

U.S. PATENT NO. 11,039,218 V/S SPORTRADAR EmBET

I. INTRODUCTION

The chart below demonstrates how Defendant Sportradar Group, AG (“Defendant” or “Sportradar”) as well as Defendant’s partners, customers, and end users of the products and features discussed below, directly infringe, either literally or under the doctrine of equivalents, at least claims 1 and 4 of U.S. Patent No. 11,039,218 (“the ’218 Patent”). Defendant and Defendant’s partners, customers, and end users directly infringe the identified claims at least by making, using, selling, installing, configuring to operate, importing, testing, and/or offering to sell at least Defendant’s emBET Platform and other offerings and services that integrate with Defendant’s emBET Platform (collectively, “the Accused Products”) either alone, and/or when installed and operated as designed and intended on a computer system or network, as detailed below.

Each element of each asserted claim is presently alleged to be literally present. Because discovery has not started and the Asserted Claims have not yet been construed, SportsCastr Inc. (d/b/a PANDA Interactive) (“Plaintiff” or “PANDA”) reserves the right to supplement its contentions, including with respect to the Doctrine of Equivalents as appropriate. In addition, any of Defendant’s products that perform similar functions and/or incorporate or rely on the same or similar functionality (for example, without limitation, any predecessor or successor products performing the same functions in the same ways or products which use the same software, plugins, or libraries or call the functions underlying the operations identified below) as the Accused Products also infringe in the manner set out below.

Defendant directly infringes at least claims 1 and 4 of the ’218 Patent, either literally or under the doctrine of equivalents, by making, using, offering to sell and selling the systems described below. Defendant’s partners, customers, and end users of its Accused Products and corresponding systems and services directly infringe at least claims 1 and 4 of the ’218 Patent, literally or under the doctrine of equivalents, at least by using the Accused Products and corresponding systems and services, as described below.

Defendant has actively induced and are actively inducing infringement of at least claims 1 and 4 of the ’218 Patent with specific intent to induce infringement, and/or willful blindness to the possibility that its acts induce infringement, in violation of 35 U.S.C. § 271(b). For example, Defendants encourages and induces customers to use the Genius Sports Platform in a manner that infringes claims 1 and 4 of the ’218 Patent at least by offering and providing the software and by engaging in activities relating to selling, marketing, advertising, promotion, installation, support, and distribution of the Accused Products.

Defendant encourages, instructs, directs, and/or requires third parties—including its certified partners and/or customers—to use the Accused Products in infringing ways. Defendants further encourages and induces its customers to infringe claims 1 and 4 of the

'218 Patent: 1) by widely advertising those services, and providing technical support and instructions to users, and 2) through activities relating to marketing, advertising, promotion, installation, support, and distribution of the Accused Products, including its video streaming software, and services in the United States. (See PANDA-SR_000041-PANDA-SR_00044, <https://sportradar.com/sports-entertainment/audiovisual/ott/?lang=en-us>.) Further, as the entity that provides installation, implementation, and integration of the Accused Products in addition to ensuring the Accused Product remains operational for each customer through ongoing technical support, on information and belief, Defendant and/or its partners affirmatively aid and abet each customer's use of the Accused Products in a manner that infringes the '218 Patent.

Defendant also contributes to the infringement of its partners, customers, and end-users of the Accused Products by providing within the United States or importing into the United States the Accused Products, constituting a material part of the inventions claimed, and not a staple article or commodity of commerce suitable for substantial non-infringing uses. Indeed, as shown below, the Accused Products have no substantial non-infringing uses because the accused functionality, including the one-to-many socket architecture and separate channels for video and event information, is an integral part of the Accused Products and must be performed for the Accused Products to perform their intended purpose. These functionalities are continually running when the system is in use and, on information and belief, cannot be removed or disabled (or, if they could, the system would no longer suitably function for its intended purpose). Moreover, for the same reasons, without the system and components identified below that practice the '218 Patent, that functionality could not be performed. Additionally, the accused functionality, including the one-to-many socket architecture and separate channels for video and event information, itself has no substantial non-infringing uses because the components, modules and methods identified below are a necessary part of that functionality. For example, without the on-to-many socket architecture, the Accused Products could not effectively synchronize the event information across multiple viewers. These functionalities are continually running when the system is in use and cannot be removed or disabled (or, if they could, the system would no longer suitably function for its intended purpose). Moreover, for the same reasons, without the system and components identified below that practice claimed in the '218 Patent, that functionality could not be performed.

Indeed, as shown below, the Accused Products have no substantial non-infringing uses because the accused functionality is an integral part of the Accused Products and must be performed for the Accused Products to perform their intended purpose. For example, at least the functionalities and corresponding modules (whether software and/or hardware) described below have no suitable non-infringing use and form a material part of the Accused Products. In other words, when each of the Accused Products operates, the functions below are performed. Moreover, when the functionality and modules specifically described below are operated, they perform the steps described below. The Accused Products (and the specific modules and functionalities described herein) are designed and intended specifically to perform the inventions claimed in the Asserted Claims, and do not have any other purpose. For example, the Accused Products and accused functionalities (e.g., the one-to-many architecture) constitute a material part of the inventions claimed

because such architecture is integral to the system as recited in the claims of the '218 Patent. None of these products are staple goods—they are sophisticated and customized video streaming products, methods, and systems.

Plaintiff provides the following example claim charts based on currently available information, including through public sources. Plaintiff reserves the right to modify, amend, or supplement these claim charts should it become aware of additional information regarding the Accused Products and systems, including through discovery, or should one or more claims be construed in a manner differently than interpreted herein.

