

1 SINGER CASHMAN LLP
2 Benjamin L. Singer (Bar. No. 264295)
3 bsinger@singercashman.com
4 Evan N. Budaj (Bar No. 271213)
5 ebudaj@singercashman.com
6 505 Montgomery Street, Suite 1100
7 San Francisco, California 94111
8 Telephone: (415) 500-6080
9 Facsimile: (415) 500-6080
10 *Attorneys for Plaintiff Software Research, Inc.*

11
12 UNITED STATES DISTRICT COURT
13
14 NORTHERN DISTRICT OF CALIFORNIA
15
16 SAN FRANCISCO DIVISION
17

18 SOFTWARE RESEARCH, INC.,
19
20 Plaintiff,
21
22 v.
23
24 UIPATH, INC. AND DOES 1 THROUGH
25 10,
26
27 Defendants.
28

CASE NO. 3:21-CV-4326

**COMPLAINT FOR PATENT
INFRINGEMENT**

JURY TRIAL DEMANDED

1 Plaintiff Software Research, Inc. (“SRI”), for its Complaint against UiPath, Inc. (“UiPath”)
2 and Does 1 through 10 (collectively, “Defendants”), upon information and belief, state and allege as
3 follows:

4 **NATURE OF THE ACTION**

5 1. This is a civil action for patent infringement arising under the patent laws of the
6 United States, Title 35 of the United States Code.

7 2. As set forth in more detail below, Defendants have been infringing United States
8 Patent Nos. 7,757,175 (the “’175 Patent”); 8,327,271 (the “’271 Patent”); 8,392,890 (the “’890
9 Patent”); 8,495,585 (the “’585 Patent”); 8,650,493 (the “’493 Patent”), 8,984,491 (the “’491 Patent”)
10 and 10,489,286 (the “’286 Patent”) (collectively, the “Patents-in-Suit”), and continue to do so
11 through the present date.

12 **THE PARTIES**

13 3. SRI is a corporation organized and existing under the laws of the State of California
14 with its principal place of business in this District.

15 4. Upon information and belief, UiPath is a Delaware corporation with its principal place
16 of business at 90 Park Ave , 20th Floor, New York, NY 10016 and an office at 75 East Santa Clara
17 Street, San Jose, CA 95113.

18 5. Upon information and belief, Defendants Does 1 through 10 are directors, officers,
19 employees, representatives, and/or agents of UiPath who participated and/or are currently
20 participating in the use, development, sale, offer for sale, import, offer for import, and/or other
21 commercialization of software offerings that infringe one or more of the Patents-in-Suit. The true
22 identities of Defendants Does 1 through 10 are presently unknown to SRI; SRI will amend its
23 complaint to state such names when they become known to SRI through discovery and/or continued
24 investigation.

25 6. Unless specifically stated otherwise, the acts complained of herein were committed by,
26 on behalf of, and/or for the benefit of Defendants.

27 **JURISDICTION AND VENUE**

28 7. This Court has subject matter jurisdiction pursuant to 28 U.S.C. §§ 1331 and 1338(a).

12. As the global economy has moved away from traditional brick and mortar business and into the digital age, a company's fortunes can rise and fall with the functionality and performance of its websites. An enterprise can lose millions in revenue if its website is down even briefly, its shopping cart or checkout processes are malfunctioning, or its pages fall victim to any of a number of other glitches arising from anything from simple mistake to sophisticated malware. As internet infrastructure, programming languages, and other website technology have grown by leaps and bounds at an ever-increasing pace, the process of identifying website functionality and performance issues—a tricky exercise to begin with—has become more complicated by the day.

1 13. SRI was founded by Edward Miller, PhD in 1979 and has been located in San
2 Francisco ever since. Dr. Miller founded SRI to develop, refine, and commercialize website
3 performance and functionality testing solutions that can keep up with the relentless advance of
4 internet technology.

5 14. Dr. Miller has been a leader in the software testing community for over 30 years,
6 beginning his career in the 1970s by verifying the quality of missile-borne software for anti-ICBM
7 defense. He organized the original Florida Software Testing Workshop in 1978. More recently in
8 San Francisco he organized and chaired fifteen QualityWeek conferences, the long-acknowledged top
9 technical conferences on software testing. These QualityWeek conferences were held annually from
10 1987 through 2002 with a total attendance over 25,000. Dr. Miller is also widely published in
11 conferences and publications of the Association for Computing Machinery and the Institute of
12 Electrical and Electronics Engineers.

13 15. In the decades since founding SRI, Dr. Miller has personally conceived of, developed,
14 and patented a number of revolutionary innovations that dramatically advanced the start of the art of
15 website performance testing. He has been granted nine patents for these innovations, including the
16 Patents-in-Suit. The foundational nature of these patents is reflected in their forward citations
17 (i.e., subsequent patents that cite them as prior art), which currently number over 400. These patents
18 have been licensed for many years by SRI's competitors in the software testing industry.

19 16. SRI has commercially exploited the Patents-in-Suit by making, marketing, selling, and
20 using products covered by the patents, including its popular eValid™ software testing products.
21 eValid™ is a tool suite for client-side testing and performance analysis of web applications and
22 websites. eValid™ has been a commercial success, generating millions of dollars in revenue, and
23 continues to generate revenue for SRI today. Thousands of customers—including Google, American
24 Express, Intel, Microsoft, Cisco, IBM, Lockheed Martin, Princeton University, and Verizon—have
25 purchased and employed SRI's offerings embodying one or more of the Patents-in-Suit, including but
26 not limited to eValid™, in order to ensure their websites function properly and continue to generate
27 revenue.

17. At its peak, SRI generated millions of dollars in annual revenues. But since that time, rampant infringement has decimated SRI's business, leaving SRI with no choice but to enforce its right to keep others from using Dr. Miller's inventions to compete against SRI.

18. Defendants develop robotic process automation software for, *inter alia*, testing websites and web applications known as, upon information and belief, the UiPath Platform. See <https://www.uipath.com/product>.

19. Defendants offer for sale and sell the UiPath Platform to the public.

20. Defendants use the UiPath Platform, including at least in order to test the UiPath Platform as part of their development efforts.

21. The UiPath Platform consists of robotic process automation software used to automate testing of websites and web-based software applications.

22. Defendants claim "website testing" is one of the "major common tasks that can readily benefit from web automation." See <https://www.uipath.com/solutions/technology/web-automation>.

23. The UiPath Platform includes functionality for creating, storing, and executing tests for websites and web-based software applications. See <https://www.uipath.com/solutions/technology/web-automation> ("Easy web testing. Create reliable tests without programming knowledge").

COUNT I – INFRINGEMENT OF THE '175 PATENT

24. SRI re-alleges and incorporates the allegations of the preceding paragraphs of this Complaint as if fully set forth herein.

25. SRI is the assignee and owner of all right, title, and interest in and to the '175 Patent, which was issued on July 13, 2010. A true and correct copy of the '175 Patent is attached hereto as Exhibit A.

26. The '175 Patent addresses an invention for testing websites. This disclosed innovation tests many facets of the website's experience and operation, including by providing novel approaches to creating, storing, and executing test scripts using website elements as opposed to the previously disclosed use of recording test scripts based upon user actions only.

1 27. SRI has the exclusive right to make, use, sell, and offer to sell any product embodying
2 the '175 Patent throughout the United States, and to import any product embodying the '175 Patent
3 into the United States.

4 28. SRI has commercially exploited the '175 Patent by making, marketing, selling, and
5 using products covered by the '175 Patent, including its popular eValid™ software products. SRI
6 continues to commercially exploit the '175 Patent through the present, at least by continuing to
7 provide maintenance and support to users of its popular eValid™ software products.

8 29. Defendants have had knowledge of the '175 Patent, SRI, and SRI's products
9 embodying the inventions claimed in the Patents-in-Suit since at least as early as the filing of this
10 Complaint.

11 30. At all relevant times, SRI provided public notice of the '175 Patent at least by properly
12 marking its products and its website pursuant to 35 U.S.C. § 287(a).

13 31. Defendants have been, and are currently, directly infringing at least claim 11 of the
14 '175 Patent in violation of 35 U.S.C. § 271(a), literally or under the doctrine of equivalents, by
15 making, using, selling, offering for sale, and/or importing into the United States certain robotic
16 process automation software, including without limitation Defendants' software for testing websites
17 and web-based software applications titled, upon information and belief, the UiPath Platform and/or
18 other related software products and services offered by Defendants (Defendants' "Infringing
19 Products"), which, as set forth in documentation available on Defendants' websites, comprise the
20 non-transitory computer readable media disclosed in the '175 Patent—both as maintained in
21 Defendants' files and as made accessible to its users to whom Defendants offer and sell the Infringing
22 Products—including at least computer program code stored therein for providing a test-enabled web
23 browser for operation on a computing device to test a website hosted by a remote server, the website
24 having at least one webpage (for example, the UiPath Platform can be used for "[e]asy web testing"
25 and to "[c]reate reliable tests without programming knowledge."
26 (<https://www.uipath.com/solutions/technology/web-automation>); "This is where web automation
27 comes in. Form filling, ... website testing and periodical report generation are major common tasks
28 that can readily benefit from web automation." (*id.*)); the UiPath Platform as used with a web

1 browser is a test-enabled web browser (for example, “In the UiPath Platform, the WebDriver protocol
 2 introduces support for headless browser automations. ... Although the WebDriver protocol can be
 3 used for both headless and visual browser automations, it is not meant to replace the UiPath browser
 4 extensions, giving you the freedom to use whichever option suits you best.”
 5 (<https://docs.uipath.com/studio/docs/about-the-webdriver-protocol>); “UiPath Web Automation uses a
 6 built-in recorder that can read and enact web-based activities. It identifies web elements by their
 7 attributes and accurately manipulates them while keeping up with website changes.”
 8 (<https://www.uipath.com/solutions/technology/web-automation>)); the website, necessarily including
 9 at least one webpage, necessarily resides on a remote server and the UiPath Platform as used with a
 10 web browser, such as Chrome or Firefox, is a “test-enabled web browser” (for example, “UiPath
 11 Extensions ... Chrome: Installs the browser extension for automating websites in Chrome. ...
 12 Firefox: Installs the browser extension for automating websites in Firefox.”
 13 (<https://docs.uipath.com/studio/docs>); “Web Automation ... The example below explains how to
 14 automate the action of browsing a webpage, retrieving the needed information, and displaying it into
 15 a message box.” (<https://docs.uipath.com/activities/docs/web-automation>); web browsing
 16 components (for example, the UiPath Platform allows a user to browse the web via common web
 17 browsing activities, such as navigating to a web page, clicking and double clicking a mouse, or
 18 selecting an item from a drop-down list or combo box
 19 (<https://docs.uipath.com/activities/docs/navigate-to>;
 20 <https://docs.uipath.com/studio/v2018.3/docs/manual-recording>;
 21 <https://docs.uipath.com/activities/docs/double-click>); the UiPath Platform further allows a user to
 22 create or record a test using such common web browsing activities and then play it back by executing
 23 it (<https://docs.uipath.com/studio/v2018.3/docs/about-recording-types>;
 24 <https://docs.uipath.com/studio/docs/example-of-automatic-recording-with-web>;
 25 <https://docs.uipath.com/test-suite/docs/executing-tests>)); a page evaluation component that operates
 26 to read, extract, and analyze and confirm the contents of page components, including Document
 27 Object Model (DOM) elements with their associated at least one index and their values (for example,
 28 the UiPath Platform allows a user to create or record a test and then play it back by executing it (for

1 example, “Recording can help you save a lot of time when automating your tasks. The App/Web
 2 Recorder captures your actions as you perform them on the screen and generates a Use
 3 Application/Browser activity with a series of activities inside it based on your actions.”
 4 (<https://docs.uipath.com/activities/docs/app-web-recorder>); “There are four types of recordings
 5 available in UiPath Studio: ... Web – designed for recording in web apps and browsers, generates
 6 containers and uses the Simulate Type/Click input method by default.”
 7 (<https://docs.uipath.com/studio/v2018.3/docs/about-recording-types>); “Executing tests can be done
 8 either manually or automatically. Manual execution is invoked via Test Manager while automated
 9 execution is invoked via UiPath Orchestrator only.” ([https://docs.uipath.com/test-](https://docs.uipath.com/test-suite/docs/executing-tests)
 10 [suite/docs/executing-tests](https://docs.uipath.com/test-suite/docs/executing-tests))); the UiPath Platform interrogates the DOM to identify and extract
 11 relevant information regarding at least the page elements germane to the test, including each such
 12 element’s value, and stores those details in the test ([https://docs.uipath.com/activities/docs/get-](https://docs.uipath.com/activities/docs/get-attribute)
 13 [attribute](https://docs.uipath.com/activities/docs/get-attribute); <https://docs.uipath.com/activities/lang-en/docs/get-value>;
 14 <https://docs.uipath.com/activities/docs/extract-data-from-website>;
 15 <https://docs.uipath.com/activities/docs/digitize-document>); the UiPath Platform locates these pages
 16 elements based on their DOM indexes, which necessarily requires it to use the DOM access methods
 17 included in Dynamic Linked Libraries associated with a browser code library
 18 (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>;
 19 <https://docs.uipath.com/activities/docs/ui-element-exists>; [https://docs.uipath.com/activities/docs/on-](https://docs.uipath.com/activities/docs/on-ui-element-appear)
 20 [ui-element-appear](https://docs.uipath.com/activities/docs/on-ui-element-appear)); a test data component that operates to store facts about the at least one webpage
 21 (for example, the UiPath Platform uses explicit wait commands, such as a command that “[w]aits for
 22 the specified UI element to appear on the screen” or to “[w]aits for the value of specified UI element
 23 attribute to be equal to a string” and, in order to perform such a verification, the UiPath Platform
 24 must necessarily store facts about the webpage being rendered, i.e., the expected condition to be
 25 checked for during verification (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>;
 26 <https://docs.uipath.com/activities/docs/wait-attribute>; [https://docs.uipath.com/activities/docs/execute-](https://docs.uipath.com/activities/docs/execute-x-path)
 27 [x-path](https://docs.uipath.com/activities/docs/execute-x-path))); and a graphical user interface to provide user access to at least said web browsing
 28 components and at least one of said page evaluation components and said test data component (for

1 example, the UiPath Platform has a graphical user interface to provide user access to test details
 2 including results for each test case and the complete log for each test set ([https://docs.uipath.com/test-](https://docs.uipath.com/test-suite/docs/analyzing-test-results)
 3 [suite/docs/analyzing-test-results](https://docs.uipath.com/test-suite/docs/analyzing-test-results))), as disclosed in the '175 Patent.

4 32. Defendants will, on information and belief, continue to directly infringe the '175
 5 Patent unless enjoined.

