent Owner fails to address the broader technical question of how a POSA would have combined the teachings of both references.

B. Spivey's Topic Queues Are Uniquely Addressable Communication Endpoints that Transmit Event Information

11. PANDA argues that Spivey's topic queues "lack an address through which they can communicate" and thus cannot establish MQSD-Client connections. POR, 6; *see also id.,* 27. As I explained in my first declaration, Spivey's topic queues are uniquely addressable via URLs that include topic labels as part of the publicly addressable URLs, and thus are communication endpoints for multiple client devices. Ex.1001, ¶¶352, 354, 355, 357, 358. A POSA would have known that the URL includes a domain name that can be resolved into an IP address through the DNS process.

12. In his deposition, Dr. Rubin concedes that, according to the '687 specification, a URL is sufficient to make an endpoint uniquely addressable, stating, "[t]he URL is sufficient for that" when asked if an event socket is a uniquely addressable communication endpoint. Ex.1063, 48:6-8. Spivey itself states, "[a] topic label may be, for example, a text string such as gamecenter.nfl.ravenssteelers" (Ex.1017 ¶66), and that client devices "request a live data connection" by selecting or entering the address or connection information which "may resemble a uniform resource locator (URL)." *Id.,* ¶82.

13. Spivey's topic queues are not only logically but also functionally implemented as uniquely addressable communication endpoints because each topic queue corresponds to a specific topic label that is parsed from incoming live data

entries and used to route data payloads to the appropriate queue. *See id.*, ¶49 and ¶103. The message queue server maintains a collection of these topic queues (topic set 330), and each queue is directly associated with a unique topic identifier, enabling precise subscription and data delivery mechanisms. When a client or live data access server subscribes to a topic, the message queue server registers this subscription to the corresponding topic queue, ensuring that only data relevant to that topic is pushed to the subscriber. *See id.,* ¶78. This architecture allows the message queue server to efficiently manage and distribute live event data by treating each topic queue as a distinct communication endpoint, which can be individually addressed for both publishing and subscribing operations, and supports hierarchical and granular topic structures for fine-grained data dissemination. *See id.,* ¶52 and ¶104.

14. Spivey further discloses queue data structures (*e.g.,* first-in first-out) and the ability to push data directly to subscribers based on their topic subscriptions, demonstrating that each topic queue functions as a unique, addressable endpoint within the system's communication framework. *See id.,* ¶105.

15. Spivey further discloses that the topic queue connections are bidirectional, websocket connections that transmit event information. *See id.,* ¶¶43, 61);*see also id.,* Cl. 8 ("[T]he connection is a websocket connection"), Cl. 11 (transmitting a "play, a score, … a statistic," etc.).

C. Spivey's Disclosure that the MQSD Can Be Implemented as a RabbitMQ Confirms that Spivey's LDAS is Optionally a Separate Component

16. Spivey explains that "RabbitMQ facilitates Topic Queues that receive published messages and push the messages to clients that have subscribed to the topic queues." *Id.,* ¶58. RabbitMQ was a well-established message broker that operates by accepting messages and immediately pushing them to client devices, mirroring the behavior of the message queue server device (MQSD) described in Spivey. Ex.1065. In RabbitMQ, messages are routed through queues, which function analogously to Spivey's topic queues. RabbitMQ explicitly supports a one-to-many architecture, allowing multiple client devices to receive data from a single queue. *Id.,* ("many consumers can receive data from one queue"). Clients connect to the RabbitMQ server and receive messages from the queue by establishing a connection and channel, which directly links the receiver to the designated topic queue.

Full `receive.py` code:

```
#!/usr/bin/env python
import pika

connection = pika.BlockingConnection(pika.ConnectionParameters(
        host='localhost'))
channel = connection.channel()

channel.queue_declare(queue='hello')
```

D. Spivey Discloses Bi-Directional Chat Communication (Dependent Claim 5)

17. PANDA argues that the prior art applied in Ground 1B and Ground 2 of the Petition do not teach claim 5's chat limitations because "topic queues are incapable of at least 'receiving first chat information.'" POR, 51, 62. PANDA is incorrect for multiple reasons.

