IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
WACO DIVISION

ANCORA TECHNOLOGIES, INC. Plaintiff, v. ROKU, INC., Defendant.	Civil Action No. 6:21-cv-00737 Jury Trial Requested

COMPLAINT FOR PATENT INFRINGEMENT

This is an action for patent infringement in which Ancora Technologies, Inc. makes the following allegations against Roku, Inc. ("Roku"):

RELATED CASE

1. This case is related to the actions *Ancora Technologies, Inc. v. Google, Inc.* (W.D. Tex. Jul. 16, 2021); *Ancora Technologies Inc. v. Nintendo Co. Ltd. et al.* (W.D. Tex. Jul. 16, 2021); and *Ancora Technologies Inc. v. Vizio, Inc.* (W.D. Tex. Jul. 16, 2021)—each of which was filed on July 16, 2021, in the United States District Court for the Western District of Texas, Waco Division, asserting infringement of United States Patent No. 6,411,941.

PARTIES

2. Plaintiff Ancora Technologies, Inc. is a corporation organized and existing under the laws of the State of Delaware with a place of business at 23977 S.E. 10th Street, Sammamish, Washington 98075.

3. Defendant Roku, Inc., is a corporation organized and existing under the laws of Delaware with a principal place of business at 9606 N. Mopac Expressway, Suite 400, Austin, Texas.

JURISDICTION AND VENUE

4. This action arises under the patent laws of the United States, Title 35 of the United States Code, such that this Court has subject matter jurisdiction pursuant to 28 U.S.C. §§ 1331 and 1338(a).

5. This Court also has personal jurisdiction over Roku, including because Roku also maintains a regular and established place of business in the Western District of Texas, including at 9606 N. Mopac Expressway, Suite 400, Austin, Texas.

6. In addition, directly or through intermediaries, Roku has committed acts within the Western District of Texas giving rise to this action and/or has established minimum contacts with the Western District of Texas such that the exercise of jurisdiction would not offend traditional notions of fair play and substantial justice.

7. For example, Roku has placed or contributed to placing infringing products like the Roku Ultra into the stream of commerce via an established distribution channel knowing or understanding that such products would be sold and used in the United States, including in the Western District of Texas.

8. Further, on information and belief, Roku also has derived substantial revenues from infringing acts in the Western District of Texas, including from the sale and use of infringing products like the Roku Ultra.

Nintendo - Ancora Exh. 1077

9. In addition, venue is proper under 28 U.S.C. § 1391(b)-(c) and 28 U.S.C. § 1400 as Roku maintains a regular and established place of business in the Western District of Texas, including at 9606 N. Mopac Expressway, Suite 400, Austin, Texas.

THE ASSERTED PATENT

10. This lawsuit asserts causes of action for infringement of United States Patent No. 6,411,941 ("the '941 Patent"), which is entitled "Method of Restricting Software Operation Within a License Limitation."

11. The U.S. Patent and Trademark Office duly and legally issued the '941 Patent on June 25, 2002.

12. Subsequent to issue, and at least by December 21, 2004, all right, title, and interest in the '941 Patent, including the sole right to sue for any infringement, were assigned to Ancora Technologies, Inc., which has held, and continues to hold, all right, title, and interest in the '941 Patent.

13. The president of Ancora Technologies, Inc.—Mr. Miki Mullor—is one of the inventors of the '941 Patent.

14. A reexamination certificate to the '941 Patent subsequently was issued on June 1, 2010.

15. Since being assigned to Ancora Technologies, Inc., the '941 Patent has been asserted in patent infringement actions filed against Microsoft Corporation, Dell Incorporated, Hewlett Packard Incorporated, Toshiba America Information Systems, Apple Inc., HTC America, Inc., HTC Corporation, Samsung Electronics Co., Ltd., Samsung Electronics America, Inc., LG Electronics, Inc., LG Electronics U.S.A., Inc., Sony Mobile Communications AB, Sony Mobile Communications, Inc., Sony Mobile Communications (USA) Inc., Lenovo Group Ltd., Lenovo

Nintendo - Ancora Exh. 1077

(United States) Inc., Motorola Mobility, LLC, TCT Mobile (US) Inc., and Huizhou TCL Mobile Communication Co., Ltd.

16. In the course of these litigations, a number of the ’941 Patent’s claim terms have been construed, and the validity of the ’941 Patent has been affirmed repeatedly.