6 33. To the extent Defendants' Infringing Products, without more, do not directly infringe
 7 at least claim 11 of the '175 Patent, at least as of the filing of this Complaint, Defendants contribute
 8 to infringement of the same under 35 U.S.C. § 271(c) inasmuch as the Infringing Products offered for
 9 sale and sold by Defendants are each a component of a patented machine or an apparatus used in
 10 practicing a patented process, constituting a material part of SRI's invention, knowing the same to be
 11 especially made or especially adapted for use in infringement of the '175 Patent. For example, as set
 12 forth above, the UiPath Platform, when used in its normal and intended usage (pursuant to the
 13 instructions set forth on Defendants' websites), infringes claim 11 of the '175 Patent. *See supra*,
 14 ¶ 31.

15 34. Defendants will, on information and belief, continue to contribute to infringement the
 16 '175 Patent unless enjoined.

17 35. Defendants actively encourage their customers to use Defendants' Infringing Products
 18 in an infringing manner. For example, Defendants' website is replete with written directions,
 19 screenshots, and videos instructing users on how to use the Infringing Products in an infringing
 20 manner. For example, as set forth above, Defendants' website regarding the UiPath Platform
 21 specifically instructs users of the Infringing Products how to infringe claim 11 of the '175 patent. *See*
 22 *supra*, ¶ 31. Defendants' website also touts the identities of customers who use the Infringing

23 //

24 //

25 //

26 //

27 //

28 //

Products, each of whom is a direct infringer inasmuch as they use the Infringing Products in the infringing manner as instructed by Defendants:



SUCCESS STORY

Deloitte

Deloitte teams set out to use Robotic Process Automation (RPA) to make the organization faster, leaner, and better than ever. But first, they needed to get the rest of...



SUCCESS STORY

The New York Foundling

The New York Foundling gets back close to 100,000 hours in manual work annually by leveraging RPA



SUCCESS STORY

American Fidelity

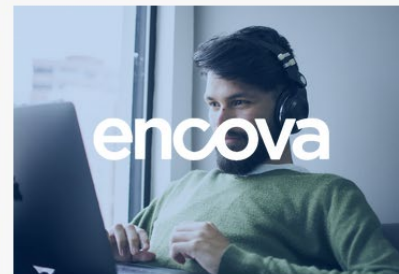
With UiPath and DataRobot, American Fidelity successfully automated many of its most critical customer-facing processes.



SUCCESS STORY

PwC

Empowering employees by rolling out RPA to its 50,000+ US and Mexico employees and is in the process of expanding this globally



SUCCESS STORY

Encova

Delights Agents and Improves CX by Automating Tedious, Time-Consuming Tasks



SUCCESS STORY

dentsu

Learn how dentsu built 60 robots in 30 days and saved 125K hours in the process

36. Upon information and belief, and particularly by way of the detailed documentation instructing users on how to use the Infringing Products in an infringing manner (*see supra*, ¶¶ 31, 35), Defendants have encouraged this infringement with knowledge of the '175 Patent and with a specific intent to cause their customers and distributors to infringe.

37. Defendants' acts thus constitute active inducement of patent infringement in violation of 35 U.S.C. § 271(b).

38. Defendants will, on information and belief, continue to induce infringement of the '175 Patent unless enjoined.

39. Defendants' direct infringement, contributory infringement, and inducement of infringement have irreparably harmed SRI.

40. Defendants will, on information and belief, continue to irreparably harm SRI unless enjoined.

41. Pursuant to 35 U.S.C. § 284, SRI is entitled to damages adequate to compensate for the infringement but in no event less than a reasonable royalty.

42. This case is "exceptional" within the meaning of 35 U.S.C. § 285, and SRI is entitled to an award of attorneys' fees.

COUNT II – INFRINGEMENT OF THE '271 PATENT

43. SRI re-alleges and incorporates the allegations of the preceding paragraphs of this Complaint as if fully set forth herein.

44. SRI is the assignee and owner of all right, title, and interest in and to the '271 Patent, which was issued on December 4, 2012. A true and correct copy of the '271 Patent is attached hereto as Exhibit B.

45. The '271 Patent addresses an invention for testing websites. This disclosed innovation tests many facets of the website's experience and operation, including by providing novel approaches to creating, storing, and executing test scripts using website elements as opposed to the previously disclosed use of recording test scripts based upon user actions only.

46. SRI has the exclusive right to make, use, sell, and offer to sell any product embodying the '271 Patent throughout the United States, and to import any product embodying the '271 Patent into the United States.

47. SRI has commercially exploited the '271 Patent by making, marketing, selling, and using products covered by the '271 Patent, including its popular eValid™ software products. SRI continues to commercially exploit the '271 Patent through the present, at least by continuing to provide maintenance and support to users of its popular eValid™ software products.

48. Defendants have had knowledge of the '271 Patent, SRI, and SRI's products embodying the inventions claimed in the Patents-in-Suit since at least as early as the filing of this Complaint.

1 49. At all relevant times, SRI provided public notice of the '271 Patent at least by properly
2 marking its products and its website pursuant to 35 U.S.C. § 287(a).

3 50. Defendants have been, and are currently, directly infringing at least claim 1 of the
4 '271 Patent in violation of 35 U.S.C. § 271(a), literally or under the doctrine of equivalents, by
5 making, using, selling, offering for sale, and/or importing into the United States Defendants'
6 Infringing Products, which, as set forth in documentation available on Defendants' website, comprise
7 the non-transitory computer readable media disclosed in the '271 Patent—both as maintained in
8 Defendants' files and as made accessible to its users to whom Defendants offer and sell the Infringing
9 Products—including at least computer program code stored therein for providing a test-enabled web
10 browser for testing a website residing on a network webpage (for example, the UiPath Platform can
11 be used for “[e]asy web testing” and to “[c]reate reliable tests without programming knowledge.”
12 (<https://www.uipath.com/solutions/technology/web-automation>); “This is where web automation
13 comes in. Form filling, ... website testing and periodical report generation are major common tasks
14 that can readily benefit from web automation.” (*id.*)); the UiPath Platform as used with a web
15 browser is a test-enabled web browser (for example, “In the UiPath Platform, the WebDriver protocol
16 introduces support for headless browser automations. ... Although the WebDriver protocol can be
17 used for both headless and visual browser automations, it is not meant to replace the UiPath browser
18 extensions, giving you the freedom to use whichever option suits you best.”
19 (<https://docs.uipath.com/studio/docs/about-the-webdriver-protocol>); “UiPath Web Automation uses a
20 built-in recorder that can read and enact web-based activities. It identifies web elements by their
21 attributes and accurately manipulates them while keeping up with website changes.”
22 (<https://www.uipath.com/solutions/technology/web-automation>)); the website, necessarily including
23 at least one webpage, necessarily resides on a remote server and the UiPath Platform as used with a
24 web browser, such as Chrome or Firefox, is a “test-enabled web browser” (for example, “UiPath
25 Extensions ... Chrome: Installs the browser extension for automating websites in Chrome. ...
26 Firefox: Installs the browser extension for automating websites in Firefox.”
27 (<https://docs.uipath.com/studio/docs>); “Web Automation ... The example below explains how to
28 automate the action of browsing a webpage, retrieving the needed information, and displaying it into

1 a message box.” (<https://docs.uipath.com/activities/docs/web-automation>)); computer program code
 2 for interfacing with web browsing components, the web browsing components including DOM
 3 access methods of the web browsing components (for example, the UiPath Platform allows a user to
 4 browse the web via common web browsing activities, such as navigating to a web page, clicking and
 5 double clicking a mouse, or selecting an item from a drop-down list or combo box
 6 (<https://docs.uipath.com/activities/docs/navigate-to>;
 7 <https://docs.uipath.com/studio/v2018.3/docs/manual-recording>;
 8 <https://docs.uipath.com/activities/docs/double-click>)); the UiPath Platform interrogates the DOM to
 9 identify and extract relevant information regarding at least the page elements germane to the test,
 10 including each such element’s value, and stores those details in the test
 11 (<https://docs.uipath.com/activities/docs/get-attribute>; [https://docs.uipath.com/activities/lang-
 12 en/docs/get-value](https://docs.uipath.com/activities/lang-en/docs/get-value); <https://docs.uipath.com/activities/docs/extract-data-from-website>;
 13 <https://docs.uipath.com/activities/docs/digitize-document>); the UiPath Platform locates these pages
 14 elements based on their DOM indexes, which necessarily requires it to use the DOM access methods
 15 included in Dynamic Linked Libraries associated with a browser code library
 16 (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>;
 17 <https://docs.uipath.com/activities/docs/ui-element-exists>; [https://docs.uipath.com/activities/docs/on-
 18 ui-element-appear](https://docs.uipath.com/activities/docs/on-ui-element-appear))); computer program code for rendering and examining at least one webpage of the
 19 website so as to at least extract details of organization and structure of elements of the webpage, and
 20 store such details of the webpage in a recorded script, such as recorded scripts generated through the
 21 testing component of Defendants’ Infringing Products (for example, the UiPath Platform allows a
 22 user to record a test using a browser and then play it back by running it; to achieve such functionality,
 23 the UiPath Platform necessarily renders and examines the web page for the creation of tests by
 24 recording a user’s interactions with the web page in question and allowing the user to play back those
 25 tests (“Recording can help you save a lot of time when automating your tasks. The App/Web
 26 Recorder captures your actions as you perform them on the screen and generates a Use
 27 Application/Browser activity with a series of activities inside it based on your actions.”
 28 (<https://docs.uipath.com/activities/docs/app-web-recorder>); “There are four types of recordings

1 available in UiPath Studio: ... Web – designed for recording in web apps and browsers, generates
 2 containers and uses the Simulate Type/Click input method by default.”
 3 (<https://docs.uipath.com/studio/v2018.3/docs/about-recording-types>)); the UiPath Platform
 4 interrogates the DOM to identify and extract relevant information regarding at least the page
 5 elements germane to the test, including each such element’s value, and stores those details in the test
 6 (<https://docs.uipath.com/activities/docs/get-attribute>; [https://docs.uipath.com/activities/lang-
 7 en/docs/get-value](https://docs.uipath.com/activities/lang-en/docs/get-value); <https://docs.uipath.com/activities/docs/extract-data-from-website>;
 8 <https://docs.uipath.com/activities/docs/digitize-document>); the UiPath Platform uses explicit wait
 9 commands, such as a command that “[w]aits for the specified UI element to appear on the screen” or
 10 to “[w]aits for the value of specified UI element attribute to be equal to a string” and, in order to
 11 perform such a verification, the UiPath Platform must necessarily store facts about the webpage being
 12 rendered, i.e., the expected condition to be checked for during verification
 13 (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>;
 14 <https://docs.uipath.com/activities/docs/wait-attribute>; [https://docs.uipath.com/activities/docs/execute-
 15 x-path](https://docs.uipath.com/activities/docs/execute-x-path)); the UiPath Platform locates these pages elements based on their DOM indexes, which
 16 necessarily requires it to use the DOM access methods included in Dynamic Linked Libraries
 17 associated with a browser code library ([https://docs.uipath.com/activities/docs/wait-ui-element-
 18 appear#misc](https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc); <https://docs.uipath.com/activities/docs/ui-element-exists>;
 19 <https://docs.uipath.com/activities/docs/on-ui-element-appear>); computer program code for selecting a
 20 validation test to be performed (for example, the UiPath Platform allows for the creation of test
 21 scripts to test websites by recording a user’s interactions with the webpage in question and allowing
 22 the user to play back those test scripts ([https://docs.uipath.com/studio/v2018.3/docs/about-recording-
 23 types](https://docs.uipath.com/studio/v2018.3/docs/about-recording-types); <https://docs.uipath.com/studio/docs/example-of-automatic-recording-with-web>;
 24 <https://docs.uipath.com/test-suite/docs/executing-tests>)); the UiPath Platform uses explicit wait
 25 commands, such as a command that “[w]aits for the specified UI element to appear on the screen” or
 26 to “[w]aits for the value of specified UI element attribute to be equal to a string” and, in order to
 27 perform such a verification, the UiPath Platform must necessarily store facts about the webpage being
 28 rendered, i.e., the expected condition to be checked for during verification

(<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>;
<https://docs.uipath.com/activities/docs/wait-attribute>; <https://docs.uipath.com/activities/docs/execute-x-path>); and computer program code for performing the validation test using at least one of the DOM access methods of the web browsing components, wherein during the validation test, the at least one webpage is newly rendered and details of organization and structure of elements for the at least one webpage as newly rendered are accessed via the at least one of the DOM access methods and compared to the stored details in the recorded script (for example, the UiPath Platform allows a user to create or record a test and then play it back by executing it (“Recording can help you save a lot of time when automating your tasks. The App/Web Recorder captures your actions as you perform them on the screen and generates a Use Application/Browser activity with a series of activities inside it based on your actions.” (<https://docs.uipath.com/activities/docs/app-web-recorder>); “There are four types of recordings available in UiPath Studio: ... Web – designed for recording in web apps and browsers, generates containers and uses the Simulate Type/Click input method by default.” (<https://docs.uipath.com/studio/v2018.3/docs/about-recording-types>); “Executing tests can be done either manually or automatically. Manual execution is invoked via Test Manager while automated execution is invoked via UiPath Orchestrator only.” (<https://docs.uipath.com/test-suite/docs/executing-tests>)); the UiPath Platform interrogates the DOM to identify and extract relevant information regarding at least the page elements germane to the test, including each such element’s value, and stores those details in the test (<https://docs.uipath.com/activities/docs/get-attribute>; <https://docs.uipath.com/activities/lang-en/docs/get-value>; <https://docs.uipath.com/activities/docs/extract-data-from-website>; <https://docs.uipath.com/activities/docs/digitize-document>); the UiPath Platform uses explicit wait commands to search for the expected elements against which it validates the webpage being rendered (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/wait-attribute>; <https://docs.uipath.com/activities/docs/execute-x-path>); and the UiPath Platform locates these pages elements based on their DOM indexes, which necessarily requires it to use the DOM access methods included in Dynamic Linked Libraries associated with a browser code library (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>;

1 [appear#misc; https://docs.uipath.com/activities/docs/ui-element-exists;](https://docs.uipath.com/activities/docs/ui-element-exists)
 2 <https://docs.uipath.com/activities/docs/on-ui-element-appear>); as disclosed in the '271 Patent.

3 51. Defendants will, on information and belief, continue to directly infringe the '271
 4 Patent unless enjoined.

5 52. To the extent Defendants' Infringing Products, without more, do not directly infringe
 6 at least claim 1 of the '271 Patent, at least as of the filing of this Complaint, Defendants contribute to
 7 infringement of the same under 35 U.S.C. § 271(c) inasmuch as the Infringing Products offered for
 8 sale and sold by Defendants are each a component of a patented machine or an apparatus used in
 9 practicing a patented process, constituting a material part of SRI's invention, knowing the same to be
 10 especially made or especially adapted for use in infringement of the '271 Patent. For example, as set
 11 forth above, the UiPath Platform, when used in its normal and intended usage (pursuant to the
 12 instructions set forth on Defendants' website), infringes claim 1 of the '271 Patent. *See supra*, ¶ 50.

13 53. Defendants will, on information and belief, continue to contribute to infringement of
 14 the '271 Patent unless enjoined.