II. CLAIM CHARTS

1. Independent Claim 1

Key Features	Accused Product
[1P] A system for controlling a plurality of viewer client devices to receive first digital content relating to a first sporting event and first event information germane to the first sporting event, the first event information including online gaming information, the system comprising:	Sportradar emBET includes “<i>a system for controlling a plurality of viewer client devices to receive first digital content relating to a first sporting event and first event information germane to the first sporting event, the first event information including online gaming information.</i>” Sportradar emBET, which is offered under their brand name Betradar, is an Over-the-Top (OTT) product that provides an interactive overlay for video streams (from either Sportradar’s event video stream portfolio or an external video source). The overlay includes a display of game-related data and an interface for placing bets. The emBET system is implemented using an architecture that includes a control server (see limitation 1A) that feeds a socket server (see limitation 1B) information relevant to a sporting event, which the socket server then distributes to viewer client devices to cause them to update their display of information. The system also includes a webserver (see limitation 1C) which connects the client devices to digital content relating to a first sporting event, such as live streams of sporting events. Two deployment options are available for emBet, “SDK” and “Platform.” The SDK option integrates with an emBET purchaser’s front-end system, such that the emBET purchaser can configure how video and data feeds will be displayed on the client devices of end-user bettors. The “Platform” option “plugs into existing Sportradar feeds” (both video feeds and data feeds), and then Sportradar’s partner,

Key Features	Accused Product
	LiveLike “synchronises the video and data feeds, which convert into graphical overlays.” Thus, both options operate to combine and synchronize live sports data with live sports video, as claimed. emBET comes in two components: SDK An SDK that integrates with your platform’s front-end, displaying all the graphical overlays directly onto your video player. Platform A back-end (administrative) platform that plugs into existing Sportradar feeds – where all overlays are configured. Our partner, LiveLike, then synchronises the video and data feeds, which convert into graphical overlays. See, PANDA-SR_000063 at PANDA-SR_000064, https://sportradar.com/sports-entertainment/audiovisual/embet/?lang=en-us. As described below, emBET provides various types of information to users, such as “live odds, insights, and stats overlayed on live [sporting event] streams”:

Key Features	Accused Product
	What is emBET? It's a piece of software (SDK) that lets rightsholders and media companies show engaging live odds, insights and stats overlayed on live streams. It's an entertaining interactive tool that keeps viewers spending and coming back. If your odds feature in these emBET overlays you'll attract new customers to your sportsbook. The kind of customers who might have never considered placing a live bet otherwise.  <i>See: See, PANDA-SR_000029 at PANDA-SR_000034, Audio-Visual Brochure.</i> Further, emBET provides event information, such as match insights, match odds, and live stats, related to a sporting event, such as a basketball game, to emBET subscribers or clients. The event information and live stream of the sporting event are transmitted over Internet and displayed on client devices.

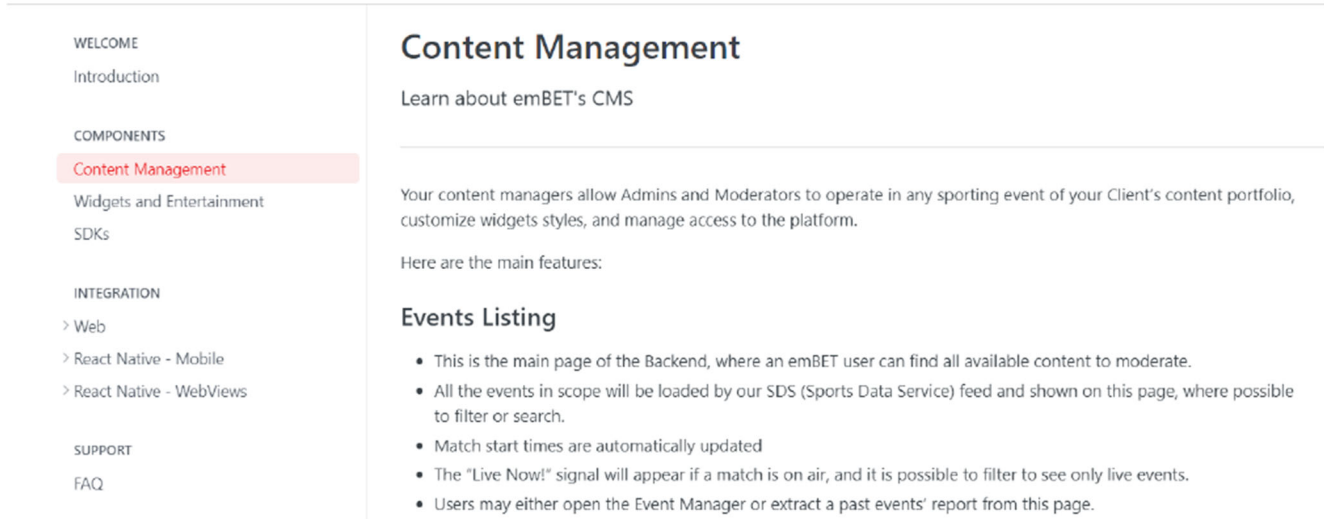
Key Features	Accused Product
	WELCOME Introduction COMPONENTS Content Management Widgets and Entertainment SDKs INTEGRATION > Web > React Native - Mobile Welcome to emBET! The following document will guide you through the integration and usage of the new Sportradar engagement product. emBET adds another level of entertainment to any streamed sports event and is built in partnership with one of the most expert teams in overlayed graphical messages: LiveLike It is built in two components: The backend, or CMS, and the SDK, which allows viewers to receive the widgets while watching the stream and interact with them. One of emBET's main goals is to offer users the widest possible range of flexibility and customization. Hence, when integrating the SDK the integrator should expect to input all of his preferences in terms of widget positioning, size and behaviour (E.g. On Mobile the Landscape vs Portrait scenario). The diagram illustrates the architecture of the emBET system. At the top, a red box labeled 'emBet Backend' contains three icons: a circuit board for 'Match Insights', a '1x2' odds board for 'Live Odds', and a bar chart for 'Live Stats'. A red arrow points from this box to a blue box labeled 'emBet SDK'. Below the SDK box, three sample widget overlays are shown: 'LIVE MONEYLINE' for a basketball game (BOSTON vs CHICAGO), 'LIVE SPREAD' for a basketball game (BOSTON vs CHICAGO), and 'LIVE MONEYLINE' for a football game (MANCHESTER vs LIVERPOOL). Each widget includes a 'PLACE BET' button. A photograph of a laptop on a desk. The laptop screen shows a live basketball game. Overlaid on the right side of the screen are several emBET widgets, including a 'LIVE MONEYLINE' widget showing odds for Boston and Chicago, and a 'LIVE SPREAD' widget showing odds for Boston and Chicago. A 'PLACE BET' button is visible on the bottom right of the screen. Customer Platform

