MICHAEL A. SHERMAN (SBN 94783)
masherman@stubbsalderton.com
JEFFREY F. GERSH (SBN 87124)
jgersh@stubbsalderton.com
SANDEEP SETH (SBN 195914)
sseth@stubbsalderton.com
WESLEY W. MONROE (SBN 149211)
wmonroe@stubbsalderton.com
STANLEY H. THOMPSON, JR. (SBN 198825)
sthompson@stubbsalderton.com
VIVIANA BOERO HEDRICK (SBN 239359)
vhedrick@stubbsalderton.com
STUBBS, ALDERTON & MARKILES, LLP
15260 Ventura Blvd., 20th Floor
Sherman Oaks, CA 91403
Telephone: (818) 444-4500
Facsimile: (818) 444-4520

Attorneys for PersonalWeb Technologies, LLC

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN JOSE DIVISION

IN RE PERSONALWEB TECHNOLOGIES, LLC, ET AL., PATENT LITIGATION

CASE NO.: 5:18-md-02834-BLF

PERSONALWEB TECHNOLOGIES, LLC, a Texas limited liability company, and LEVEL 3 COMMUNICATIONS, LLC, a Delaware limited liability company,

Plaintiffs,

v.

TWITCH INTERACTIVE, INC. a Delaware corporation,

Defendant.

Case No.: 5:18-cv-05619-BLF

DECLARATION OF ERIK DE LA IGLESIA IN SUPPORT OF PERSONALWEB TECHNOLOGIES, LLC'S NON-OPPOSITION TO TWITCH INTERACTIVE, INC. MOTION FOR SUMMARY JUDGEMENT OF NONINFRINGEMENT AND PARTIAL OPPOSITION TO MOTION TO EXCLUDE TESTIMONY OF ERIK DE LA IGLESIA

Trial Date: March 16, 2020

DECLARATION OF ERIK DE LA IGLESIA

I, Erik de la Iglesia, declare as follows:

1. I am over the age of eighteen (18) and make this declaration of my own personal knowledge, under penalty of perjury.

2. I have been retained as an independent expert witness by the law firm of Stubbs Alderton & Markiles, LLP on behalf of PersonalWeb Technologies, LLC ("PersonalWeb") to testify as a technical expert in lawsuits concerning U.S. Patent No. 6,928,442 ("'442 Patent"), U.S. Patent No. 7,802,310 ("'310 Patent"), and U.S. Patent No. 8,099,420 ("'420 Patent") (collectively, "the Asserted Patents"), the lawsuits including In re PersonalWeb Technologies, LLC, et al., Patent Litigation, Case No.: 5:18-md-02834-BLF (Northern District of California) and PersonalWeb Technologies, LLC v. Twitch Interactive, Inc., Case No. 5:18-cv-05619-BLF (Northern District of California). I refer to Twitch Interactive, Inc. as "Twitch" in this declaration.

3. On August 23, 2019, I submitted my report summarizing my findings regarding infringement by Twitch. For at least all the reasons summarized in that report, it is my opinion that Twitch's web server met certain limitations of each of claim 20 of the '310 patent, claims 25, 26, 27, 32, 34, 35, 36, and 166 of the '420 patent, and claims 10 and 11 of the '442 patent. I understand from counsel that PersonalWeb is withdrawing certain portions of my August 23 report. A true and correct copy of my August 23 report with the redactions is attached hereto as Exhibit 1, which I verify under penalty of perjury.

4. The asserted patent claims relate to controlling the distribution of files in a network of computers. Requests for content or access to content are permitted or not permitted by the system using specific methods that include the use of content-based identifiers. This subject matter includes the protocols used to transfer those files, technology such as caching to accelerate distribution and the configuration of such caching to optimize efficiency using content-based identifiers.

5. While the evidence of the claim limitations and my analysis are detailed in my report, I will address in this declaration certain specific points raised by Twitch in its summary judgment motion that relates to:

a. How Twitch used MD5 ETags that were generated by applying the MD5 hash algorithm to the content and only the content of a Twitch webpage file to determine whether or not to send a message that permitted browsers to keep using cached version of that webpage file after the original permitted time to use that cached version has expired;

b. How Twitch used MD5 ETags to determine whether or a file at a browser was a copy of the current version of a webpage file in making the decision of (a); and

c. How Twitch compared an MD5 ETag sent in a conditional GET request from a browser to see if it matched one of a plurality of stored ETags in making the determinations of (a) and (b).