15 54. Defendants actively encourage their customers to use Defendants' Infringing Products
 16 in an infringing manner. For example, Defendants' website is replete with written directions,
 17 screenshots, and videos instructing users on how to use the Infringing Products in an infringing
 18 manner. For example, as set forth above, Defendants' website regarding the UiPath Platform
 19 specifically instructs users of the Infringing Products how to infringe claim 1 of the '271 patent. *See*
 20 *supra*, ¶ 50. Defendants' website also touts the identities of customers who use the Infringing
 21 Products, each of whom is a direct infringer inasmuch as they use the Infringing Products in the
 22 infringing manner as instructed by Defendants:

23 //

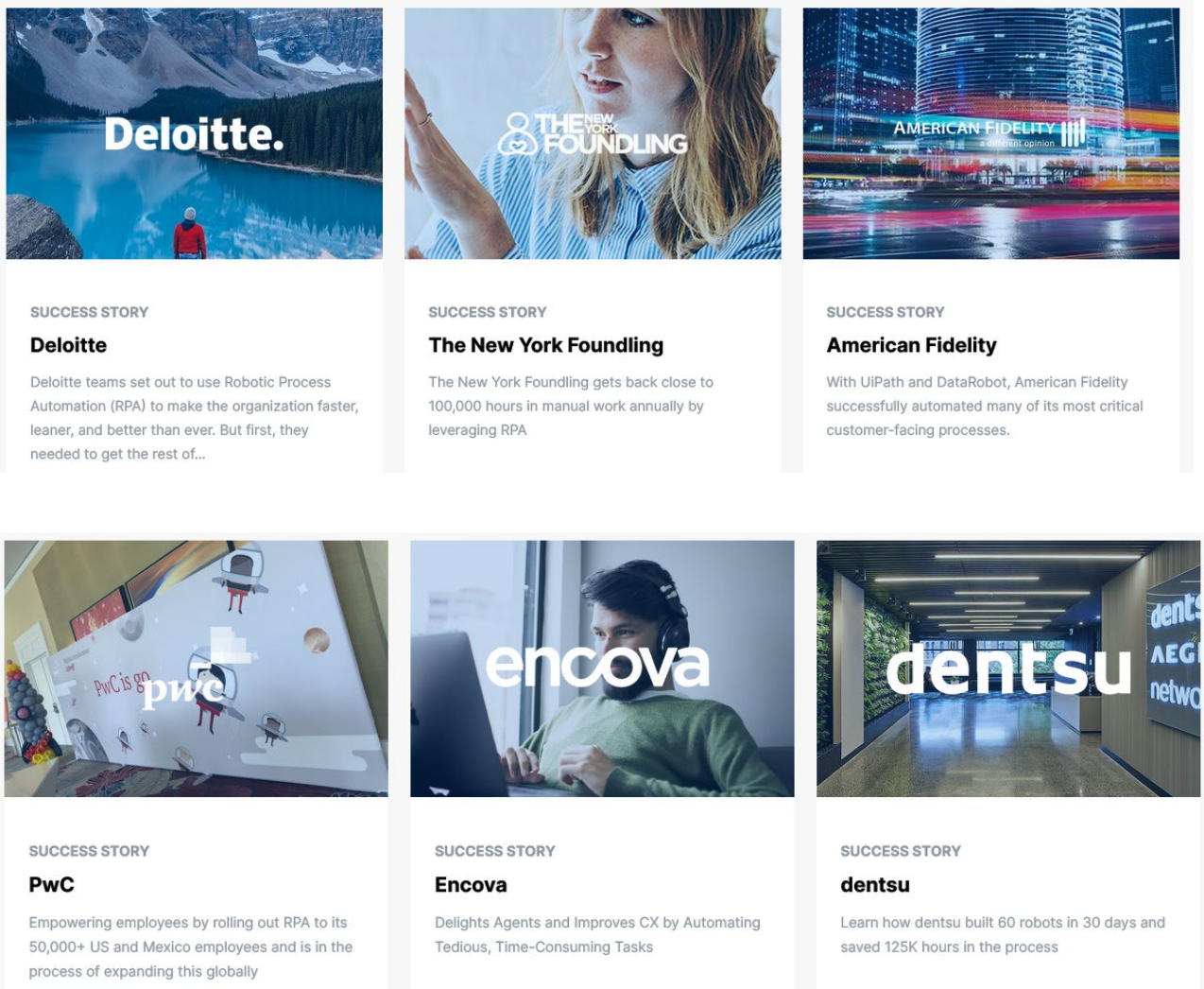
24 //

25 //

26 //

27 //

28 //



55. Upon information and belief, and particularly by way of the detailed documentation instructing users on how to use the Infringing Products in an infringing manner (*see supra*, ¶¶ 50, 54), Defendants have encouraged this infringement with knowledge of the '271 Patent and with a specific intent to cause their customers and distributors to infringe.

56. Defendants' acts thus constitute active inducement of patent infringement in violation of 35 U.S.C. § 271(b).

57. Defendants will, on information and belief, continue to induce infringement of the '271 Patent unless enjoined.

1 58. Defendants' direct infringement, contributory infringement, and inducement of
2 infringement have irreparably harmed SRI.

3 59. Defendants will, on information and belief, continue to irreparably harm SRI unless
4 enjoined.

5 60. Pursuant to 35 U.S.C. § 284, SRI is entitled to damages adequate to compensate for
6 the infringement but in no event less than a reasonable royalty.

7 61. This case is "exceptional" within the meaning of 35 U.S.C. § 285, and SRI is entitled
8 to an award of attorneys' fees.

9 **COUNT III – INFRINGEMENT OF THE '890 PATENT**

10 62. SRI re-alleges and incorporates the allegations of the preceding paragraphs of this
11 Complaint as if fully set forth herein.

12 63. SRI is the assignee and owner of all right, title, and interest in and to the '890 Patent,
13 which was issued on March 5, 2013. A true and correct copy of the '890 Patent is attached hereto as
14 Exhibit C.

15 64. The '890 Patent addresses an invention for testing websites. The disclosed innovation
16 tests many facets of the website's experience and operation, including by providing novel approaches
17 to creating, storing, and executing test scripts capable of accurately testing Asynchronous Javascript
18 and XML ("AJAX") webpage elements.

19 65. SRI has the exclusive right to make, use, sell, and offer to sell any product embodying
20 the '890 Patent throughout the United States, and to import any product embodying the '890 Patent
21 into the United States.

22 66. SRI has commercially exploited the '890 Patent by making, marketing, selling, and
23 using products covered by the '890 Patent, including its popular eValid™ software products. SRI
24 continues to commercially exploit the '890 Patent through the present, at least by continuing to
25 provide maintenance and support to users of its popular eValid™ software products.

26 67. Defendants have had knowledge of the '890 Patent, SRI, and SRI's products
27 embodying the inventions claimed in the Patents-in-Suit since at least as early as the filing of this
28 Complaint.

68. At all relevant times, SRI provided public notice of the '890 Patent by properly marking its products and its website pursuant to 35 U.S.C. § 287(a). Defendants have been, and are currently, directly infringing at least claim 1 of the '890 Patent in violation of 35 U.S.C. § 271(a), literally or under the doctrine of equivalents, by making, using, selling, offering for sale, and/or importing into the United States Defendants' Infringing Products, which, as set forth in documentation available on Defendants' website, comprise the non-transitory computer readable media disclosed in the '890 Patent—both as maintained in Defendants' files and as made accessible to its users to whom Defendants offer and sell the Infringing Products—including at least computer program code stored therein for providing a test-enabled web browser, said medium comprising computer program code for providing web browsing capabilities (for example, "In the UiPath Platform, the WebDriver protocol introduces support for headless browser automations. ... Although the WebDriver protocol can be used for both headless and visual browser automations, it is not meant to replace the UiPath browser extensions, giving you the freedom to use whichever option suits you best." (<https://docs.uipath.com/studio/docs/about-the-webdriver-protocol>); "UiPath Web Automation uses a built-in recorder that can read and enact web-based activities. It identifies web elements by their attributes and accurately manipulates them while keeping up with website changes." (<https://www.uipath.com/solutions/technology/web-automation>); "Web Automation ... The example below explains how to automate the action of browsing a webpage, retrieving the needed information, and displaying it into a message box." (<https://docs.uipath.com/activities/docs/web-automation>)); the UiPath Platform as used with a web browser is a test-enabled web browser (for example, "In the UiPath Platform, the WebDriver protocol introduces support for headless browser automations. ... Although the WebDriver protocol can be used for both headless and visual browser automations, it is not meant to replace the UiPath browser extensions, giving you the freedom to use whichever option suits you best." (<https://docs.uipath.com/studio/docs/about-the-webdriver-protocol>); "UiPath Web Automation uses a built-in recorder that can read and enact web-based activities. It identifies web elements by their attributes and accurately manipulates them while keeping up with website changes." (<https://www.uipath.com/solutions/technology/web-automation>)); the UiPath Platform as used with a

1 web browser, such as Chrome or Firefox, is a “test-enabled web browser” (for example, “UiPath
 2 Extensions ... Chrome: Installs the browser extension for automating websites in Chrome. ...
 3 Firefox: Installs the browser extension for automating websites in Firefox.”
 4 (<https://docs.uipath.com/studio/docs>); “Web Automation ... The example below explains how to
 5 automate the action of browsing a webpage, retrieving the needed information, and displaying it into
 6 a message box.” (<https://docs.uipath.com/activities/docs/web-automation>)) and allows a user to
 7 browse the web via common web browsing activities, such as navigating to a web page, clicking and
 8 double clicking a mouse, or selecting an item from a drop-down list or combo box
 9 (<https://docs.uipath.com/activities/docs/navigate-to>;
 10 <https://docs.uipath.com/studio/v2018.3/docs/manual-recording>;
 11 <https://docs.uipath.com/activities/docs/double-click>)); computer program code for testing
 12 capabilities of a website hosted by a server and accessible to the computer via a network wherein the
 13 computer program code for testing capabilities of the website includes at least computer program
 14 code configured to receive a synchronization check from a user using the test enabled browser, to
 15 insert the synchronization check into a test script for testing at least one webpage of the website (for
 16 example, the UiPath Platform allows for the creation of tests for websites by recording a user’s
 17 interactions with the web page in question and allowing the user to play back those tests
 18 (<https://docs.uipath.com/activities/docs/app-web-recorder>;
 19 <https://docs.uipath.com/studio/v2018.3/docs/about-recording-types>); the UiPath Platform interrogates
 20 the DOM to identify and extract relevant information regarding at least the page elements germane to
 21 the test, including each such element’s value, and stores those details in the test
 22 (<https://docs.uipath.com/activities/docs/get-attribute>; [https://docs.uipath.com/activities/lang-](https://docs.uipath.com/activities/lang-en/docs/get-value)
 23 [en/docs/get-value](https://docs.uipath.com/activities/docs/extract-data-from-website); <https://docs.uipath.com/activities/docs/extract-data-from-website>;
 24 <https://docs.uipath.com/activities/docs/digitize-document>); the UiPath Platform uses explicit wait
 25 commands, such as a command that “[w]aits for the specified UI element to appear on the screen” or
 26 to “[w]aits for the value of specified UI element attribute to be equal to a string” and, in order to
 27 perform such a verification, the UiPath Platform must necessarily store facts about the webpage being
 28 rendered, i.e., the expected condition to be checked for during verification

(<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>;
<https://docs.uipath.com/activities/docs/wait-attribute>; <https://docs.uipath.com/activities/docs/execute-x-path>); the UiPath Platform locates these pages elements based on their DOM indexes, which necessarily requires it to use the DOM access methods included in Dynamic Linked Libraries associated with a browser code library (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/ui-element-exists>; <https://docs.uipath.com/activities/docs/on-ui-element-appear>); and the UiPath Platform allows for the testing of content dynamically generated by AJAX programming including using, upon information and belief, its various wait commands to synchronize playback and allow for testing of content dynamically generated by AJAX programming (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/wait-attribute>; <https://www.uipath.com/solutions/technology/web-scraping-software>; <https://www.uipath.com/solutions/technology/web-automation>); the test script being separate from the at least one webpage being tested (for example, the UiPath Platform stores and accesses test scripts separately from the webpage itself (<https://www.uipath.com/solutions/technology/web-automation>; <https://docs.uipath.com/studio/docs/about-the-webdriver-protocol>; <https://docs.uipath.com/studio/lang-ru/docs/creating-basic-process>)), the at least one webpage being tested including AJAX programming, and to automatically synchronize playback of the test script using at least the synchronization check to maintain the test enabled browser's state with respect to the AJAX programming by means of the synchronization check in the test script to a Document Object Model (DOM) associated with the at least one webpage of the website (for example, the UiPath Platform allows a user to create or record a test and then play it back by executing it (“Recording can help you save a lot of time when automating your tasks. The App/Web Recorder captures your actions as you perform them on the screen and generates a Use Application/Browser activity with a series of activities inside it based on your actions.” (<https://docs.uipath.com/activities/docs/app-web-recorder>); “There are four types of recordings available in UiPath Studio: ... Web – designed for recording in web apps and browsers, generates containers and uses the Simulate Type/Click input method by default.”