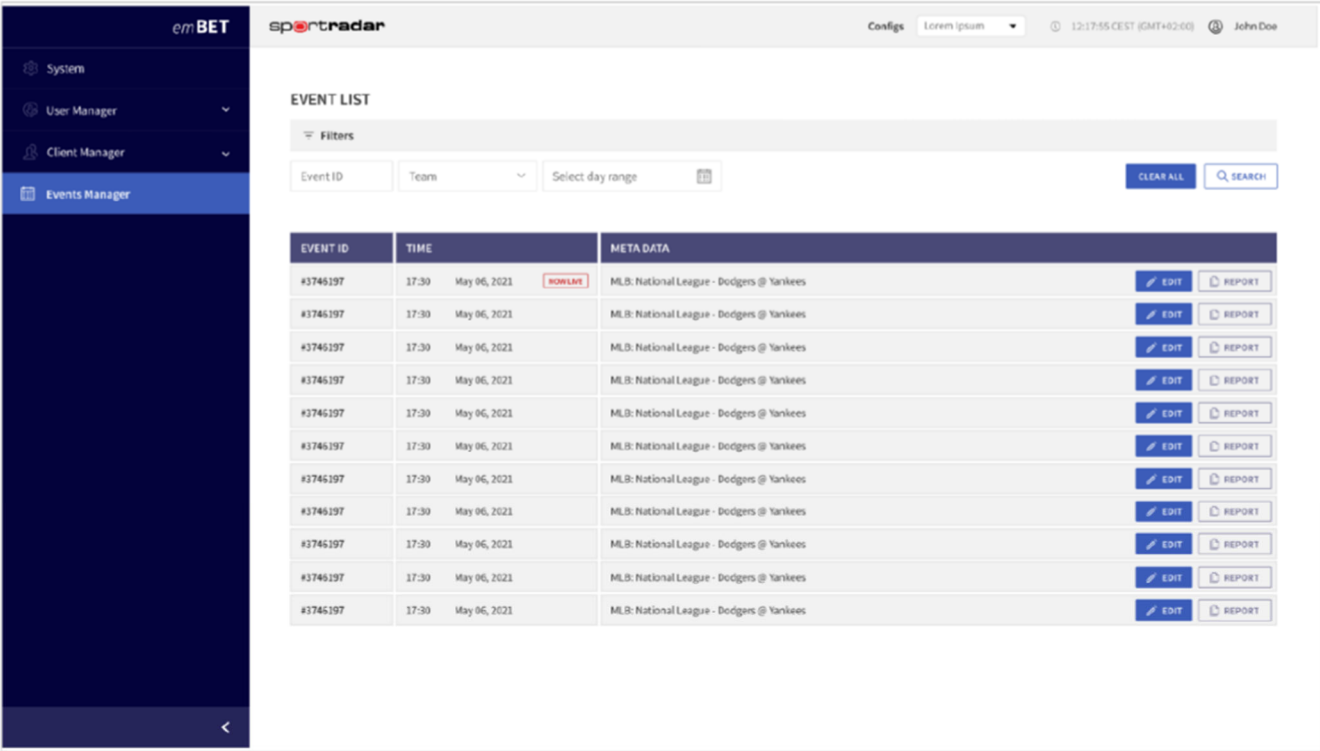
Key Features	Accused Product
	 <i>See, https://embet.readme.io/docs.</i> As shown below, the information provided by emBET includes online gaming information, such as live odds, live stats, insights, and trivia and predictions:

Key Features	Accused Product
	 The screenshot displays the Sportradar website interface, which is divided into several sections for live betting and entertainment. The top section is titled 'LIVE ODDS' with the tagline 'The bread and butter.' It features three 'LIVE MONEYLINE' widgets for different sports events, each showing odds for 'HOME' and 'AWAY' teams. Below these are three buttons: 'Cast you vote bet', 'See the results', and 'Make your bet'. To the right, the 'LIVE STATS' section is titled 'Live stats can alert your viewers to exciting betting opportunities.' It shows a 'BAL vs NYM' game with various statistics and a 'WHO WILL WIN?' widget. The bottom left section is titled 'INSIGHTS' with the tagline 'Live match insights matched with Live Odds to keep fans involved.' It features a 'LIVE INSIGHT' widget about Tyler Matzek's performance. The bottom right section is titled 'TRIVIA & PREDICTIONS' with the tagline 'Guess trivia questions and predict results to gain points, climb the leaderboard, and get rewards!'. It includes a 'LIVE PREDICTION' widget about Adam Smith's performance and a 'TRIVIA' widget about the Yankees' home record. The Sportradar logo is visible at the bottom of the page. See, PANDA-SR_000102, https://embet.readme.io/docs/widgets-and-entertainment.
A) a control server to periodically retrieve, via the Internet, the first event information germane to the first sporting event;	The embET includes “a control server to periodically retrieve, via the Internet, the first event information germane to the first sporting event.” For example, Sportradar’s APIs demonstrates a socket server periodically receiving information germane to a sporting event from a control server. The control server used by the embET platform retrieves event information feeds from the Internet and in turn sends instructions to socket servers (discussed immediately below); the socket servers in turn distribute the instructions from the control server to end users which results in information being displayed on client devices. The embET software sets up “listeners,” such as “mainListener” below, which are blocks of code implemented at the socket server that cause information received from the control server to be

Key Features	Accused Product
	distributed to and control end user device displays. <pre data-bbox="569 337 1812 544">const mainListener = { message: (msg) => { msg.message.event === 'program-status-updated' && eventDispatcher(document, msg.message.event, msg.message.payload); msg.timetoken && (msg.message.payload.timetoken = msg.timetoken); } };</pre> Tracing the code through the “listeners” and down through the event dispatch system reveals that the information is provided to live display items (e.g., scores or online gaming information), which are then added and/or modified in response to the controlling instructions received.