6. More particularly, as I explain below, Twitch servers sent Twitch webpage file content in HTTP 200 messages with MD5 ETags and max-age values set by Twitch. By doing so, the Twitch server instructed browsers operating under the HTTP 1.1 protocol how long they were permitted to use the file content without having to first check back with Twitch whether they may still continue to use the content after their permitted use of the content has expired. After the permitted time to use the content has expired, the browser sent a conditional GET request to which it must receive an HTTP 304 response to continue to access and use the cached file content.

7. If the browser instead received from a Twitch server an HTTP 200 response to the conditional GET request, it used the content provided in the 200 response instead of the previously cached content. Moreover, the Twitch servers used MD5 ETags (i.e., ETag values generated by applying the MD5 hash algorithm to the file content and only the file content) in making the decision whether or not to continue to permit the browsers' access to the previously cached file content or to provide new file content for the browser to access and use instead of the previously cached file content.

8. The MD5 ETags informed the Twitch server whether a copy of the current version of the webpage file was already cached (present) at the browser or whether a copy of the current version needed to be provided. If a copy of the current version was determined to be already

present at the browser, Twitch sent the HTTP 304 message permitting the browser to continue accessing the cached copy. If the file at the browser was determined to be a copy of the current file version, the Twitch server sent the HTTP 200 message for the browser to access instead of the previously cached version. By using this system of HTTP 304 and 200 messages, Twitch controlled how long browsers accessed Twitch's webpage file content and what webpage file content they accessed.

9. I will now address Twitch's three summary judgment arguments that are not based upon the Court's construction of "unauthorized or unlicensed."

10. Claim 20 of the '310 patent recites, in relevant part:

> based at least in part on said content-dependent name of said particular data item, the first device (A) permitting the content to be provided to or accessed by the at least one other computer if it is not determined that the content is unauthorized or unlicensed, otherwise, (B) if it is determined that the content is unauthorized or unlicensed, not permitting the content to be provided to or accessed by the at least one other computer.

11. The evidence that I have reviewed shows that Twitch's webpage servers each made a determination to permit or not permit content to be provided to or accessed by a client, such as a browser, based at least on part on an MD5 ETag value, which is a content-dependent name of said particular data item. The Twitch servers operated during the relevant infringement time period in accordance with the HTTP 1.1 protocol, RFC 2616. Specifically, the servers communicated with connected computers communicate via messages, including but not limited to those specified in RFC 2616 regarding GET requests ("HTTP GET requests") (e.g., Sec. 9.3), conditional GET requests ("HTTP conditional GET requests") with If-None-Match Headers (e.g., Sec. 14.9.4), ETags (e.g., Sec. 14.19), 304 messages ("HTTP 304 messages") (e.g., Sec. 10.3.5), 200 messages ("HTTP 200 messages") (e.g., Sec. 10.2.1), and cache control directives (e.g., Secs. 13.1, 13.2, 13.3.2-4, 14.9, 14.21, 14.26) to implement cache control including in instructing browsers when they were allowed to re-use previously cached content or had to use instead use

newly provided content.

12. HTTP 1.1 provides a mechanism for using ETags to instruct clients (such as browsers) whether or not file content stored in their caches may continue to be used to fulfill requests for content after their original permitted time to use the content has expired. More particularly, HTTP 1.1 allowed website operators to send the file content in an HTTP 200 message with an "ETag" value for that content and a "max-age" value (*i.e.*, a permitted time to use the content) and force a browser to check back with the server before using that content after the permitted time had expired. If a requested file is served along with a max-age caching directive and an ETag value, the client browser cache will store the file, the max-age and the ETag. As long as the file's age in the cache is less than the max-age, the client cache will reuse the file for future requests. (RFC 2616 @ 51-52). However, after the permitted time to use the content has been exceeded, conditional GETs must be used to revalidate that the client is permitted to keep using the cached file content for some extended period of time.

13. The evidence I reviewed confirmed that, during the relevant time period, Twitch servers used content-based ETags that were generated by applying the MD5 hash algorithm to the content, and only the content, of the associated file. My evidentiary review also confirmed that the servers sent the MD5 ETag along with the file content and cache control directives in HTTP 200 messages and subsequently compared such MD5 ETags sent by clients (e.g. browsers and intermediate cache servers) in conditional GET requests with the current ETag values for the requested file stored at the server.

14. The following source code that Twitch's server used compared the ETag sent by a browser in a conditional GET request with a value for a data item stored at the server:

```
if (ngx_strncmp(start, etag->data, etag->len) != 0).
```

(PERSONALWEB106919, at line 193.)

15. If the server processing the conditional GET request verified that the ETag sent in the "If-None-Match" request header of the conditional GET request matched the current MD5 ETag value of the requested file, the server then made a determination to permit a browser to keep using and accessing the cached content when it sent a 304 NOT MODIFIED message to the

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.