(<https://docs.uipath.com/studio/v2018.3/docs/about-recording-types>); “Executing tests can be done either manually or automatically. Manual execution is invoked via Test Manager while automated execution is invoked via UiPath Orchestrator only.” (<https://docs.uipath.com/test-suite/docs/executing-tests>)); the UiPath Platform interrogates the DOM to identify and extract relevant information regarding at least the page elements germane to the test, including each such element’s value, and stores those details in the test (<https://docs.uipath.com/activities/docs/get-attribute>; <https://docs.uipath.com/activities/lang-en/docs/get-value>; <https://docs.uipath.com/activities/docs/extract-data-from-website>; <https://docs.uipath.com/activities/docs/digitize-document>); the UiPath Platform uses explicit wait commands, such as a command that “[w]aits for the specified UI element to appear on the screen” or to “[w]aits for the value of specified UI element attribute to be equal to a string” and, in order to perform such a verification, the UiPath Platform must necessarily store facts about the webpage being rendered, i.e., the expected condition to be checked for during verification (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/wait-attribute>; <https://docs.uipath.com/activities/docs/execute-x-path>); the UiPath Platform locates these page elements based on their DOM indexes, which necessarily requires it to use the DOM access methods included in Dynamic Linked Libraries associated with a browser code library (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/ui-element-exists>; <https://docs.uipath.com/activities/docs/on-ui-element-appear>); and the UiPath Platform allows for the testing of content dynamically generated by AJAX programming including using, upon information and belief, its various wait commands to synchronize playback and allow for testing of content dynamically generated by AJAX programming (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/wait-attribute>; <https://www.uipath.com/solutions/technology/web-scraping-software>; <https://www.uipath.com/solutions/technology/web-automation>); wherein the synchronization check in the test script and web browsing activities provided by the web browsing capabilities are able to separately access the DOM associated with the at least one webpage of the website (for example, for

1 example, the UiPath Platform stores and accesses test scripts separately from the webpage itself
 2 (<https://www.uipath.com/solutions/technology/web-automation>;
 3 <https://docs.uipath.com/studio/docs/about-the-webdriver-protocol>;
 4 <https://docs.uipath.com/studio/lang-ru/docs/creating-basic-process>)); wherein the synchronization
 5 check is inserted into the test script as at least one command, and the at least one command operates,
 6 when executed, to: find a current index of at least one DOM element of the at least one webpage
 7 based on a specified property name and/or property value; and (i) submit a named event to the at least
 8 one DOM element of the at least one webpage having the current index, or (ii) insert or verify a value
 9 in the at least one DOM element of the at least one webpage having the current index (for example,
 10 the UiPath Platform allows a user to create or record a test using a browser and then play it back by
 11 executing it (“Recording can help you save a lot of time when automating your tasks. The App/Web
 12 Recorder captures your actions as you perform them on the screen and generates a Use
 13 Application/Browser activity with a series of activities inside it based on your actions.”
 14 (<https://docs.uipath.com/activities/docs/app-web-recorder>); “There are four types of recordings
 15 available in UiPath Studio: ... Web – designed for recording in web apps and browsers, generates
 16 containers and uses the Simulate Type/Click input method by default.”
 17 (<https://docs.uipath.com/studio/v2018.3/docs/about-recording-types>); “Executing tests can be done
 18 either manually or automatically. Manual execution is invoked via Test Manager while automated
 19 execution is invoked via UiPath Orchestrator only.” ([https://docs.uipath.com/test-](https://docs.uipath.com/test-suite/docs/executing-tests)
 20 [suite/docs/executing-tests](https://docs.uipath.com/test-suite/docs/executing-tests))); the UiPath Platform interrogates the DOM to identify and extract
 21 relevant information regarding at least the page elements germane to the test, including each such
 22 element’s value, and stores those details in the test ([https://docs.uipath.com/activities/docs/get-](https://docs.uipath.com/activities/docs/get-attribute)
 23 [attribute](https://docs.uipath.com/activities/docs/get-attribute); <https://docs.uipath.com/activities/lang-en/docs/get-value>;
 24 <https://docs.uipath.com/activities/docs/extract-data-from-website>;
 25 <https://docs.uipath.com/activities/docs/digitize-document>); the UiPath Platform uses explicit wait
 26 commands, such as a command that “[w]aits for the specified UI element to appear on the screen” or
 27 to “[w]aits for the value of specified UI element attribute to be equal to a string” and, in order to
 28 perform such a verification, the UiPath Platform must necessarily store facts about the webpage being

1 rendered, i.e., the expected condition to be checked for during verification
 2 (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>;
 3 <https://docs.uipath.com/activities/docs/wait-attribute>; [https://docs.uipath.com/activities/docs/execute-](https://docs.uipath.com/activities/docs/execute-x-path)
 4 [x-path](https://docs.uipath.com/activities/docs/execute-x-path)); the UiPath Platform locates these pages elements based on their DOM indexes, which
 5 necessarily requires it to use the DOM access methods included in Dynamic Linked Libraries
 6 associated with a browser code library ([https://docs.uipath.com/activities/docs/wait-ui-element-](https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc)
 7 [appear#misc](https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc); <https://docs.uipath.com/activities/docs/ui-element-exists>;
 8 <https://docs.uipath.com/activities/docs/on-ui-element-appear>); and the UiPath Platform allows for the
 9 testing of content dynamically generated by AJAX programming including using, upon information
 10 and belief, its various wait commands to synchronize playback and allow for testing of content
 11 dynamically generated by AJAX programming ([https://docs.uipath.com/activities/docs/wait-ui-](https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc)
 12 [element-appear#misc](https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc); <https://docs.uipath.com/activities/docs/wait-attribute>;
 13 <https://www.uipath.com/solutions/technology/web-scraping-software>;
 14 <https://www.uipath.com/solutions/technology/web-automation>), as disclosed in the '890 Patent.

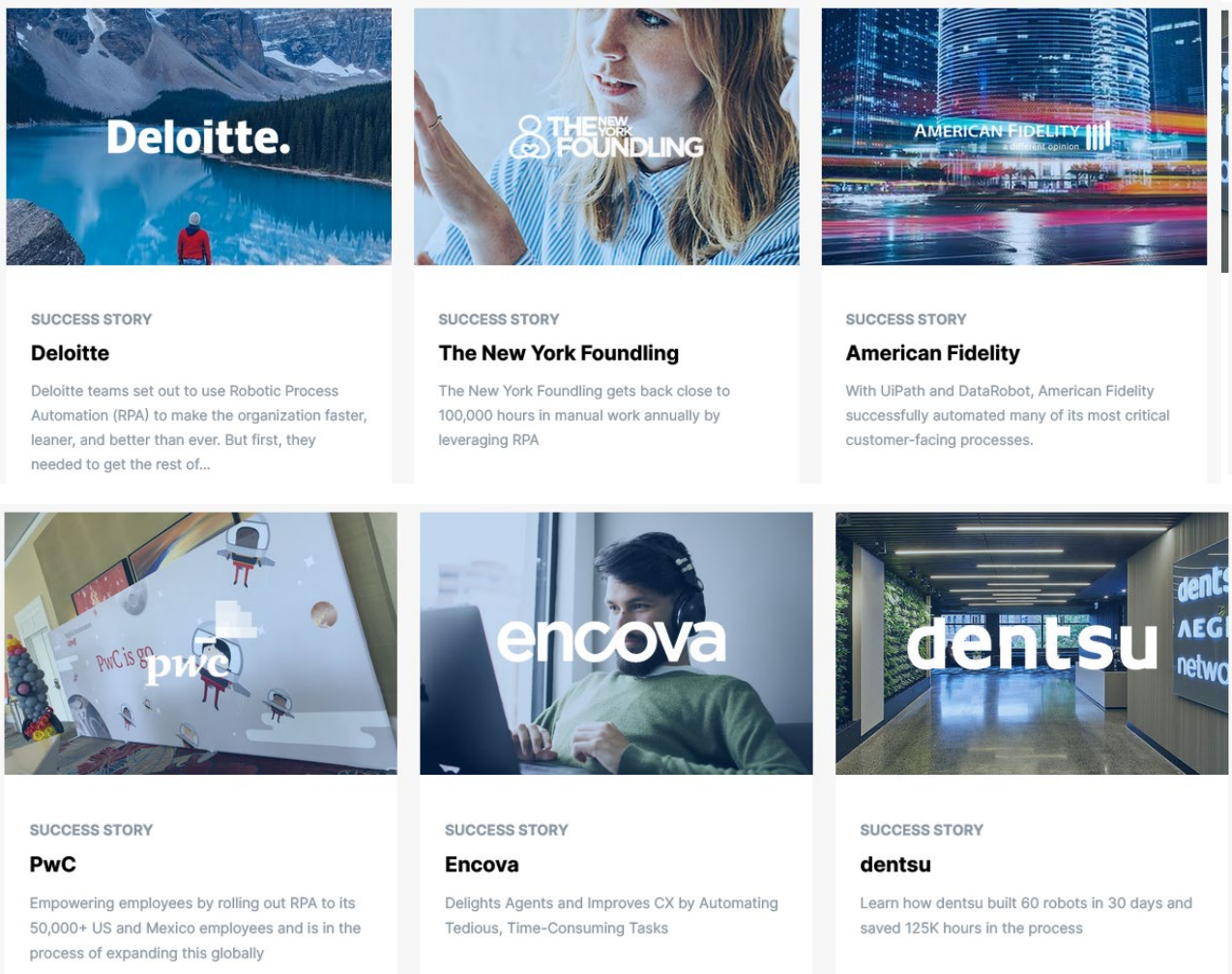
15 69. Defendants will, on information and belief, continue to directly infringe the '890
 16 Patent unless enjoined.

17 70. To the extent Defendants' Infringing Products, without more, do not directly infringe
 18 at least claim 1 of the '890 Patent, at least as of the filing of this Complaint, Defendants contribute to
 19 infringement of the same under 35 U.S.C. § 271(c) inasmuch as the Infringing Products offered for
 20 sale and sold by Defendants are each a component of a patented machine or an apparatus used in
 21 practicing a patented process, constituting a material part of SRI's invention, knowing the same to be
 22 especially made or especially adapted for use in infringement of the '890 Patent. For example, the
 23 UiPath Platform, when used in its normal and intended usage (pursuant to the instructions set forth on
 24 Defendants' website) infringes claim 1 of the '890 Patent. *See supra*, ¶ 68.

25 71. Defendants will, on information and belief, continue to contribute to infringement of
 26 the '890 Patent unless enjoined.

27 72. Defendants actively encourage their customer to use Defendants' Infringing Products
 28 in an infringing manner. For example, Defendants' website is replete with written directions,

1 screenshots, and videos instructing users on how to use the Infringing Products in an infringing
 2 manner. For example, as set forth above, Defendants' website regarding the UiPath Platform
 3 specifically instructs users of the Infringing Products how to infringe claim 1 of the '890 patent. *See*
 4 *supra*, ¶ 68. Defendants' website also touts the identities of customers who use the Infringing
 5 Products, each of whom is a direct infringer inasmuch as they use the Infringing Products in the
 6 infringing manner as instructed by Defendants:



14 Upon information and belief, and particularly by way of the detailed documentation instructing users
 15 on how to use the Infringing Products in an infringing manner (*see supra*, ¶¶ 68, 72), Defendants
 16 have encouraged this infringement with knowledge of the '890 Patent and with a specific intent to
 17 cause their customers and distributors to infringe.

73. Defendants' acts thus constitute active inducement of patent infringement in violation of 35 U.S.C. § 271(b).

74. Defendants will, on information and belief, continue to induce infringement of the '890 Patent unless enjoined.

75. Defendants' direct infringement, contributory infringement, and inducement of infringement have irreparably harmed SRI.

76. Defendants will, on information and belief, continue to irreparably harm SRI unless enjoined.

77. Pursuant to 35 U.S.C. § 284, SRI is entitled to damages adequate to compensate for the infringement but in no event less than a reasonable royalty.

78. This case is "exceptional" within the meaning of 35 U.S.C. § 285, and SRI is entitled to an award of attorneys' fees.

COUNT IV – INFRINGEMENT OF THE '585 PATENT

79. SRI re-alleges and incorporates the allegations of the preceding paragraphs of this Complaint as if fully set forth herein.

80. SRI is the assignee and owner of all right, title, and interest in and to the '585 Patent, which was issued on July 23, 2013. A true and correct copy of the '585 Patent is attached hereto as Exhibit D.

81. The '585 Patent addresses an invention for testing websites. The disclosed innovation tests many facets of the website's experience and operation, including by providing novel approaches to creating, storing, and executing test scripts capable of accurately testing AJAX webpage elements.

82. SRI has the exclusive right to make, use, sell, and offer to sell any product embodying the '585 Patent throughout the United States, and to import any product embodying the '585 Patent into the United States.

83. SRI has commercially exploited the '585 Patent by making, marketing, selling, and using products covered by the '585 Patent, including its popular eValid™ software products. SRI continues to commercially exploit the '585 Patent through the present, at least by continuing to provide maintenance and support to users of its popular eValid™ software products.

1 84. Defendants have had knowledge of the '585 Patent, SRI, and SRI's products
2 embodying the inventions claimed in the Patents-in-Suit since at least as early as the filing of this
3 Complaint.

4 85. At all relevant times, SRI provided public notice of the '585 Patent by properly
5 marking its products and its website pursuant to 35 U.S.C. § 287(a).

6 86. Defendants have been, and are currently, directly infringing at least claim 1 of the
7 '585 Patent in violation of 35 U.S.C. § 271(a), literally or under the doctrine of equivalents, by
8 making, using, selling, offering for sale, and/or importing into the United States Defendants'
9 Infringing Products, which, as set forth in documentation available on Defendants' website, comprise
10 the non-transitory computer readable media disclosed in the '585 Patent—both as maintained in
11 Defendants' files and as made accessible to its users to whom Defendants offer and sell the Infringing
12 Products—including at least computer program code for providing a test enabled web browser, said
13 medium comprising computer program code for providing web browsing capabilities (for example,
14 “In the UiPath Platform, the WebDriver protocol introduces support for headless browser
15 automations. ... Although the WebDriver protocol can be used for both headless and visual browser
16 automations, it is not meant to replace the UiPath browser extensions, giving you the freedom to use
17 whichever option suits you best.” ([https://docs.uipath.com/studio/docs/about-the-webdriver-
18 protocol](https://docs.uipath.com/studio/docs/about-the-webdriver-protocol))); “UiPath Web Automation uses a built-in recorder that can read and enact web-based
19 activities. It identifies web elements by their attributes and accurately manipulates them while
20 keeping up with website changes.” (<https://www.uipath.com/solutions/technology/web-automation>);
21 “Web Automation ... The example below explains how to automate the action of browsing a
22 webpage, retrieving the needed information, and displaying it into a message box.”
23 (<https://docs.uipath.com/activities/docs/web-automation>)); the UiPath Platform as used with a web
24 browser, such as Chrome or Firefox, is a test-enabled web browser (for example, “UiPath Extensions
25 ... Chrome: Installs the browser extension for automating websites in Chrome. ... Firefox: Installs
26 the browser extension for automating websites in Firefox.” (<https://docs.uipath.com/studio/docs>);
27 “Web Automation ... The example below explains how to automate the action of browsing a
28 webpage, retrieving the needed information, and displaying it into a message box.”