Key Features	Accused Product
	<pre> let LiveTotal = class LiveTotal extends EmbetWidget {}; LiveTotal = __decorate\$1([element\$1('live-total')], LiveTotal); let LiveSpread = class LiveSpread extends EmbetWidget {}; LiveSpread = __decorate\$1([element\$1('live-spread')], LiveSpread); let LiveMoneyline = class LiveMoneyline extends EmbetWidget {}; LiveMoneyline = __decorate\$1([element\$1('live-moneyline')], LiveMoneyline); let LiveAtBat = class LiveAtBat extends EmbetWidget { constructor() { super(...arguments); this.insightsRenderer = insightsRenderer('table'); } }; LiveAtBat = __decorate\$1([element\$1('live-at-bat')], LiveAtBat); let LiveStatsOdds = class LiveStatsOdds extends EmbetWidget { constructor() { super(...arguments); this.insightsRenderer = insightsRenderer('stats'); } }; LiveStatsOdds = __decorate\$1([element\$1('live-stats-odds')], LiveStatsOdds); </pre>
B) at least one socket server communicatively coupled to the control server to:	The emBET includes “at least one socket server communicatively coupled to the control server.” As shown in limitation 1A, in emBET the listeners implemented show that information is received by the socket server from the control server.

Key Features	Accused Product
[B.1] receive from the control server at least the first event information; and	As shown in the prior limitations (1A and B), the socket server in emBET “ <i>receive[s] from the control server at least the first event information.</i> ” For example, the control server periodically retrieves first event information germane to the first sporting event and sends the first event information to the socket server for provisioning to the widgets.
[B.2] transmit at least some of the first event information, including the online gaming information, to at least a first viewer client device of the plurality of viewer client devices via a first event information Internet communication channel between a first event socket of the at least one socket server and the first viewer client device, wherein the first event socket corresponds to the first event information germane to the first sporting event; and	The emBET includes a socket server to “<i>transmit at least some of the first event information, including the online gaming information, to at least a first viewer client device of the plurality of viewer client devices via a first event information Internet communication channel between a first event socket of the at least one socket server and the first viewer client device, wherein the first event socket corresponds to the first event information germane to the first sporting event.</i>” The emBET Content Management tool allows an emBET purchaser/user (e.g., a sportsbook operator and/or media company) to control and arrange how their end-user bettors will see the video display and the graphic overlays of game-related data, including betting data:  See, PANDA-SR_000038, https://embet.readme.io/docs/content-management#events-listing.

Key Features	Accused Product
	 See, PANDA-SR_000038, https://embet.readme.io/docs/content-management#events-listing. For a given event on the EVENT LIST, emBET's Content Manager also includes an Event Manager:

Key Features	Accused Product
	WELCOME Introduction COMPONENTS Content Management Widgets and Entertainment SDKs INTEGRATION > Web > React Native - Mobile > React Native - WebViews SUPPORT Event Manager • Event Manager contains a Media Library which allows you to see all widgets at a glance, or filter them by category (E.g. Betting, Statistics) and type (Moneyline, Over/Under). • Selecting a widget will project it onto the preview space with the most up-to-date odds. • Preview Player mode allows you to watch the on-air event to make sure the look and timing of the widgets are just right. It is possible to place the overlays on any corner of the stream. • If an event stream is not in Sportradar's portfolio, you have the option to manually insert the URL for ingestion • You may set the duration of a widget's appearance on the viewers' player to 5, 10, 15, or 25 seconds. • Analytics are available, including a timeline record of the widgets and their impact on the audience. You can even extract complete reports after the event has ended (See "Analytics") Visual Assets • When allowed by local regulation, widgets will display the official teams' logo • Otherwise, the widgets will display teams jerseys and colours See, PANDA-SR_000038-PANDA-SR_000039, https://embet.readme.io/docs/content-management#event-manager. As illustrated in the screenshot below, emBet transmits online gaming information, which further includes event ID, event name, event history markets (including 1st half, 1st half-total, 1st half-total corners, etc.), odds, etc.

Key Features

Accused Product

The screenshot displays the emBET Sportradar interface. The top navigation bar includes the emBET and Sportradar logos, a client dropdown set to 'ATCMP', a timestamp '14.08.22 (GMT+01:00)', and a user profile 'Hello, g.ferrante@sportradar.com'. The left sidebar contains a menu with 'Events' and 'Theme Styling'. The main content area is titled 'Events' and shows details for a match: 'ID: 31632571 Event: NK DOMZALE - SK RUSTRIAKLAGENFURT Date: 2022-01-26 15:30'. Below this, the 'Markets' section lists various betting options like '1st half - 1x2', '1st half - total', and '1st half - total corners', each with a 'Show Odds' button. A 'History' table is also present, showing a list of bets with columns for 'MARKET', 'TYPE', 'TIME', 'VIEWERS', and 'INTERACTIONS'. On the right, a 'Widget Preview' shows a 'MONEYLINE' widget for 'WHO WILL WIN TONIGHT?' with options for 'TERMINES +120', 'DRAW +220', and 'BACKGAMES +240'. Below the widget is a 'Live Stream' section with a video player showing a football match and a 'Trigger' button.