17. For example, in December 2012, the United States District Court for the Northern District of California issued a claim construction order construing the terms (1) “volatile memory”; (2) “non-volatile memory”; (3) “BIOS”; (4) “program”; (5) “license record”; and (6) “verifying the program using at least the verification structure.” *Ancora Techs., Inc. v. Apple Inc.*, No. 11–CV–06357 YGR, 2012 WL 6738761, at *1 (N.D. Cal. Dec. 31, 2012).

18. Further, the court rejected Apple’s indefiniteness arguments and further held that, at least with respect to Claims 1-3 and 5-17, “[t]he steps of the Claim do not need to be performed in the order recited.” *Ancora Techs., Inc. v. Apple Inc.*, No. 11–CV–06357 YGR, 2012 WL 6738761, at *5, *13 (N.D. Cal. Dec. 31, 2012).

19. Subsequently, the United States Court of Appeals for the Federal Circuit affirmed the district court’s rejection of Apple’s indefiniteness argument. *Ancora Techs., Inc. v. Apple, Inc.*, 744 F.3d 732, 739 (Fed. Cir. 2014).

20. The Federal Circuit also agreed with Ancora Technologies, Inc. that “the district court erred in construing ‘program’ to mean ‘a set of instructions for software applications that can be executed by a computer’”—holding that, as Ancora had argued, the term should be accorded its normal meaning of “‘a set of instructions’ for a computer.” *Ancora Techs., Inc. v. Apple, Inc.*, 744 F.3d 732, 734-35, 737 (Fed. Cir. 2014).

21. Subsequently, in a more recent decision, the Federal Circuit held that the ’941 Patent satisfied § 101 as a matter of law—stating: “[W]e conclude that claim 1 of the ’941 patent is not

Nintendo - Ancora Exh. 1077

directed to an abstract idea." *Ancora Techs., Inc. v. HTC Am., Inc.*, 908 F.3d 1343 (Fed. Cir. 2018), *as amended* (Nov. 20, 2018).

22. In addition, the Patent Trial and Appeal Board rejected HTC's request to institute covered business method review proceedings on the '941 Patent—explaining that "the '941 [P]atent's solution to the addressed problem is rooted in technology, and thus, is a 'technical solution'" and also rejecting HTC's argument that "the '941 [P]atent recites a technological solution that is not novel and nonobvious."

23. This Court likewise issued a claim construction order construing or adopting the plain and ordinary meaning of various claims of the '941 Patent, including (1) "non-volatile memory"; (2) "license"; (3) "license record"; (4) "volatile memory"; (5) "BIOS"; (6) "memory of the BIOS"; (7) "program"; (8) "selecting a program residing in the volatile memory"; (9) "using an agent to set up a verification structure in the erasable, non-volatile memory of the BIOS"; (10) "set up a verification structure"; (11) "verifying the program using at least the verification structure"; (12) "acting on the program according to the verification"; (13) "first non-volatile memory area of the computer"; (14) the Claim 1 preamble; and (15) the order of Claim 1 steps. *Ancora Technologies, Inc. v. LG Electronics, Inc.*, 1:20-cv-00034-ADA, at Dkt. 69 (W.D Tex. June 2, 2020).

24. Finally, and most recently, the United States District Court for the Central District of California issued a claim construction order construing the terms (1) "volatile memory"; (2) "selecting a program residing in the volatile memory"; (3) "set up a verification structure"; (4) "license record"; (5) "memory of the BIOS"; and (6) the whole of Claim 8. *Ancora Techs., Inc v. TCT Mobile (US), Inc., et al.*, No. 8:19-cv-02192-GW-AS, ECF No. 66 & 69 (C.D. Cal. Nov. 18-19, 2020).

Nintendo - Ancora Exh. 1077

COUNT 1 – INFRINGEMENT

25. Plaintiff repeats and incorporates by reference each preceding paragraph as if fully set forth herein and further state:

26. Roku has infringed the ’941 Patent in violation of 35 U.S.C. § 271(a) by, prior to the expiration of the ’941 Patent, selling, and/or offering for sale in the United States, and/or importing into the United States, without authorization, products (including operating system software for products) that are capable of performing at least Claim 1 of the ’941 Patent literally or under the doctrine of equivalents and, without authorization, then causing such products to perform each step of at least Claim 1 of the ’941 Patent.

27. At a minimum, such Accused Products include those servers/software utilized by Roku to transmit an over-the-air (“OTA”) software update, as well as those televisions, streaming players, and other devices and technology that included Roku’s operating system software and to which Roku sent or had sent an OTA update that caused such device to perform the method recited in Claim 1 prior to the expiration of the ’941 Patent.