1 (<https://docs.uipath.com/activities/docs/web-automation>)); and allows a user to browse the web via
 2 common web browsing activities, such as navigating to a web page, clicking and double clicking a
 3 mouse, or selecting an item from a drop-down list or combo box
 4 (<https://docs.uipath.com/activities/docs/navigate-to>;
 5 <https://docs.uipath.com/studio/v2018.3/docs/manual-recording>;
 6 <https://docs.uipath.com/activities/docs/double-click>); computer program code for testing capabilities
 7 of a website hosted by a server and accessible to a computer via a network wherein the computer
 8 program code for testing capabilities of the website includes computer program code configured to
 9 receive a synchronization check from a user using the test enabled web browser, to insert the
 10 synchronization check into a test script for testing at least one webpage of the website (for example,
 11 the UiPath Platform allows a user to create or record a test using a browser and then play it back by
 12 executing it (“Recording can help you save a lot of time when automating your tasks. The App/Web
 13 Recorder captures your actions as you perform them on the screen and generates a Use
 14 Application/Browser activity with a series of activities inside it based on your actions.”
 15 (<https://docs.uipath.com/activities/docs/app-web-recorder>); “There are four types of recordings
 16 available in UiPath Studio: ... Web – designed for recording in web apps and browsers, generates
 17 containers and uses the Simulate Type/Click input method by default.”
 18 (<https://docs.uipath.com/studio/v2018.3/docs/about-recording-types>); “Executing tests can be done
 19 either manually or automatically. Manual execution is invoked via Test Manager while automated
 20 execution is invoked via UiPath Orchestrator only.” ([https://docs.uipath.com/test-](https://docs.uipath.com/test-suite/docs/executing-tests)
 21 [suite/docs/executing-tests](https://docs.uipath.com/test-suite/docs/executing-tests))); the UiPath Platform interrogates the DOM to identify and extract
 22 relevant information regarding at least the page elements germane to the test, including each such
 23 element’s value, and stores those details in the test ([https://docs.uipath.com/activities/docs/get-](https://docs.uipath.com/activities/docs/get-attribute)
 24 [attribute](https://docs.uipath.com/activities/docs/get-attribute); <https://docs.uipath.com/activities/lang-en/docs/get-value>;
 25 <https://docs.uipath.com/activities/docs/extract-data-from-website>;
 26 <https://docs.uipath.com/activities/docs/digitize-document>); the UiPath Platform uses explicit wait
 27 commands, such as a command that “[w]aits for the specified UI element to appear on the screen” or
 28 to “[w]aits for the value of specified UI element attribute to be equal to a string” and, in order to

1 perform such a verification, the UiPath Platform must necessarily store facts about the webpage being
 2 rendered, i.e., the expected condition to be checked for during verification
 3 (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>;
 4 <https://docs.uipath.com/activities/docs/wait-attribute>; [https://docs.uipath.com/activities/docs/execute-
 5 x-path](https://docs.uipath.com/activities/docs/execute-x-path)); the UiPath Platform locates these pages elements based on their DOM indexes, which
 6 necessarily requires it to use the DOM access methods included in Dynamic Linked Libraries
 7 associated with a browser code library ([https://docs.uipath.com/activities/docs/wait-ui-element-
 8 appear#misc](https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc); <https://docs.uipath.com/activities/docs/ui-element-exists>;
 9 <https://docs.uipath.com/activities/docs/on-ui-element-appear>); the UiPath Platform allows for the
 10 testing of content dynamically generated by AJAX programming including using, upon information
 11 and belief, its various wait commands to synchronize playback and allow for testing of content
 12 dynamically generated by AJAX programming ([https://docs.uipath.com/activities/docs/wait-ui-
 13 element-appear#misc](https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc); <https://docs.uipath.com/activities/docs/wait-attribute>;
 14 <https://www.uipath.com/solutions/technology/web-scraping-software>;
 15 <https://www.uipath.com/solutions/technology/web-automation>); the test script being separate from
 16 the at least one webpage being tested (for example, the UiPath Platform stores and accesses test
 17 scripts separately from the webpage itself ([https://www.uipath.com/solutions/technology/web-
 18 automation](https://www.uipath.com/solutions/technology/web-automation); <https://docs.uipath.com/studio/docs/about-the-webdriver-protocol>;
 19 <https://docs.uipath.com/studio/lang-ru/docs/creating-basic-process>)), the at least one webpage being
 20 tested including AJAX programming, and to automatically synchronize playback of the test script
 21 using at least the synchronization check to maintain the test enabled browser's state with respect to
 22 the AJAX programming by means of the synchronization check in the test script to a DOM
 23 associated with the website (for example, the UiPath Platform allows a user to create or record a test
 24 using a browser and then play it back by executing it ("Recording can help you save a lot of time
 25 when automating your tasks. The App/Web Recorder captures your actions as you perform them on
 26 the screen and generates a Use Application/Browser activity with a series of activities inside it based
 27 on your actions." (<https://docs.uipath.com/activities/docs/app-web-recorder>); "There are four types of
 28 recordings available in UiPath Studio: ... Web – designed for recording in web apps and browsers,

generates containers and uses the Simulate Type/Click input method by default.”

(<https://docs.uipath.com/studio/v2018.3/docs/about-recording-types>); “Executing tests can be done either manually or automatically. Manual execution is invoked via Test Manager while automated execution is invoked via UiPath Orchestrator only.” (<https://docs.uipath.com/test-suite/docs/executing-tests>)); the UiPath Platform interrogates the DOM to identify and extract relevant information regarding at least the page elements germane to the test, including each such element’s value, and stores those details in the test (<https://docs.uipath.com/activities/docs/get-attribute>; <https://docs.uipath.com/activities/lang-en/docs/get-value>; <https://docs.uipath.com/activities/docs/extract-data-from-website>; <https://docs.uipath.com/activities/docs/digitize-document>); the UiPath Platform uses explicit wait commands, such as a command that “[w]aits for the specified UI element to appear on the screen” or to “[w]aits for the value of specified UI element attribute to be equal to a string” and, in order to perform such a verification, the UiPath Platform must necessarily store facts about the webpage being rendered, i.e., the expected condition to be checked for during verification

(<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/wait-attribute>; <https://docs.uipath.com/activities/docs/execute-x-path>); the UiPath Platform locates these pages elements based on their DOM indexes, which necessarily requires it to use the DOM access methods included in Dynamic Linked Libraries associated with a browser code library (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/ui-element-exists>; <https://docs.uipath.com/activities/docs/on-ui-element-appear>); the UiPath Platform allows for the testing of content dynamically generated by AJAX programming including using, upon information and belief, its various wait commands to synchronize playback and allow for testing of content dynamically generated by AJAX programming (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/wait-attribute>; <https://www.uipath.com/solutions/technology/web-scraping-software>; <https://www.uipath.com/solutions/technology/web-automation>); wherein the synchronization check in the test script and web browsing activities provided by the web browsing capabilities are able to

separately access the DOM associated with the at least one webpage of the website (for example, the UiPath Platform stores and accesses test scripts separately from the webpage itself (<https://www.uipath.com/solutions/technology/web-automation>; <https://docs.uipath.com/studio/docs/about-the-webdriver-protocol>; <https://docs.uipath.com/studio/lang-ru/docs/creating-basic-process>)), and wherein the synchronization check is inserted into the test script as at least one command, and the at least one command operates, when executed, to find a current index of at least one DOM element of the at least one webpage based on a specified property name and/or property value, and (i) submit a named event to the at least one DOM element of the at least one webpage having the current index, or (ii) insert or verify a value in the at least one DOM element of the at least one webpage having the current index (for example, the UiPath Platform allows a user to create or record a test using a browser and then play it back by executing it (“Recording can help you save a lot of time when automating your tasks. The App/Web Recorder captures your actions as you perform them on the screen and generates a Use Application/Browser activity with a series of activities inside it based on your actions.” (<https://docs.uipath.com/activities/docs/app-web-recorder>); “There are four types of recordings available in UiPath Studio: ... Web – designed for recording in web apps and browsers, generates containers and uses the Simulate Type/Click input method by default.” (<https://docs.uipath.com/studio/v2018.3/docs/about-recording-types>); “Executing tests can be done either manually or automatically. Manual execution is invoked via Test Manager while automated execution is invoked via UiPath Orchestrator only.” (<https://docs.uipath.com/test-suite/docs/executing-tests>))); the UiPath Platform interrogates the DOM to identify and extract relevant information regarding at least the page elements germane to the test, including each such element’s value, and stores those details in the test (<https://docs.uipath.com/activities/docs/get-attribute>; <https://docs.uipath.com/activities/lang-en/docs/get-value>; <https://docs.uipath.com/activities/docs/extract-data-from-website>; <https://docs.uipath.com/activities/docs/digitize-document>); the UiPath Platform uses explicit wait commands, such as a command that “[w]aits for the specified UI element to appear on the screen” or to “[w]aits for the value of specified UI element attribute to be equal to a string” and, in order to

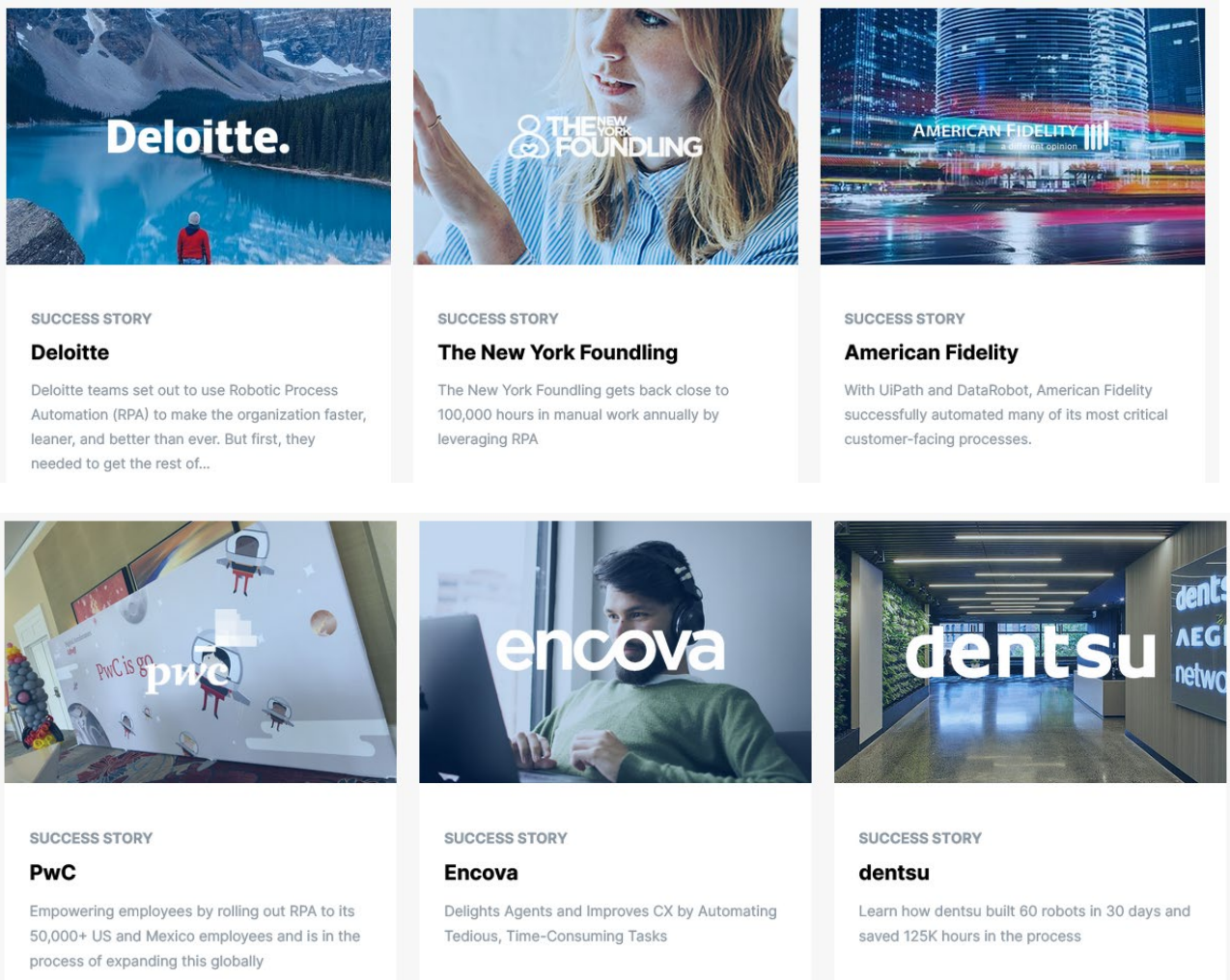
perform such a verification, the UiPath Platform must necessarily store facts about the webpage being rendered, i.e., the expected condition to be checked for during verification (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/wait-attribute>; <https://docs.uipath.com/activities/docs/execute-x-path>); the UiPath Platform locates these pages elements based on their DOM indexes, which necessarily requires it to use the DOM access methods included in Dynamic Linked Libraries associated with a browser code library (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/ui-element-exists>; <https://docs.uipath.com/activities/docs/on-ui-element-appear>); and the UiPath Platform allows for the testing of content dynamically generated by AJAX programming including using, upon information and belief, its various wait commands to synchronize playback and allow for testing of content dynamically generated by AJAX programming (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/wait-attribute>; <https://www.uipath.com/solutions/technology/web-scraping-software>; <https://www.uipath.com/solutions/technology/web-automation>), as disclosed in the '585 Patent.

87. Defendants will, on information and belief, continue to directly infringe the '585 Patent unless enjoined.

88. To the extent Defendants' Infringing Products, without more, do not directly infringe at least claim 1 of the '585 Patent, at least as of the filing of this Complaint, Defendants contribute to infringement of the same under 35 U.S.C. § 271(c) inasmuch as the Infringing Products offered for sale and sold by Defendants are each a component of a patented machine or an apparatus used in practicing a patented process, constituting a material part of SRI's invention, knowing the same to be especially made or especially adapted for use in infringement of the '585 Patent. For example, as set forth above, the UiPath Platform, when used in its normal and intended usage (pursuant to the instructions set forth on Defendants' website) infringes claim 1 of the '585 Patent. *See supra*, ¶ 86.

89. Defendants will, on information and belief, continue to contribute to infringement of the '585 Patent unless enjoined.

90. Defendants actively encourage their customer to use Defendants' Infringing Products in an infringing manner. For example, Defendants' website is replete with written directions, screenshots, and videos instructing users on how to use the Infringing Products in an infringing manner. For example, as set forth above, Defendants' website regarding the UiPath Platform specifically instructs users of the Infringing Products how to infringe claim 1 of the '585 patent. *See supra*, ¶ 86. Defendants' website also touts the identities of customers who use the Infringing Products, each of whom is a direct infringer inasmuch as they use the Infringing Products in the infringing manner as instructed by Defendants:



91. Upon information and belief, and particularly by way of the detailed documentation instructing users on how to use the Infringing Products in an infringing manner (*see supra*, ¶¶ 86, 90),

1 Defendants have encouraged this infringement with knowledge of the '585 Patent and with a specific
2 intent to cause their customers and distributors to infringe.

3 92. Defendants' acts thus constitute active inducement of patent infringement in violation
4 of 35 U.S.C. § 271(b).

5 93. Defendants will, on information and belief, continue to induce infringement of the
6 '585 Patent unless enjoined.

7 94. Defendants' direct infringement, contributory infringement, and inducement of
8 infringement have irreparably harmed SRI.

9 95. Defendants will, on information and belief, continue to irreparably harm SRI unless
10 enjoined.

11 96. Pursuant to 35 U.S.C. § 284, SRI is entitled to damages adequate to compensate for
12 the infringement but in no event less than a reasonable royalty.

13 97. This case is "exceptional" within the meaning of 35 U.S.C. § 285, and SRI is entitled
14 to an award of attorneys' fees.

15 **COUNT V – INFRINGEMENT OF THE '493 PATENT**

16 98. SRI re-alleges and incorporates the allegations of the preceding paragraphs of this
17 Complaint as if fully set forth herein.

18 99. SRI is the assignee and owner of all right, title, and interest in and to the '493 Patent,
19 which was issued on February 11, 2014. A true and correct copy of the '493 Patent is attached hereto
20 as Exhibit E.

21 100. The '493 Patent addresses an invention for testing websites. The disclosed innovation
22 tests many facets of the website's experience and operation, including by providing novel approaches
23 to creating, storing, and executing test scripts using website elements as opposed to the previously
24 disclosed use of recording test scripts based upon user actions only.