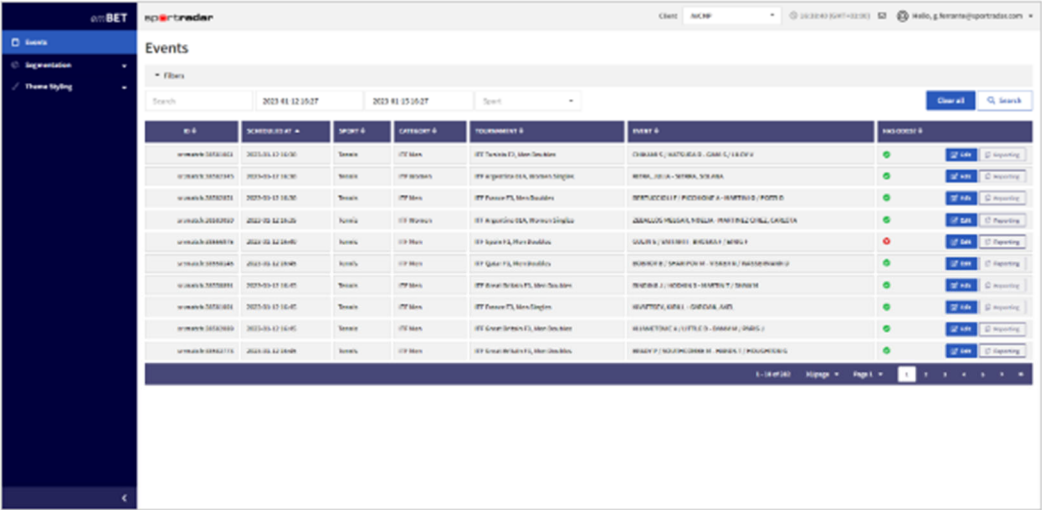
MARKET	TYPE	TIME	VIEWERS	INTERACTIONS
1x2	LIVE MONEYLINE	2022-01-26 13:11:57	3	1 (33.33%)
1x2	LIVE MONEYLINE	2022-01-26 13:11:26	1	2 (66.67%)
1x2	LIVE MONEYLINE	2022-01-26 13:00:35	1	3 (100%)
1x2	LIVE MONEYLINE	2022-01-26 13:05:26	1	3 (100%)
1x2	LIVE MONEYLINE	2022-01-26 13:03:34	1	3 (100%)

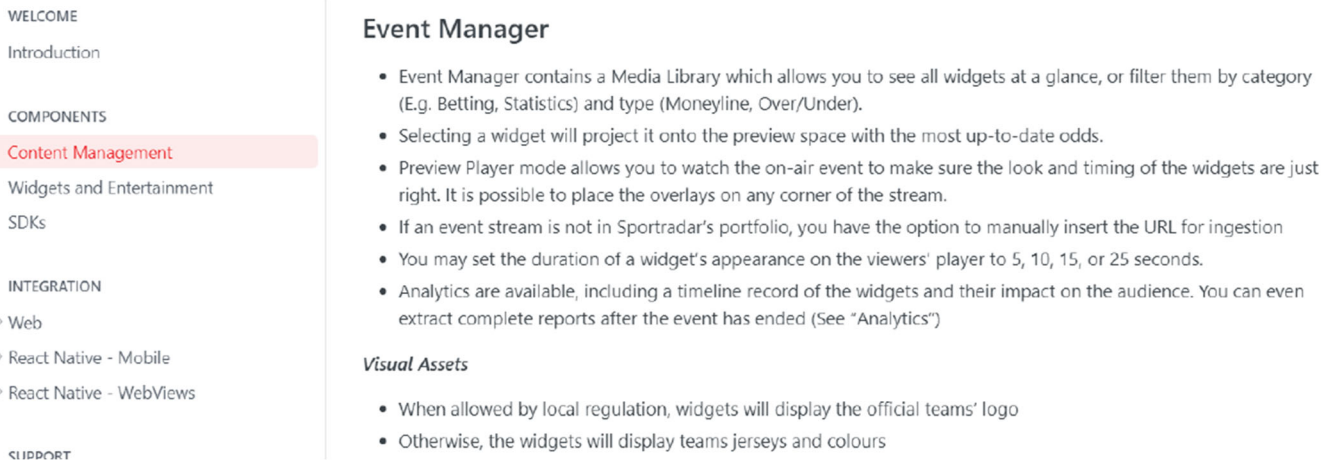
The source code for the emBET SDK, which is publicly available, confirm the use of socket and socket server.

Key Features	Accused Product																											
	@livelike/embet 0.5.1 • Public • Published 2 months ago Readme CodeBeta 0 Dependencies 0 Dependents 43 Versions / @livelike/embet / <table><tr><td>CHANGELOG.md</td><td>text/markdown</td><td>7.64 kB</td></tr><tr><td>README.md</td><td>text/markdown</td><td>76 B</td></tr><tr><td>embet.cjs.js</td><td>application/javascript</td><td>8.43 MB</td></tr><tr><td>embet.cjs.min.js</td><td>application/javascript</td><td>1.19 MB</td></tr><tr><td>embet.esm.js</td><td>application/javascript</td><td>8.43 MB</td></tr><tr><td>embet.esm.min.js</td><td>application/javascript</td><td>1.19 MB</td></tr><tr><td>embet.umd.js</td><td>application/javascript</td><td>8.49 MB</td></tr><tr><td>embet.umd.min.js</td><td>application/javascript</td><td>1.19 MB</td></tr><tr><td>package.json</td><td>application/json</td><td>196 B</td></tr></table> Install > npm i @livelike/embet ± Weekly Downloads 98 Version License 0.5.1 none Unpacked Size Total Files 28.9 MB 9 Last publish 2 months ago Collaborators See, PANDA-SR_00004-PANDA-SR_000005, https://www.npmjs.com/package/@livelike/embet?activeTab=code. LiveLike is implemented on the embBET SDK (in collaboration with or under the direction/control of Sportradar). LiveLike uses PubNub to ensure effective operation of their SDKs and APIs:	CHANGELOG.md	text/markdown	7.64 kB	README.md	text/markdown	76 B	embet.cjs.js	application/javascript	8.43 MB	embet.cjs.min.js	application/javascript	1.19 MB	embet.esm.js	application/javascript	8.43 MB	embet.esm.min.js	application/javascript	1.19 MB	embet.umd.js	application/javascript	8.49 MB	embet.umd.min.js	application/javascript	1.19 MB	package.json	application/json	196 B
CHANGELOG.md	text/markdown	7.64 kB																										
README.md	text/markdown	76 B																										
embet.cjs.js	application/javascript	8.43 MB																										
embet.cjs.min.js	application/javascript	1.19 MB																										
embet.esm.js	application/javascript	8.43 MB																										
embet.esm.min.js	application/javascript	1.19 MB																										
embet.umd.js	application/javascript	8.49 MB																										
embet.umd.min.js	application/javascript	1.19 MB																										
package.json	application/json	196 B																										

Key Features	Accused Product
	PERSONALIZATION CAPABILITIES Informational widgets provide the right personalized data at the right time, keeping your users in your app rather than having them switch out. These widgets offer you sponsorship opportunities.  See, PANDA-SR_000053 at PANDA-SR_000055, https://www.pubnub.com/customers/liveline/. PubNub is a socket communication service that allows developers “to open and maintain a socket connection for bidirectional network communication.” Socket.IO is a framework with abstracted concepts that makes network communication more robust, <u>allowing developers to open and maintain a socket connection for bidirectional network communication</u>. It has some great real-time features and uses patterns to make it easy to use. One way to think about it: Socket.IO is to networking as jQuery is to HTML/JavaScript.