18. Ellis describes chat room functionality as part of enhanced sports media access, where interactive features such as chat rooms are integrated with live digital content streams, allowing users to interact in real time during primary and secondary sporting events. As Ellis states, media content that may be provided by a content source includes "web services (e.g., websites, newsgroups, and chat rooms …). *See* Ex. 1016, ¶40. Spivey, on the other hand, discloses the use of message queue servers (*e.g.,* RabbitMQ) for managing real-time data, a technology routinely used for chat room implementations. *See* Ex. 1017, ¶58. Spivey discloses multiple topic queues per live event, with hierarchical topics, at paragraphs [0052] and [0104], where it describes that topic queues may be hierarchically related (*e.g.,* "gamecenter.nfl" for all NFL data, "gamecenter.nfl.ravenssteelers" for a specific game, and "gamecenter.nfl.ravenssteelers.stats" for statistics about that game), and that a data payload may be placed in multiple topic queues according to this hierarchy.

19. Importantly, as acknowledged by Dr. Rubin, Spivey discloses bidirectional websocket connections between the topic queues and client devices,

which support communications to and from the event sockets. *See* Ex.2001, ¶135 ("The websocket connection may be a two-way connection between two entities, where each of the two entities can send and receive data simultaneously.")

III. A POSA WOULD HAVE BEEN MOTIVATED TO COMBINE ELLIS-SPIVEY

20. I provided a comprehensive and well-reasoned explanation why a POSA would have been motivated to combine the content distribution systems disclosed in Ellis and Spivey, including the reduction in latency in Ellis's supplemental content distribution system. Ex.1001, ¶¶106-115. In response, PANDA contends that I failed (i) to demonstrate that Ellis and Spivey are compatible and (ii) to articulate "why a POSITA would have been motivated to make the necessary modifications." POR, 43. PANDA further asserts that there are critical differences in Ellis and Spivey. I explain next why my first declaration withstands scrutiny.

21. PANDA incorrectly relies on its belief that Spivey does not disclose MQSD-client connections between topic queues and at least one client device. These arguments fail because Spivey does, in fact, teach a system with MQSD-client connections between topic queues and multiple client devices that transmit event information via websocket connections. See Section II.A above.

22. PANDA further contends that Spivey does not contain a database, and therefore a POSA would not have been motivated to combine Ellis-Spivey. POR,

43. This is not true. As I explained in my first declaration, Spivey discloses a database within the MQSD, like the database in Ellis's Supplemental Content Source. Ex.1001 ¶¶360-362. In any event, the presence of a database is not required in Ellis.

23. The presence of a database is not required in Ellis because the system is designed to flexibly handle sports-related content, user preferences, and interactive features through a combination of in-memory data structures, real-time data feeds, and direct communication between user equipment and content or supplemental sources, rather than relying on persistent storage in a traditional database. As described in paragraph 42 of Ellis, supplemental content source 120 may provide real-time sports scores, game summary information, commentary, and other sports-related data to user equipment as a plurality of real-time data feeds, which may be individually accessed or subscribed to by the user equipment. This information can be stored locally, managed in memory, or received dynamically, and the system architecture explicitly allows for the storage and delivery of such data without requiring a centralized or persistent database. The interactive features, such as scoreboards, user profiles, and fantasy league data, are supported by this flexible data delivery and storage approach, enabling real-time updates and user interaction without necessitating a database for the core operation of the invention. *See* Ex.1016 ¶42.

24. PANDA contends that removing an LDAS from Spivey's architecture (*i.e.,* operating environment 400) would not have decreased latency, as I opined. POR, 45 46. For context, Ellis indicates that its goal is to take "full advantage of today's high-end user equipment devices." Ex.1016 ¶5. In my first declaration I explained that a POSA would have been motivated to combine Spivey with Ellis because Spivey discusses "providing a significant advantage over the conventional practices" of data distribution. Ex.1001 ¶109.