25 101. SRI has the exclusive right to make, use, sell, and offer to sell any product embodying
26 the '493 Patent throughout the United States, and to import any product embodying the '493 Patent
27 into the United States.

1 102. SRI has commercially exploited the '493 Patent by making, marketing, selling, and
2 using products covered by the '493 Patent, including its popular eValid™ software products. SRI
3 continues to commercially exploit the '493 Patent through the present, at least by continuing to
4 provide maintenance and support to users of its popular eValid™ software products.

5 103. Defendants have had knowledge of the '493 Patent, SRI, and SRI's products
6 embodying the inventions claimed in the Patents-in-Suit since at least as early as the filing of this
7 Complaint.

8 104. At all relevant times, SRI provided public notice of the '493 Patent by properly
9 marking its products and its website under 35 U.S.C. § 287(a).

10 105. Defendants have been, and are currently, directly infringing at least claim 1 of the
11 '493 Patent in violation of 35 U.S.C. § 271(a), literally or under the doctrine of equivalents, by
12 making, using, selling, offering for sale, and/or importing into the United States Defendants'
13 Infringing Products, which, as set forth in documentation available on Defendants' website, comprise
14 the non-transitory computer readable media disclosed in the '493 Patent—both as maintained in
15 Defendants' files and as made accessible to its users to whom Defendants offer and sell the Infringing
16 Products—including at least computer program code stored therein for providing a test-enabled
17 browser for testing a website residing on a network (for example, the UiPath Platform can be used for
18 “[e]asy web testing” and to “[c]reate reliable tests without programming knowledge.”
19 (<https://www.uipath.com/solutions/technology/web-automation>); “In the UiPath Platform, the
20 WebDriver protocol introduces support for headless browser automations. ... Although the
21 WebDriver protocol can be used for both headless and visual browser automations, it is not meant to
22 replace the UiPath browser extensions, giving you the freedom to use whichever option suits you
23 best.” (<https://docs.uipath.com/studio/docs/about-the-webdriver-protocol>); “UiPath Web Automation
24 uses a built-in recorder that can read and enact web-based activities. It identifies web elements by
25 their attributes and accurately manipulates them while keeping up with website changes.”
26 (<https://www.uipath.com/solutions/technology/web-automation>); the UiPath Platform as used with a
27 web browser, such as Chrome or Firefox, is a test-enabled web browser (for example, “UiPath
28 Extensions ... Chrome: Installs the browser extension for automating websites in Chrome. ...

Firefox: Installs the browser extension for automating websites in Firefox.”

(<https://docs.uipath.com/studio/docs>); “Web Automation ... The example below explains how to automate the action of browsing a webpage, retrieving the needed information, and displaying it into a message box.” (<https://docs.uipath.com/activities/docs/web-automation>)); the website, necessarily including at least one webpage, necessarily resides on a remote server and the UiPath Platform as used with a web browser, such as Chrome or Firefox, is a “test-enabled web browser”

(<https://docs.uipath.com/studio/docs>; <https://docs.uipath.com/activities/docs/web-automation>); said medium comprising computer program code for interfacing with web browsing components, the web browsing components including DOM access methods, computer program code for accessing a website to be tested (for example, the UiPath Platform allows a user to browse the web via common web browsing activities, such as navigating to a web page, clicking and double clicking a mouse, or selecting an item from a drop-down list or combo box

(<https://docs.uipath.com/activities/docs/navigate-to>; <https://docs.uipath.com/studio/v2018.3/docs/manual-recording>; <https://docs.uipath.com/activities/docs/double-click>)); the UiPath Platform interrogates the DOM to identify and extract relevant information regarding at least the page elements germane to the test, including each such element’s value, and stores those details in the test

(<https://docs.uipath.com/activities/docs/get-attribute>; <https://docs.uipath.com/activities/lang-en/docs/get-value>; <https://docs.uipath.com/activities/docs/extract-data-from-website>; <https://docs.uipath.com/activities/docs/digitize-document>); the UiPath Platform locates these pages elements based on their DOM indexes, which necessarily requires it to use the DOM access methods included in Dynamic Linked Libraries associated with a browser code library

(<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/ui-element-exists>; <https://docs.uipath.com/activities/docs/on-ui-element-appear>); computer program code for rendering and examining at least one webpage of the website so as to extract details of elements of the webpage, and store the details of the webpage in a recorded script, such as recorded scripts generated through the testing component of the Infringing Products (for example, the UiPath Platform allows a user to create or record a test using a browser

1 and then play it back by executing it (“Recording can help you save a lot of time when automating
 2 your tasks. The App/Web Recorder captures your actions as you perform them on the screen and
 3 generates a Use Application/Browser activity with a series of activities inside it based on your
 4 actions.” (<https://docs.uipath.com/activities/docs/app-web-recorder>); “There are four types of
 5 recordings available in UiPath Studio: ... Web – designed for recording in web apps and browsers,
 6 generates containers and uses the Simulate Type/Click input method by default.”
 7 (<https://docs.uipath.com/studio/v2018.3/docs/about-recording-types>); “Executing tests can be done
 8 either manually or automatically. Manual execution is invoked via Test Manager while automated
 9 execution is invoked via UiPath Orchestrator only.” ([https://docs.uipath.com/test-](https://docs.uipath.com/test-suite/docs/executing-tests)
 10 [suite/docs/executing-tests](https://docs.uipath.com/test-suite/docs/executing-tests))); the UiPath Platform interrogates the DOM to identify and extract
 11 relevant information regarding at least the page elements germane to the test, including each such
 12 element’s value, and stores those details in the test ([https://docs.uipath.com/activities/docs/get-](https://docs.uipath.com/activities/docs/get-attribute)
 13 [attribute](https://docs.uipath.com/activities/docs/get-attribute); <https://docs.uipath.com/activities/lang-en/docs/get-value>;
 14 <https://docs.uipath.com/activities/docs/extract-data-from-website>;
 15 <https://docs.uipath.com/activities/docs/digitize-document>); the UiPath Platform uses explicit wait
 16 commands, such as a command that “[w]aits for the specified UI element to appear on the screen” or
 17 to “[w]aits for the value of specified UI element attribute to be equal to a string” and, in order to
 18 perform such a verification, the UiPath Platform must necessarily store facts about the webpage being
 19 rendered, i.e., the expected condition to be checked for during verification
 20 (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>;
 21 <https://docs.uipath.com/activities/docs/wait-attribute>; [https://docs.uipath.com/activities/docs/execute-](https://docs.uipath.com/activities/docs/execute-x-path)
 22 [x-path](https://docs.uipath.com/activities/docs/execute-x-path)); the UiPath Platform locates these pages elements based on their DOM indexes, which
 23 necessarily requires it to use the DOM access methods included in Dynamic Linked Libraries
 24 associated with a browser code library ([https://docs.uipath.com/activities/docs/wait-ui-element-](https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc)
 25 [appear#misc](https://docs.uipath.com/activities/docs/ui-element-exists); <https://docs.uipath.com/activities/docs/ui-element-exists>;
 26 <https://docs.uipath.com/activities/docs/on-ui-element-appear>); computer program code for selecting a
 27 validation test to be performed (for example, the UiPath Platform allows a user to create or record a
 28 test using a browser and then play it back by executing it (“Recording can help you save a lot of time

1 when automating your tasks. The App/Web Recorder captures your actions as you perform them on
 2 the screen and generates a Use Application/Browser activity with a series of activities inside it based
 3 on your actions.” (<https://docs.uipath.com/activities/docs/app-web-recorder>); “There are four types of
 4 recordings available in UiPath Studio: ... Web – designed for recording in web apps and browsers,
 5 generates containers and uses the Simulate Type/Click input method by default.”
 6 (<https://docs.uipath.com/studio/v2018.3/docs/about-recording-types>); “Executing tests can be done
 7 either manually or automatically. Manual execution is invoked via Test Manager while automated
 8 execution is invoked via UiPath Orchestrator only.” ([https://docs.uipath.com/test-](https://docs.uipath.com/test-suite/docs/executing-tests)
 9 [suite/docs/executing-tests](https://docs.uipath.com/test-suite/docs/executing-tests))); and the UiPath Platform uses explicit wait commands, such as a
 10 command that “[w]aits for the specified UI element to appear on the screen” or to “[w]aits for the
 11 value of specified UI element attribute to be equal to a string” and, in order to perform such a
 12 verification, the UiPath Platform must necessarily store facts about the webpage being rendered, i.e.,
 13 the expected condition to be checked for during verification
 14 (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>;
 15 <https://docs.uipath.com/activities/docs/wait-attribute>; [https://docs.uipath.com/activities/docs/execute-](https://docs.uipath.com/activities/docs/execute-x-path)
 16 [x-path](https://docs.uipath.com/activities/docs/execute-x-path))); and computer program code for performing the validation test using at least one of the
 17 DOM access methods of the web browsing components, wherein during the validation test, the at
 18 least one webpage is newly rendered and details of elements for the at least one webpage as newly
 19 rendered are accessed via the at least one of the DOM access methods and compared to the stored
 20 details in the recorded script (for example, the UiPath Platform allows a user to create or record a test
 21 using a browser and then play it back by executing it (“Recording can help you save a lot of time
 22 when automating your tasks. The App/Web Recorder captures your actions as you perform them on
 23 the screen and generates a Use Application/Browser activity with a series of activities inside it based
 24 on your actions.” (<https://docs.uipath.com/activities/docs/app-web-recorder>); “There are four types of
 25 recordings available in UiPath Studio: ... Web – designed for recording in web apps and browsers,
 26 generates containers and uses the Simulate Type/Click input method by default.”
 27 (<https://docs.uipath.com/studio/v2018.3/docs/about-recording-types>); “Executing tests can be done
 28 either manually or automatically. Manual execution is invoked via Test Manager while automated




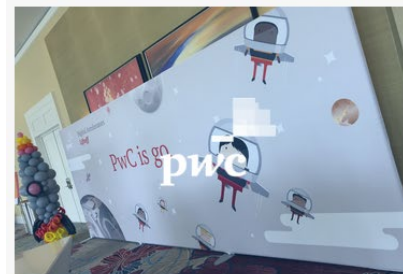
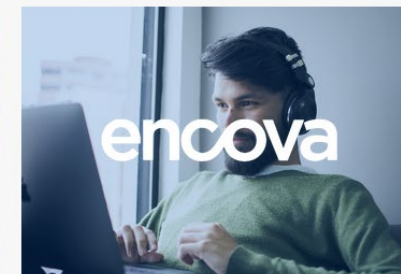

execution is invoked via UiPath Orchestrator only.” (<https://docs.uipath.com/test-suite/docs/executing-tests>)); the UiPath Platform interrogates the DOM to identify and extract relevant information regarding at least the page elements germane to the test, including each such element’s value, and stores those details in the test (<https://docs.uipath.com/activities/docs/get-attribute>; <https://docs.uipath.com/activities/lang-en/docs/get-value>; <https://docs.uipath.com/activities/docs/extract-data-from-website>; <https://docs.uipath.com/activities/docs/digitize-document>); the UiPath Platform uses explicit wait commands, such as a command that “[w]aits for the specified UI element to appear on the screen” or to “[w]aits for the value of specified UI element attribute to be equal to a string” and, in order to perform such a verification, the UiPath Platform must necessarily store facts about the webpage being rendered, i.e., the expected condition to be checked for during verification (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/wait-attribute>; <https://docs.uipath.com/activities/docs/execute-x-path>); and the UiPath Platform locates these pages elements based on their DOM indexes, which necessarily requires it to use the DOM access methods included in Dynamic Linked Libraries associated with a browser code library (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/ui-element-exists>; <https://docs.uipath.com/activities/docs/on-ui-element-appear>); as disclosed in the ’493 Patent.

106. Defendants will, on information and belief, continue to directly infringe the ’493 Patent unless enjoined.

107. To the extent Defendants’ Infringing Products, without more, do not directly infringe at least claim 1 of the ’493 Patent, at least as of the filing of this Complaint, Defendants contribute to infringement of the same under 35 U.S.C. § 271(c) inasmuch as the Infringing Products offered for sale and sold by Defendants are each a component of a patented machine or an apparatus used in practicing a patented process, constituting a material part of SRI’s invention, knowing the same to be especially made or especially adapted for use in infringement of the ’493 Patent. For example, as set forth above, the UiPath Platform, when used in its normal and intended usage (pursuant to the instructions set forth on Defendants’ website) infringes claim 1 of the ’493 Patent. *See supra*, ¶ 105.

108. Defendants will, on information and belief, continue to contribute to infringement of the '493 Patent unless enjoined.

109. Defendants actively encourage their customer to use Defendants' Infringing Products in an infringing manner. For example, Defendants' website is replete with written directions, screenshots, and videos instructing users on how to use the Infringing Products in an infringing manner. For example, as set forth above, Defendants' website regarding the UiPath Platform specifically instructs users of the Infringing Products how to infringe claim 1 of the '493 patent. *See supra*, ¶ 105. Defendants' website also touts the identities of customers who use the Infringing Products, each of whom is a direct infringer inasmuch as they use the Infringing Products in the infringing manner as instructed by Defendants:

 <p>SUCCESS STORY</p> <p>Deloitte</p> <p>Deloitte teams set out to use Robotic Process Automation (RPA) to make the organization faster, leaner, and better than ever. But first, they needed to get the rest of...</p>	 <p>SUCCESS STORY</p> <p>The New York Foundling</p> <p>The New York Foundling gets back close to 100,000 hours in manual work annually by leveraging RPA</p>	 <p>SUCCESS STORY</p> <p>American Fidelity</p> <p>With UiPath and DataRobot, American Fidelity successfully automated many of its most critical customer-facing processes.</p>
 <p>SUCCESS STORY</p> <p>PwC</p> <p>Empowering employees by rolling out RPA to its 50,000+ US and Mexico employees and is in the process of expanding this globally</p>	 <p>SUCCESS STORY</p> <p>Encova</p> <p>Delights Agents and Improves CX by Automating Tedious, Time-Consuming Tasks</p>	 <p>SUCCESS STORY</p> <p>dentsu</p> <p>Learn how dentsu built 60 robots in 30 days and saved 125K hours in the process</p>

110. Upon information and belief, and particularly by way of the detailed documentation instructing users on how to use the Infringing Products in an infringing manner (*see supra*, ¶¶ 105, 109), Defendants have encouraged this infringement with knowledge of the '493 Patent and with a specific intent to cause their customers and distributors to infringe.

111. Defendants' acts thus constitute active inducement of patent infringement in violation of 35 U.S.C. § 271(b).

112. Defendants will, on information and belief, continue to induce infringement of the '493 Patent unless enjoined.

113. Defendants' direct infringement, contributory infringement, and inducement of infringement have irreparably harmed SRI.

114. Defendants will, on information and belief, continue to irreparably harm SRI unless enjoined.

115. Pursuant to 35 U.S.C. § 284, SRI is entitled to damages adequate to compensate for the infringement but in no event less than a reasonable royalty.