Key Features	Accused Product
	See, PANDA-SR_000046 at PANDA-SR_000047, https://www.pubnub.com/blog/pubnub-takes-socketio-next-level/. The Sportradar emBET source code, as shown in limitation 1A above, shows the socket server receives event information from a control server and updates the overlay information. The emBET source code below further shows that clients are directed to connect to an event socket of the socket server (e.g., the channel) to receive the event information: <pre> const subscribe = (channel) => pubnub === null pubnub === void 0 ? void 0 : pubnub.subscribe({ channels: [channel] }); const safeSubscribe = (channel) => (pubnub === null pubnub === void 0 ? void 0 : pubnub.getSubscribedChannels) && !(pubnub === null pubnub === void 0 ? void 0 : pubnub.getSubscribedChannels().some((c) => c === channel)) && subscribe(channel); const unsubscribe = (channel) => pubnub === null pubnub === void 0 ? void 0 : pubnub.unsubscribe({ channels: [channel] }); const addListener = (listener) => pubnub === null pubnub === void 0 ? void 0 : pubnub.addListener(listener); const removeListener = (listener) => pubnub === null pubnub === void 0 ? void 0 : pubnub.removeListener(listener); const getMessages = (channel, opts = params({ count: 25, end: undefined, start: undefined })) => new Promise((res, rej) => { pubnub === null pubnub === void 0 ? void 0 : pubnub.fetchMessages({ channels: [channel], count: opts.count, stringifiedTimeToken: true, start: opts.start, end: opts.end, includeMessageActions: true, }, (status, response) => (status.error ? rej(status) : res(response))); }); </pre>
C) at least one webserver communicatively coupled to the at least one socket server to transmit, to the first viewer client device:	EmBET further includes “at least one webserver communicatively coupled to the at least one socket server to transmit, to the first viewer client device.” For example, the emBET Events listing page demonstrates the use of at least one webserver. The emBET Events listing page is “the main page of [emBET’s] Backend.” The events listing is a webpage, where “[a]ll the events in scope will be loaded by [Sportradar’s] SDS (Sports Data Service) feed and shown on this page....” Using the links on this page, “[u]sers may either open the Event Manager or extract a past events’ report from this page.”

Key Features	Accused Product
	Events Listing • This is the main page of the Backend, where an emBET user can find all available content to moderate. • All the events in scope will be loaded by our SDS (Sports Data Service) feed and shown on this page, where possible to filter or search. • Match start times are automatically updated • The “Live Now!” signal will appear if a match is on air, and it is possible to filter to see only live events. • Users may either open the Event Manager or extract a past events’ report from this page.  See, PANDA-SR_000038-PANDA-SR_000039, https://embet.readme.io/docs/content-management#events-listing. Furthermore, emBET’s use of widgets demonstrates the use of “<i>at least one webserver [is] communicatively coupled to the at least one socket server.</i>” See discussion of widgets in Claim Elements [1A] & [1B], <i>supra</i>.

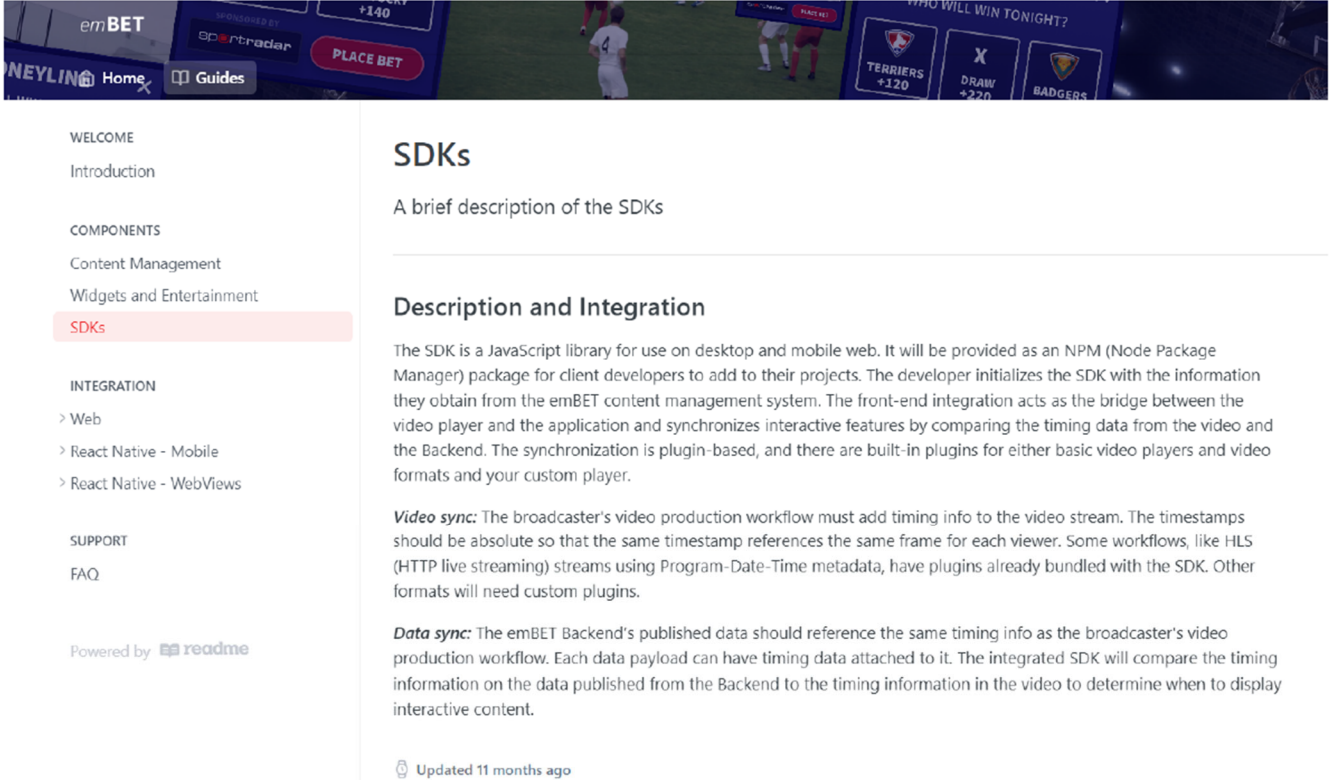
Key Features	Accused Product
	For example, as explained in the screenshot below, “[s]electing a widget will project it onto the preview space with the most up-to-date odds.”  The screenshot shows a sidebar on the left with sections: WELCOME, Introduction, COMPONENTS, Content Management (highlighted), Widgets and Entertainment, SDKs, INTEGRATION, and CLIPPORT. The main content area is titled 'Event Manager' and contains a bulleted list of features: Event Manager contains a Media Library which allows you to see all widgets at a glance, or filter them by category (E.g. Betting, Statistics) and type (Moneyline, Over/Under); Selecting a widget will project it onto the preview space with the most up-to-date odds; Preview Player mode allows you to watch the on-air event to make sure the look and timing of the widgets are just right. It is possible to place the overlays on any corner of the stream; If an event stream is not in Sportradar's portfolio, you have the option to manually insert the URL for ingestion; You may set the duration of a widget's appearance on the viewers' player to 5, 10, 15, or 25 seconds; Analytics are available, including a timeline record of the widgets and their impact on the audience. You can even extract complete reports after the event has ended (See "Analytics"). Below this is a section titled 'Visual Assets' with a bulleted list: When allowed by local regulation, widgets will display the official teams' logo; Otherwise, the widgets will display teams jerseys and colours. See, PANDA-SR_000038-PANDA-SR_000039, https://embet.readme.io/docs/content-management#event-manager.
[C.1] a first Internet address of a first media source to establish a first video Internet communication channel between the first media source and the first viewer client device to carry the first digital content relating to the first sporting event; and	The emBET webserver transmits “<i>a first Internet address of a first media source to establish a first video Internet communication channel between the first media source and the first viewer client device to carry the first digital content relating to the first sporting event.</i>” For example, as part of the content configuration process, via operation of the emBET SDK, Sportradar allows the emBET purchaser to request a video feed, at least for those sporting events that are in Sportradar’s portfolio. As explained below, emBET’s “Event Manager contains a Media Library which allows [customers] to see all widgets at a glance....” Further, “[i]f an event stream is not in Sportradar’s portfolio, [the customer has] the option to manually insert the URL for ingestion.”