25. A POSA would have understood that, by incorporating any of Spivey's embodiments, including operating environment 400 (Fig. 4) or distributed system 2200 (Fig. 22), the latency in a traditional distribution system such as Ellis's would have been improved by leveraging real-time message queue architectures and open duplex communication channels. Spivey explicitly states that "various embodiments increase the speed at which live event data (*e.g.,* sports scores and breaking news), are broadcast to mobile devices by using topic queues and open duplex communication connections, among other innovative techniques," and that these methods "may reduce that delay to an order of milliseconds (ms) (*e.g.,* 100 ms) or less, providing a significant advantage over the conventional practices." Ex. 1017, ¶¶ 27-29. In contrast, Ellis describes a system where "supplemental content source 120 may be any suitable content source configured to provide real-time sports scores and game summary information, tournament brackets and tournament information,

sports commentary, athlete biographies, spread and wagering data, and any other sports-related information to user equipment 102. In some embodiments, this sports-related information is provided to user equipment 102 as a plurality of real-time data feeds" (Ex.1016, ¶41), but does not detail the use of message queue servers or topic-based data management for minimizing latency.

26. By integrating Spivey's architecture, which includes "a message queue server...capable of receiving, processing, and sending information for the system" and "push[ing] the data payload" directly to client devices, a POSA would have recognized that the system would achieve faster, more efficient, and lower-latency delivery of live event data compared to the more traditional, less optimized distribution mechanisms described in Ellis. Ex.1017, ¶46.

27. PANDA further contends that Spivey's push provisions are not compatible with Ellis but does not provide any reasons to support this assertion. POR, 46-47. As I explained in my first declaration, Ellis's video delivery and Spivey's push provisions are compatible because both disclose compatible hardware and software components that act as sources for event information to be transmitted to users. Ex.1001 ¶114. To the extent PANDA argues that event data must be synchronized with the video feed, neither Ellis nor the challenged claims require synchronization. Ex.1016 ¶41 (providing a plurality of real-time data feeds); *see*

e.g., Ex.1004, Cl. 1 (requiring a video feed and related event information, not that the two are synchronized to one another).

28. Ellis describes that "supplemental content source 120 may be any suitable content source configured to provide real-time sports scores and game summary information, tournament brackets and tournament information, sports commentary, athlete biographies, spread and wagering data, and any other sports-related information to user equipment 102," and further details that "the wagering interface may support both parimutuel and fixed odds betting" and allows users to "build wagers and place these wagers with a network transaction processing system." Ex.1016, ¶¶ 41 and 116. By delivering odds and related event data in real time—potentially even before the corresponding video stream reaches the user—the system ensures that the betting experience is synchronized with the actual state of the event, reducing the risk of stale or outdated information influencing wagers and enhancing user engagement and trust in the platform's responsiveness.

29. Additionally, Dr. Rubin asserted that adding Spivey's operating system 400 would leave "the system without a source of supplemental content." Ex.2001 ¶212. This is incorrect, because the supplemental content (event information) is received by the LDSD at the LDS Communication Component. Ex.1001 ¶¶109, 346-47. Furthermore, Ellis does not require that the Supplemental Content Source

generates the content, it only needs to function as an "intermediate content provider" that "is configured to provide real-time sports scores." Ex.1016 ¶¶41-42.

IV. A POSA WOULD HAVE BEEN MOTIVATED TO COMBINE HERZOG-SPIVEY-ELLIS

30. PANDA's primary issue with my obviousness rationale appears to stem from a misunderstanding. PANDA argues that Herzog requires latency to function, and thus a POSA would not have been motivated to reduce Herzog's data-delivery latency. POR, 7, 53, 58. I agree, but only to the extent that the video data must have latency. A POSA would have known that reducing the data-distribution latency would have improved the system's capabilities and thus a POSA would have been motivated to combine Herzog-Spivey-Ellis.