116. This case is "exceptional" within the meaning of 35 U.S.C. § 285, and SRI is entitled to an award of attorneys' fees.

COUNT VI – INFRINGEMENT OF THE '491 PATENT

117. SRI re-alleges and incorporates the allegations of the preceding paragraphs of this Complaint as if fully set forth herein.

118. SRI is the assignee and owner of all right, title, and interest in and to the '491 Patent, which was issued on March 17, 2015. A true and correct copy of the '491 Patent is attached hereto as Exhibit F.

119. The '491 Patent addresses an invention for testing websites. The disclosed innovation tests many facets of the website's experience and operation, including by providing novel approaches to creating, storing, and executing test scripts using website elements as opposed to the previously disclosed use of recording test scripts based upon user actions only.

//

//

120. SRI has the exclusive right to make, use, sell, and offer to sell any product embodying the '491 Patent throughout the United States, and to import any product embodying the '491 Patent into the United States.

121. SRI has commercially exploited the '491 Patent by making, marketing, selling, and using products covered by the '491 Patent, including its popular eValid™ software products. SRI continues to commercially exploit the '491 Patent through the present, at least by continuing to provide maintenance and support to users of its popular eValid™ software products.

122. Defendants have had knowledge of the '491 Patent, SRI, and SRI's products embodying the inventions claimed in the Patents-in-Suit since at least as early as the filing of this Complaint.

123. At all relevant times, SRI provided public notice of the '491 Patent by properly marking its products and its website pursuant to 35 U.S.C. § 287(a).

124. Defendants have been, and are currently, directly infringing at least claim 1 of the '491 Patent in violation of 35 U.S.C. § 271(a), literally or under the doctrine of equivalents, by making, using, selling, offering for sale, and/or importing into the United States Defendants' Infringing Products, which, as set forth in documentation available on Defendants' website, comprise the non-transitory computer readable media disclosed in the '491 Patent—both as maintained in Defendants' files and as made accessible to its users to whom Defendants offer and sell the Infringing Products—including at least computer program code for testing capabilities of a website hosted by a server and accessible to a computer via a network (for example, the UiPath Platform can be used for “[e]asy web testing” and to “[c]reate reliable tests without programming knowledge.” (<https://www.uipath.com/solutions/technology/web-automation>); “In the UiPath Platform, the WebDriver protocol introduces support for headless browser automations. ... Although the WebDriver protocol can be used for both headless and visual browser automations, it is not meant to replace the UiPath browser extensions, giving you the freedom to use whichever option suits you best.” (<https://docs.uipath.com/studio/docs/about-the-webdriver-protocol>); “UiPath Web Automation uses a built-in recorder that can read and enact web-based activities. It identifies web elements by their attributes and accurately manipulates them while keeping up with website changes.”

1 (<https://www.uipath.com/solutions/technology/web-automation>)); the UiPath Platform as used with a
 2 web browser, such as Chrome or Firefox, is a “test-enabled web browser” (for example, “UiPath
 3 Extensions ... Chrome: Installs the browser extension for automating websites in Chrome. ...
 4 Firefox: Installs the browser extension for automating websites in Firefox.”
 5 (<https://docs.uipath.com/studio/docs>); “Web Automation ... The example below explains how to
 6 automate the action of browsing a webpage, retrieving the needed information, and displaying it into
 7 a message box.” (<https://docs.uipath.com/activities/docs/web-automation>)); wherein the computer
 8 program code for testing capabilities of the website includes at least computer program code
 9 configured to have a synchronization check in a test script for testing at least one web page of the
 10 website, and to automatically synchronize playback of the test script using at least the
 11 synchronization check to maintain the test enabled browser's state by means of the synchronization
 12 check in the test script to a Document Object Model (DOM) associated with the at least one web page
 13 of the website, (for example, the UiPath Platform allows a user to create or record a test using a
 14 browser and then play it back by executing it (“Recording can help you save a lot of time when
 15 automating your tasks. The App/Web Recorder captures your actions as you perform them on the
 16 screen and generates a Use Application/Browser activity with a series of activities inside it based on
 17 your actions.” (<https://docs.uipath.com/activities/docs/app-web-recorder>); “There are four types of
 18 recordings available in UiPath Studio: ... Web – designed for recording in web apps and browsers,
 19 generates containers and uses the Simulate Type/Click input method by default.”
 20 (<https://docs.uipath.com/studio/v2018.3/docs/about-recording-types>); “Executing tests can be done
 21 either manually or automatically. Manual execution is invoked via Test Manager while automated
 22 execution is invoked via UiPath Orchestrator only.” ([https://docs.uipath.com/test-](https://docs.uipath.com/test-suite/docs/executing-tests)
 23 [suite/docs/executing-tests](https://docs.uipath.com/test-suite/docs/executing-tests))); the UiPath Platform interrogates the DOM to identify and extract
 24 relevant information regarding at least the page elements germane to the test, including each such
 25 element’s value, and stores those details in the test ([https://docs.uipath.com/activities/docs/get-](https://docs.uipath.com/activities/docs/get-attribute)
 26 [attribute](https://docs.uipath.com/activities/docs/get-attribute); <https://docs.uipath.com/activities/docs/get-value>;
 27 <https://docs.uipath.com/activities/docs/extract-data-from-website>;
 28 <https://docs.uipath.com/activities/docs/digitize-document>); the UiPath Platform uses explicit wait

1 commands, such as a command that “[w]aits for the specified UI element to appear on the screen” or
 2 to “[w]aits for the value of specified UI element attribute to be equal to a string” and, in order to
 3 perform such a verification, the UiPath Platform must necessarily store facts about the webpage being
 4 rendered, i.e., the expected condition to be checked for during verification
 5 (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>;
 6 <https://docs.uipath.com/activities/docs/wait-attribute>; [https://docs.uipath.com/activities/docs/execute-
 7 x-path](https://docs.uipath.com/activities/docs/execute-x-path)); the UiPath Platform locates these pages elements based on their DOM indexes, which
 8 necessarily requires it to use the DOM access methods included in Dynamic Linked Libraries
 9 associated with a browser code library ([https://docs.uipath.com/activities/docs/wait-ui-element-
 10 appear#misc](https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc); <https://docs.uipath.com/activities/docs/ui-element-exists>;
 11 <https://docs.uipath.com/activities/docs/on-ui-element-appear>); and the UiPath Platform allows for the
 12 testing of content dynamically generated by AJAX programming including using, upon information
 13 and belief, its various wait commands to synchronize playback and allow for testing of content
 14 dynamically generated by AJAX programming ([https://docs.uipath.com/activities/docs/wait-ui-
 15 element-appear#misc](https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc); <https://docs.uipath.com/activities/docs/wait-attribute>;
 16 <https://www.uipath.com/solutions/technology/web-scraping-software>;
 17 <https://www.uipath.com/solutions/technology/web-automation>); wherein the synchronization check
 18 operates, when executed, to: find a current index of at least one DOM element of the at least one web
 19 page based on a specified property name and/or property value; determine whether a property name
 20 and/or value is present in the at least one DOM element of the at least one web page having the
 21 current index; and after the current index is found and the property name and/or value is determined
 22 to be present, wait for the property name and/ or value in the at least one DOM element of the at least
 23 one web page having the current index to be a particular name and/or value (for example, the UiPath
 24 Platform allows for the creation of test scripts to test websites by recording a user’s interactions with
 25 the webpage in question and allowing the user to play back those test scripts, which are stored and
 26 accessed separately from the webpage itself ([https://www.uipath.com/solutions/technology/web-
 27 automation](https://www.uipath.com/solutions/technology/web-automation); <https://docs.uipath.com/studio/docs/about-the-webdriver-protocol>;
 28 <https://docs.uipath.com/studio/lang-ru/docs/creating-basic-process>)); the UiPath Platform interrogates

the DOM to identify and extract relevant information regarding at least the page elements germane to the test, including each such element's value, and stores those details in the test (<https://docs.uipath.com/activities/docs/get-attribute>; <https://docs.uipath.com/activities/lang-en/docs/get-value>; <https://docs.uipath.com/activities/docs/extract-data-from-website>; <https://docs.uipath.com/activities/docs/digitize-document>); the UiPath Platform uses explicit wait commands, such as a command that "[w]aits for the specified UI element to appear on the screen" or to "[w]aits for the value of specified UI element attribute to be equal to a string" and, in order to perform such a verification, the UiPath Platform must necessarily store facts about the webpage being rendered, i.e., the expected condition to be checked for during verification (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/wait-attribute>; <https://docs.uipath.com/activities/docs/execute-x-path>); the UiPath Platform locates these pages elements based on their DOM indexes, which necessarily requires it to use the DOM access methods included in Dynamic Linked Libraries associated with a browser code library (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/ui-element-exists>; <https://docs.uipath.com/activities/docs/on-ui-element-appear>); the UiPath Platform allows for the testing of content dynamically generated by AJAX programming including using, upon information and belief, its various wait commands to synchronize playback and allow for testing of content dynamically generated by AJAX programming (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/wait-attribute>; <https://www.uipath.com/solutions/technology/web-scraping-software>; <https://www.uipath.com/solutions/technology/web-automation>); wherein the computer program code configured to have the synchronization check is a separate programmatic process from the at least one web page of the website being tested (for example, upon information and belief, the UiPath Platform's synchronization processes run in a separate programmatic process from the web page of the website being tested (which runs in the web browser) (<https://www.uipath.com/solutions/technology/web-automation>;

1 [https://docs.uipath.com/studio/docs/about-the-webdriver-protocol](https://docs.uipath.com/studio/docs/about-the-webdriver-protocol;);

2 <https://docs.uipath.com/studio/lang-ru/docs/creating-basic-process>)), as disclosed in the '491 Patent.

3 125. Defendants will, on information and belief, continue to directly infringe the '491
4 Patent unless enjoined.

5 126. To the extent Defendants' Infringing Products, without more, do not directly infringe
6 at least claim 1 of the '491 Patent, at least as of the filing of this Complaint, Defendants contribute to
7 infringement of the same under 35 U.S.C. § 271(c) inasmuch as the Infringing Products offered for
8 sale and sold by Defendants are each a component of a patented machine or an apparatus used in
9 practicing a patented process, constituting a material part of SRI's invention, knowing the same to be
10 especially made or especially adapted for use in infringement of the '491 Patent. For example, as set
11 forth above, the UiPath Platform, when used in its normal and intended usage (pursuant to the
12 instructions set forth on Defendants' website), infringes claim 1 of the '491 Patent. *See supra*, ¶ 124.

13 127. Defendants will, on information and belief, continue to contribute to infringement of
14 the '491 Patent unless enjoined.

15 128. Defendants actively encourage their customer to use Defendants' Infringing Products
16 in an infringing manner. For example, Defendants' website is replete with written directions,
17 screenshots, and videos instructing users on how to use the Infringing Products in an infringing
18 manner. For example, as set forth above, Defendants' website regarding the UiPath Platform
19 specifically instructs users of the Infringing Products how to infringe claim 1 of the '491 patent. *See*
20 *supra*, ¶ 124. Defendants' website also touts the identities of customers who use the Infringing
21 Products, each of whom is a direct infringer inasmuch as they use the Infringing Products in the
22 infringing manner as instructed by Defendants:

23 //

24 //

25 //

26 //

27 //

28 //



SUCCESS STORY

Deloitte

Deloitte teams set out to use Robotic Process Automation (RPA) to make the organization faster, leaner, and better than ever. But first, they needed to get the rest of...



SUCCESS STORY

The New York Foundling

The New York Foundling gets back close to 100,000 hours in manual work annually by leveraging RPA



SUCCESS STORY

American Fidelity

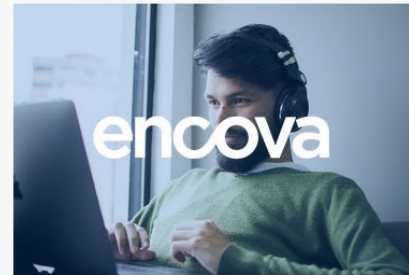
With UiPath and DataRobot, American Fidelity successfully automated many of its most critical customer-facing processes.



SUCCESS STORY

PwC

Empowering employees by rolling out RPA to its 50,000+ US and Mexico employees and is in the process of expanding this globally



SUCCESS STORY

Encova

Delights Agents and Improves CX by Automating Tedious, Time-Consuming Tasks



SUCCESS STORY

dentsu

Learn how dentsu built 60 robots in 30 days and saved 125K hours in the process

129. Upon information and belief, and particularly by way of the detailed documentation instructing users on how to use the Infringing Products in an infringing manner (*see supra*, ¶¶ 124, 128), Defendants have encouraged this infringement with knowledge of the '491 Patent and with a specific intent to cause their customers and distributors to infringe.

130. Defendants' acts thus constitute active inducement of patent infringement in violation of 35 U.S.C. § 271(b).

131. Defendants will, on information and belief, continue to induce infringement of the '491 Patent unless enjoined.

132. Defendants' direct infringement, contributory infringement, and inducement of infringement have irreparably harmed SRI.

//

//

1 133. Defendants will, on information and belief, continue to irreparably harm SRI unless
2 enjoined.

3 134. Pursuant to 35 U.S.C. § 284, SRI is entitled to damages adequate to compensate for
4 the infringement but in no event less than a reasonable royalty.

5 135. This case is “exceptional” within the meaning of 35 U.S.C. § 285, and SRI is entitled
6 to an award of attorneys’ fees.

7 **COUNT VII – INFRINGEMENT OF THE ’286 PATENT**

8 136. SRI re-alleges and incorporates the allegations of the preceding paragraphs of this
9 Complaint as if fully set forth herein.

10 137. SRI is the assignee and owner of all right, title, and interest in and to the ’286 Patent,
11 which was issued on November 26, 2019. A true and correct copy of the ’286 Patent is attached
12 hereto as Exhibit G.

13 138. The ’286 Patent addresses an invention for testing websites. The disclosed innovation
14 tests many facets of the website’s experience and operation, including by providing novel approaches
15 to creating, storing, and executing test scripts using website elements as opposed to the previously
16 disclosed use of recording test scripts based upon user actions only.

17 139. SRI has the exclusive right to make, use, sell, and offer to sell any product embodying
18 the ’286 Patent throughout the United States, and to import any product embodying the ’286 Patent
19 into the United States.

20 140. SRI has commercially exploited the ’286 Patent by making, marketing, selling, and
21 using products covered by the ’286 Patent, including its popular eValid™ software products. SRI
22 continues to commercially exploit the ’286 Patent through the present, at least by continuing to
23 provide maintenance and support to users of its popular eValid™ software products.

24 141. Defendants have had knowledge of the ’286 Patent, SRI, and SRI’s products
25 embodying the inventions claimed in the Patents-in-Suit since at least as early as the filing of this
26 Complaint.

27 142. At all relevant times, SRI provided public notice of the ’286 Patent by properly
28 marking its products and its website pursuant to 35 U.S.C. § 287(a).