Key Features	Accused Product
	WELCOME Introduction COMPONENTS Content Management Widgets and Entertainment SDKs INTEGRATION > Web > React Native - Mobile > React Native - WebViews SUPPORT Event Manager • Event Manager contains a Media Library which allows you to see all widgets at a glance, or filter them by category (E.g. Betting, Statistics) and type (Moneyline, Over/Under). • Selecting a widget will project it onto the preview space with the most up-to-date odds. • Preview Player mode allows you to watch the on-air event to make sure the look and timing of the widgets are just right. It is possible to place the overlays on any corner of the stream. • If an event stream is not in Sportradar's portfolio, you have the option to manually insert the URL for ingestion • You may set the duration of a widget's appearance on the viewers' player to 5, 10, 15, or 25 seconds. • Analytics are available, including a timeline record of the widgets and their impact on the audience. You can even extract complete reports after the event has ended (See "Analytics") Visual Assets • When allowed by local regulation, widgets will display the official teams' logo • Otherwise, the widgets will display teams jerseys and colours See, PANDA-SR_000038-PANDA-SR_000039, https://embet.readme.io/docs/content-management#event-manager. Sportradar maintains an expansive portfolio of live sporting event video feeds. As described in the Betradar Audio-Visual Brochure in which emBET is introduced, the companion "Live Channel" offerings provide the video engine that powers emBET:

Key Features	Accused Product
	Our streaming products Get more from streaming Our award-winning Live Channel product family gives you the largest, most diverse live video streaming portfolio in betting. As well as accurate information, betting engagement and match visualisation. We provide video streams for over 280,000 live sporting events a year, including all Bundesliga matches, the Australian Open, ATP and ITF Tennis tournaments, NBA, MLB, NHL and many of the top events and leagues from 17 different sports. What you'll get → Live Channel Online and Live Channel Retail put streams directly into your sportsbook or shops. So your customers get an immediate incentive to bet more, stay longer and come back again and again. And you get to make the most of the number one revenue generator in the industry: in-play betting. → Go for Live Channel Trading to give your traders an additional fast and accurate source of information during live events. → Choose Live Channel Promotion to use near-live highlight clips from your video streams as a promotional tool to ultimately increase mobile betting revenues. → And use our latest product, emBET, to connect your live betting offer with video streams outside of your sportsbook. See, PANDA-SR_000006 at PANDA-SR_000020, Product Brochure 2022.

Key Features	Accused Product
	Live Channel Online Find out more Live streams to boost your in-play betting offer Customers bet more, spend more time and visit more when they can watch the action live within your in-play betting offering. With Live Channel Online (LCO) you get hundreds of individual live event streams every day, many different sports, leagues, and competitions. So you can offer them simultaneously throughout the day, boosting the user experience on your event pages. What you'll get → 24/7 live sports streaming on your website and app. → Easy integration: you only need to do it once with our Audiovisual API to get our entire video streaming portfolio. → Broad streaming format support to reach a wide range of devices with low-latency streams (CMAF, MPEG-DASH, HLS). → Our flexible, hosted Audiovisual Video Player at no extra cost (or use your own). → Insights about your customers' streaming habits with our Audiovisual analytics platform. See, PANDA-SR_000029 at PANDA-SR_000032, Audio-Video Brochure.
[C.2] a first socket address of the first event socket to establish the first event information Internet communication channel to carry the online gaming information.	The emBET webserver transmits “a first socket address of the first event socket to establish the first event information Internet communication channel to carry the online gaming information.” As explained above, emBET is offered in two deployment options, SDK and Platform. In both deployment options, a data feed is overlayed on a video feed. Thus, since emBET overlays data graphics onto a video, the video feed and data feed must be provided to the viewer client device on different communication channels, and these two separate feeds are synchronized by the emBET functionality.