31. The latency period between the video feed and the data (meta-data) feed in Herzog is an intended feature, not a bug, because the system is architected to synchronize disparate data streams that are "published and processed at different rates" and often originate from "different physical locations" and transmission paths. As described, "the latency or transaction time of the meta-data between the client and the server is less than 2 seconds. The latency of the video between when the user requests a video and when the video is shown on the display is approximately 63 seconds. This is due to the down linking, encoding, publishing and decoding of the video." Ex. 1015 ¶¶ 58, 65, and 74. Herzog's solution is to time-stamp both streams using "consistent (e.g., GPS) timing protocols," so that "each respective data is time-

tagged to the same timing signature, even though each respective data transmits from different physical locations." *Id.,* ¶18. The system then "time synchroniz[es] the second type of data to the first type of data; and display[s] the first and second types of data on the mobile station." *Id. , Cl.*. This design allows the meta-data (slave) to be "synchronized to the video/audio data (the master)" (*id*., ¶58), ensuring that, despite their inherent latency differences, both streams are "assembled in the mobile device... for appropriate presentation on the device's display." *Id.* Thus, the latency is a deliberate aspect of the architecture, enabling "a viewer [to] comprehend the multiple information displayed on a mobile device, or a tablet" (*id.,* ¶4), which is the core objective of Herzog's disclosure.

32. Reducing the data delivery period while maintaining the video latency period allows the system additional time to process multiple user requests and alleviates demand on its components. Herzog utilizes a "pull-based" architecture, where the client application requests meta-data corresponding to the time-stamped video stream, resulting in increased server load due to the volume of individual requests: "the client application... is configured to request video and audio streaming data... [and] then... request the corresponding meta-data which has the same time code reference." *Id.,* ¶¶57 and 58. This approach contrasts with Spivey's "push-based" architecture, in which data is delivered to clients immediately as it becomes

available in the queue, thereby reducing the frequency and volume of client-initiated requests to the server.

33. PANDA also contends that the "unfettered provision of Spivey data payloads" to mobile devices would render Herzog inoperable. POR, 58-59. This is incorrect. Providing event information as taught by Spivey would not have rendered Herzog inoperable, because Spivey is simply transmitting data from one server to a client device—something Herzog (abstract), Ellis (¶46), and Spivey (*e.g.,* Fig. 22) were all designed to do.

34. In Herzog, the client application on the mobile device only makes additional requests when the user changes the video because the system designates the time stamp of the selected video stream as the master clock for synchronization, and the meta-data is then requested to match this specific time. Once a video is chosen, the client application requests the corresponding meta-data with the same time code, ensuring both are displayed in sync. As long as the user does not change the video, the meta-data remains synchronized, eliminating the need for further requests. However, when the user selects a different video, a new request is necessary to obtain meta-data that matches the new video's time stamp, thus maintaining synchronization. As described in Herzog: "When the viewer makes a change on the video player, the handshaking interchange or flow shown in FIG. 6 is

repeated anew" (Ex 1015, ¶79), indicating that additional requests are triggered only by a change in the selected video stream.

35. PANDA makes another strawman argument that Spivey's websockets are incapable of streaming audio/video. That is incorrect. Spivey's websockets are capable of streaming audio and video because the websocket protocol provides a full-duplex, low-latency, and persistent communication channel between a server and client devices. This architecture enables the efficient, real-time transmission of data packets—including audio and video streams—without the overhead and delays associated with traditional request-response protocols like HTTP polling. As described in Spivey, the use of websockets allows data to be pushed from the server to subscribed clients with delays reduced to the order of milliseconds, which is essential for live audio/video streaming applications that require minimal latency and continuous data flow. The protocol's bi-directional nature further supports interactive and synchronized media experiences, making it well-suited for real-time streaming scenarios.

V. CONCLUSION

36. I hereby declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct, and that all statements made of my own knowledge are true and that all statements made on information and belief

are believed to be true. I understand that willful false statements are punishable by fine or imprisonment or both. *See* 18 U.S.C. §1001.

Date: August 22, 2025

Respectfully submitted,

Dr. Stephen B. Wicker