143. Defendants have been, and are currently, directly infringing at least claim 1 of the '286 Patent in violation of 35 U.S.C. § 271(a), literally or under the doctrine of equivalents, by making, using, selling, offering for sale, and/or importing into the United States Defendants' Infringing Products, which, as set forth in documentation available on Defendants' website, comprise the computing device disclosed in the '286 Patent—both as maintained in Defendants' files and as made accessible to its users to whom Defendants offer and sell the Infringing Products—including at least a memory; web browser program code stored in the memory; and a processor configured to perform the web browser program code, wherein the web browser program code, when performed, provides a web browser operating on the computing device (for example, the UiPath Platform allows a user to browse the web via common web browsing activities, such as navigating to a web page, clicking and double clicking a mouse, or selecting an item from a drop-down list or combo box (<https://docs.uipath.com/activities/docs/navigate-to>; <https://docs.uipath.com/studio/v2018.3/docs/manual-recording>; <https://docs.uipath.com/activities/docs/double-click>)); wherein the web browser program code provides the web browser with Document Object Model (DOM) access capabilities (for example, the UiPath Platform locates these pages elements based on their DOM indexes, which necessarily requires it to use the DOM access methods included in Dynamic Linked Libraries associated with a browser code library (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/ui-element-exists>; <https://docs.uipath.com/activities/docs/on-ui-element-appear>)); wherein the web browser program code, executable by the computing device, includes at least: computer program code for testing and analysis of a web page as rendered by the web browser (for example, the UiPath Platform allows a user to create or record a test using a browser and then play it back by executing it (“Recording can help you save a lot of time when automating your tasks. The App/Web Recorder captures your actions as you perform them on the screen and generates a Use Application/Browser activity with a series of activities inside it based on your actions.” (<https://docs.uipath.com/activities/docs/app-web-recorder>); “There are four types of recordings available in UiPath Studio: ... Web – designed for recording in web apps and browsers, generates containers and uses the Simulate Type/Click input method by default.”

(<https://docs.uipath.com/studio/v2018.3/docs/about-recording-types>); “Executing tests can be done either manually or automatically. Manual execution is invoked via Test Manager while automated execution is invoked via UiPath Orchestrator only.” (<https://docs.uipath.com/test-suite/docs/executing-tests>)); computer program code for accessing an attribute or property value of an element of a DOM of the web page, wherein the computer program code for accessing the attribute or property value of the element of the DOM of the web page accesses the DOM of the web page using a browser programming interface that enables the web browser program code to have access to the DOM (for example, the UiPath Platform locates these pages elements based on their DOM indexes, which necessarily requires it to use the DOM access methods included in Dynamic Linked Libraries associated with a browser code library (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>; <https://docs.uipath.com/activities/docs/ui-element-exists>; <https://docs.uipath.com/activities/docs/on-ui-element-appear>)); the browser programming interface is supported by an API underlying the web browser program code for providing a plurality of library function calls or methods that are accessible by the web browser program code (for example, the UiPath Platform includes a browser programming interface capable of accessing WebDriver, which functions as is an underlying API (<https://docs.uipath.com/studio/docs/about-the-webdriver-protocol>)); and wherein the computer program code for accessing the attribute or property value of the element of the DOM of the web page accesses the attribute or property value of the element of the DOM of the web page for purposes of the testing and analysis of the web page rendered in the web browser (for example, the UiPath Platform interrogates the DOM to identify and extract relevant information regarding at least the page elements germane to the test, including each such element’s value, and stores those details in the test (<https://docs.uipath.com/activities/docs/get-attribute>; <https://docs.uipath.com/activities/lang-en/docs/get-value>; <https://docs.uipath.com/activities/docs/extract-data-from-website>; <https://docs.uipath.com/activities/docs/digitize-document>)); the UiPath Platform uses explicit wait commands, such as a command that “[w]aits for the specified UI element to appear on the screen” or to “[w]aits for the value of specified UI element attribute to be equal to a string” and, in order to perform such a verification, the UiPath Platform must necessarily store facts about the webpage being

1 rendered, i.e., the expected condition to be checked for during verification
 2 (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>;
 3 <https://docs.uipath.com/activities/docs/wait-attribute>; [https://docs.uipath.com/activities/docs/execute-
 4 x-path](https://docs.uipath.com/activities/docs/execute-x-path)); wherein the web browser program code supports at least one command, provided to the web
 5 browser via the browser programming interface, to facilitate synchronized testing and analysis of
 6 asynchronous processes of the web page rendered by the web browser using the underlying API (for
 7 example, the UiPath Platform allows for the testing of content dynamically generated by AJAX
 8 programming including using, upon information and belief, its various wait commands to
 9 synchronize playback and allow for testing of content dynamically generated by AJAX programming
 10 (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>;
 11 <https://docs.uipath.com/activities/docs/wait-attribute>;
 12 <https://www.uipath.com/solutions/technology/web-scraping-software>;
 13 <https://www.uipath.com/solutions/technology/web-automation>)); and wherein the at least one
 14 command includes a DOM index value, a DOM property name and a DOM property value, and
 15 causes examination of a name and a value of a property found in the DOM of the web page at the
 16 DOM index value to determine whether the name and the value match the DOM property name and
 17 the DOM property value, respectively (for example, to generate and subsequently perform validation
 18 tests, the UiPath Platform interrogates the DOM to identify and extract relevant information
 19 regarding at least the page elements germane to the test, including each such element's value, and
 20 stores those details in the test (<https://docs.uipath.com/activities/docs/get-attribute>;
 21 <https://docs.uipath.com/activities/lang-en/docs/get-value>;
 22 <https://docs.uipath.com/activities/docs/extract-data-from-website>;
 23 <https://docs.uipath.com/activities/docs/digitize-document>)); the UiPath Platform uses explicit wait
 24 commands, such as a command that “[w]aits for the specified UI element to appear on the screen” or
 25 to “[w]aits for the value of specified UI element attribute to be equal to a string” and, in order to
 26 perform such a verification, the UiPath Platform must necessarily store facts about the webpage being
 27 rendered, i.e., the expected condition to be checked for during verification
 28 (<https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc>;

1 <https://docs.uipath.com/activities/docs/wait-attribute>; [https://docs.uipath.com/activities/docs/execute-](https://docs.uipath.com/activities/docs/execute-x-path)
 2 [x-path](https://docs.uipath.com/activities/docs/execute-x-path)); the UiPath Platform locates these pages elements based on their DOM indexes, which
 3 necessarily requires it to use the DOM access methods included in Dynamic Linked Libraries
 4 associated with a browser code library ([https://docs.uipath.com/activities/docs/wait-ui-element-](https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc)
 5 [appear#misc](https://docs.uipath.com/activities/docs/wait-ui-element-appear#misc); <https://docs.uipath.com/activities/docs/ui-element-exists>;
 6 <https://docs.uipath.com/activities/docs/on-ui-element-appear>); as disclosed in the '286 Patent.

7 144. Defendants will, on information and belief, continue to directly infringe the '286
 8 Patent unless enjoined.

9 145. To the extent Defendants' Infringing Products, without more, do not directly infringe
 10 at least claim 1 of the '286 Patent, at least as of the filing of this Complaint, Defendants contribute to
 11 infringement of the same under 35 U.S.C. § 271(c) inasmuch as the Infringing Products offered for
 12 sale and sold by Defendants are each a component of a patented machine or an apparatus used in
 13 practicing a patented process, constituting a material part of SRI's invention, knowing the same to be
 14 especially made or especially adapted for use in infringement of the '286 Patent. For example, as set
 15 forth above, the UiPath Platform, when used in its normal and intended usage (pursuant to the
 16 instructions set forth on Defendants' website), infringes claim 1 of the '286 Patent. *See supra*, ¶ 143.

17 146. Defendants will, on information and belief, continue to contribute to infringement of
 18 the '286 Patent unless enjoined.

19 147. Defendants actively encourage their customer to use Defendants' Infringing Products
 20 in an infringing manner. For example, Defendants' website is replete with written directions,
 21 screenshots, and videos instructing users on how to use the Infringing Products in an infringing
 22 manner. For example, as set forth above, Defendants' website regarding the UiPath Platform
 23 specifically instructs users of the Infringing Products how to infringe claim 1 of the '286 patent. *See*
 24 *supra*, ¶ 143. Defendants' website also touts the identities of customers who use the Infringing
 25 Products, each of whom is a direct infringer inasmuch as they use the Infringing Products in the
 26 infringing manner as instructed by Defendants:

27 //

28 //



SUCCESS STORY

Deloitte

Deloitte teams set out to use Robotic Process Automation (RPA) to make the organization faster, leaner, and better than ever. But first, they needed to get the rest of...



SUCCESS STORY

The New York Foundling

The New York Foundling gets back close to 100,000 hours in manual work annually by leveraging RPA



SUCCESS STORY

American Fidelity

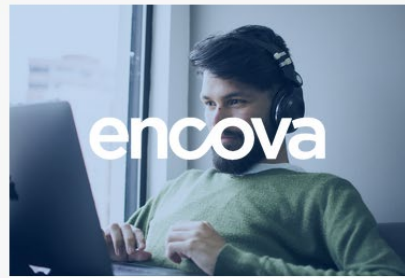
With UiPath and DataRobot, American Fidelity successfully automated many of its most critical customer-facing processes.



SUCCESS STORY

PwC

Empowering employees by rolling out RPA to its 50,000+ US and Mexico employees and is in the process of expanding this globally



SUCCESS STORY

Encova

Delights Agents and Improves CX by Automating Tedious, Time-Consuming Tasks



SUCCESS STORY

dentsu

Learn how dentsu built 60 robots in 30 days and saved 125K hours in the process

148. Upon information and belief, and particularly by way of the detailed documentation instructing users on how to use the Infringing Products in an infringing manner (*see supra*, ¶¶ 143, 147), Defendants have encouraged this infringement with knowledge of the '286 Patent and with a specific intent to cause their customers and distributors to infringe.

149. Defendants' acts thus constitute active inducement of patent infringement in violation of 35 U.S.C. § 271(b).

150. Defendants will, on information and belief, continue to induce infringement of the '286 Patent unless enjoined.

//

//

//

//

151. Defendants' direct infringement, contributory infringement, and inducement of infringement have irreparably harmed SRI.

152. Defendants will, on information and belief, continue to irreparably harm SRI unless enjoined.

153. Pursuant to 35 U.S.C. § 284, SRI is entitled to damages adequate to compensate for the infringement but in no event less than a reasonable royalty.

154. This case is "exceptional" within the meaning of 35 U.S.C. § 285, and SRI is entitled to an award of attorneys' fees.

DEMAND FOR JURY TRIAL

SRI hereby demands a trial by jury of all issues so triable under Federal Rule of Civil Procedure 38(b).

PRAYER FOR RELIEF

WHEREFORE, SRI respectfully requests that this Court:

- A. Find that United States Patent No. 7,757,175 is valid and enforceable against Defendants;
- B. Find that Defendants have infringed and are infringing United States Patent No. 7,757,175;
- C. Permanently enjoin Defendants, their officers, agents, servants, employees, and those persons acting in active concert or in participation therewith from infringing United States Patent No. 7,757,175;
- D. Award SRI damages sufficient to compensate it for Defendants' past and future infringement of United States Patent No. 7,757,175, together with costs and prejudgment interest, pursuant to 35 U.S.C. § 284;
- E. Find that United States Patent No. 8,327,271 is valid and enforceable against Defendants;
- F. Find that Defendants have infringed and are infringing United States Patent No. 8,327,271;
- G. Permanently enjoin Defendants, their officers, agents, servants, employees, and those

persons acting in active concert or in participation therewith from infringing United States Patent No. 8,327,271;

H. Award SRI damages sufficient to compensate it for Defendants' past and future infringement of United States Patent No. 8,327,271, together with costs and prejudgment interest, pursuant to 35 U.S.C. § 284;

I. Find that United States Patent No. 8,392,890 is valid and enforceable against Defendants;

J. Find that Defendants have infringed and are infringing United States Patent No. 8,392,890;

K. Permanently enjoin Defendants, their officers, agents, servants, employees, and those persons acting in active concert or in participation therewith from infringing United States Patent No. 8,392,890;

L. Award SRI damages sufficient to compensate it for Defendants' past and future infringement of United States Patent No. 8,392,890, together with costs and prejudgment interest, pursuant to 35 U.S.C. § 284;

M. Find that United States Patent No. 8,495,585 is valid and enforceable against Defendants;

N. Find that Defendants have infringed and are infringing United States Patent No. 8,495,585;

O. Permanently enjoin Defendants, their officers, agents, servants, employees, and those persons acting in active concert or in participation therewith from infringing United States Patent No. 8,495,585;

P. Award SRI damages sufficient to compensate it for Defendants' past and future infringement of United States Patent No. 8,495,585, together with costs and prejudgment interest, pursuant to 35 U.S.C. § 284;

Q. Find that United States Patent No. 8,650,493 is valid and enforceable against Defendants;

R. Find that Defendants have infringed and are infringing United States Patent No.

8,650,493;

S. Permanently enjoin Defendants, their officers, agents, servants, employees, and those persons acting in active concert or in participation therewith from infringing United States Patent No. 8,650,493;

T. Award SRI damages sufficient to compensate it for Defendants' past and future infringement of United States Patent No. 8,650,493, together with costs and prejudgment interest, pursuant to 35 U.S.C. § 284;

U. Find that United States Patent No. 8,984,491 is valid and enforceable against Defendants;

V. Find that Defendants have infringed and are infringing United States Patent No. 8,984,491;

W. Permanently enjoin Defendants, their officers, agents, servants, employees, and those persons acting in active concert or in participation therewith from infringing United States Patent No. 8,984,491;

X. Award SRI damages sufficient to compensate it for Defendants' past and future infringement of United States Patent No. 8,984,491, together with costs and prejudgment interest, pursuant to 35 U.S.C. § 284;

Y. Find that United States Patent No. 10,489,286 is valid and enforceable against Defendants;

Z. Find that Defendants have infringed and are infringing United States Patent No. 10,489,286;

AA. Permanently enjoin Defendants, their officers, agents, servants, employees, and those persons acting in active concert or in participation therewith from infringing United States Patent No. 10,489,286;

BB. Award SRI damages sufficient to compensate it for Defendants' past and future infringement of United States Patent No. 10,489,286, together with costs and prejudgment interest, pursuant to 35 U.S.C. § 284;

CC. Order an accounting of damages from Defendants' infringement;

- 1 DD. Award SRI its reasonable attorney fees and costs of suit pursuant to 35 U.S.C. § 285
2 due to the exceptional nature of this case, or as otherwise permitted by law;
3 EE. Award SRI post-judgment interest pursuant to 28 U.S.C. § 1961; and
4 FF. Award SRI such other or additional relief as the Court deems just and proper.
5

6 Date: June 7, 2021

Respectfully submitted,
SINGER CASHMAN LLP

7
8
9 By: _____

Benjamin L. Singer

Evan Budaj

Attorneys for Plaintiff Software Research, Inc.

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28