Key Features	Accused Product
	emBET comes in two components: SDK An SDK that integrates with your platform's front-end, displaying all the graphical overlays directly onto your video player. Platform A back-end (administrative) platform that plugs into existing Sportradar feeds – where all overlays are configured. Our partner, LiveLike, then synchronises the video and data feeds, which convert into graphical overlays. See, PANDA-SR_000063 at PANDA-SR_000064, https://sportradar.com/sports-entertainment/audiovisual/embet/?lang=en-us. As an example, in the SDK option, emBET provides tools for the emBET purchaser to configure how video feeds and data feeds will be displayed on the client devices of end-user bettors. As explained below, the video feed and the data feed are synchronized. Thus, the emBET SDK allows an emBET purchaser to synchronize separate video and data feeds based on information obtained from the emBET content management system, described above.

Key Features	Accused Product
	 See, PANDA-SR_000062, https://embet.readme.io/docs/sdks.

2. Dependent Claim 4

Key Features	Accused Product
4. The system of claim 1, wherein:	See Claim 1, <i>supra</i> .
in C), the at least one webserver transmits, to a second viewer client	As explained above, emBET includes “at least one webserver [that] transmits ... the first socket address of the first event socket.” See discussion of Claim Element [1C], <i>supra</i> .

Key Features	Accused Product
device of the plurality of viewer client devices, the first socket address of the first event socket to establish a second event information Internet communication channel between the first event socket and the second viewer client device; and	Further, emBET's "at least one webserver transmits, to a second viewer client device of the plurality of viewer client devices, the first socket address of the first event socket to establish a second event information Internet communication channel between the first event socket and the second viewer client device." For example, emBET's source code reveals that the client establishes an unauthenticated, one-to-many connection to a socket channel, which then relays event information: <pre>const subscribe = (channel) => pubnub === null pubnub === void 0 ? void 0 : pubnub.subscribe({ channels: [channel] }); const safeSubscribe = (channel) => (pubnub === null pubnub === void 0 ? void 0 : pubnub.getSubscribedChannels) && !(pubnub === null pubnub === void 0 ? void 0 : pubnub.getSubscribedChannels().some((c) => c === channel)) && subscribe(channel); const unsubscribe = (channel) => pubnub === null pubnub === void 0 ? void 0 : pubnub.unsubscribe({ channels: [channel] }); const addListener = (listener) => pubnub === null pubnub === void 0 ? void 0 : pubnub.addListener(listener); const removeListener = (listener) => pubnub === null pubnub === void 0 ? void 0 : pubnub.removeListener(listener); const getMessages = (channel, opts = params({ count: 25, end: undefined, start: undefined })) => new Promise((res, rej) => { pubnub === null pubnub === void 0 ? void 0 : pubnub.fetchMessages({ channels: [channel], count: opts.count, stringifiedTimeToken: true, start: opts.start, end: opts.end, includeMessageActions: true, }, (status, response) => (status.error ? rej(status) : res(response))); });</pre>
in B), the at least one socket server transmits at least some of the first event information, including the online gaming information germane to the first sporting event, to the second viewer client device via the second event information Internet communication channel,	As explained above, emBET's "at least one socket server transmits at least some of the first event information, including the online gaming information germane to the first sporting event." See discussion of Claim Element [1B], <i>supra</i>. Further, emBET's "at least one socket server transmits at least some of the first event information, including the online gaming information germane to the first sporting event, to the second viewer client device via the second event information Internet communication channel, such that the online gaming information is shared in a synchronized manner by the first viewer client device and the second viewer client device," as claimed. For example, tracing the code through the "listeners" and down through the event dispatch system reveals that live display items (e.g., the first event information and the online gaming information

Key Features	Accused Product
such that the online gaming information is shared in a synchronized manner by the first viewer client device and the second viewer client device.	germane to the first sporting event) are added in response to data sent via the one-to-many socket connection: <pre data-bbox="569 302 1894 516"> const mainListener = { message: (msg) => { msg.message.event === 'program-status-updated' && eventDispatcher(document, msg.message.event, msg.message.payload); msg.timetoken && (msg.message.payload.timetoken = msg.timetoken); } }; </pre> <pre data-bbox="569 553 1894 1328"> let LiveTotal = class LiveTotal extends EmbetWidget {}; LiveTotal = __decorate\$1([element\$1('live-total')], LiveTotal); let LiveSpread = class LiveSpread extends EmbetWidget {}; LiveSpread = __decorate\$1([element\$1('live-spread')], LiveSpread); let LiveMoneyline = class LiveMoneyline extends EmbetWidget {}; LiveMoneyline = __decorate\$1([element\$1('live-moneyline')], LiveMoneyline); let LiveAtBat = class LiveAtBat extends EmbetWidget { constructor() { super(...arguments); this.insightsRenderer = insightsRenderer('table'); } }; LiveAtBat = __decorate\$1([element\$1('live-at-bat')], LiveAtBat); let LiveStatsOdds = class LiveStatsOdds extends EmbetWidget { constructor() { super(...arguments); this.insightsRenderer = insightsRenderer('stats'); } }; LiveStatsOdds = __decorate\$1([element\$1('live-stats-odds')], LiveStatsOdds); </pre>

Key Features	Accused Product
	As previously documented, the end viewers connect to a shared resource (the channel referenced in the code) instead of a personalized resource (i.e., a specific channel for a specific user, like channel “user123”). The nature and operations of a socket server and its method of transmission is such that for the receipt of a single message from the control server with a target of the “event channel,” the socket server then sends it to all recipients in a synchronized manner. If each end user were instead connected to a personalized “channel” (using the source code parlance), it would require a message from the control server to the socket server for each individual “channel,” therefore, resulting in the end users receiving the information in the order of messages sent by the control server (sequentially) instead of synchronized – and further requiring the socket server to duplicate the same information received from the control server several times (i.e., for each connected end user).