28. Such Accused Products include products like the Roku Ultra, which—as detailed below—Roku configured such that it would be capable of performing each step of Claim 1 of the ’941 Patent and subsequently provided one or more OTA updates that caused the device to perform each step of Claim 1.[1]

29. Such Accused Products also include products like the Roku 2, Roku 3, Roku 4, Roku Express, Roku Express+, Roku Premiere, Roku Premiere+, Roku Streaming Stick, Roku Streaming Stick+, and various Roku televisions and Smart TVs (including the TCL S-SERIES LED HD TV,

[1] This description of infringement is illustrative and not intended to be an exhaustive or limiting explanation of every manner in which each Accused Product infringes the ’941 patent. Further, on information and belief, the identified functionality of the Roku Ultra is representative of components and functionality present in all Accused Products.

Nintendo - Ancora Exh. 1077

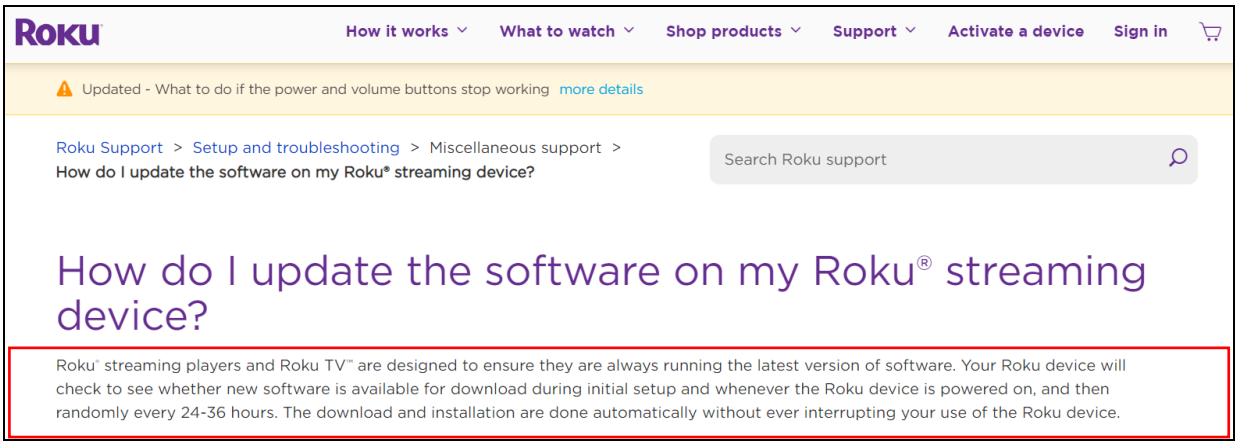
TCL 32FS4610R, TCL 40FS4610R, TCL 48FS4610R, TCL 55FS4610R, TCL S3700, TCL 32FS3700, TCL 48FS3700, TCL 50FS3700, TCL 32S3800, TCL 48FS3800, TCL 32S3850, TCL 40FS3850, TCL 50FS3850, TCL 55FS3850, TCL 50UP130, TCL 50FS3800, TCL 40FS3800, TCL 55UP130, TCL 55US5800, TCL 65US5800, TCL 43UP130, TCL UP120, TCL 55UP120, TCL 50UP120, TCL 43UP120, TCL 55P607, TCL 55P605, TCL 75C803, TCL 75C807, TCL 65C807, TCL 55C807, TCL 55C803, TCL 28S305, TCL 32S305, TCL 40S305TCL, TCL 55FS3750, TCL 28S3750, TCL 32S3750, TCL 40FS3750, TCL 48FS3750, TCL 43S305, Element E2AA32R-C, Element E2AA40R-G, Element E2AA40R-T, Element E1AA24R-G, RCA Roku Smart TV, RCA RTR4261, RCA RTRU5028, RCA RTR4061, RCA RTRU4927, RCA RTRU5027, RCA RTRU5527, RCA RTRU6527, Hisense 40H4C, Hisense 50H4C, Hisense 40H4C1, Hisense 48H4C1, Hisense 50H4C1, Hisense R7 Series ROKU TV, Hisense 65R7E, Hisense 43R7E, Hisense 50R7E, Hisense 55R7E, Hisense 43R7E7020E, Hisense 50R7E7020E, Hisense 55RE7020E, Hisense 65RE7020E, Hisense 43RE7020E, Hisense 50RE7030E, Hisense 55RE7030E, Hisense 65RE7030E, Hisense 43RE7060E, Hisense 50RE7060E, Hisense 55RE7060E, Hisense 65RE7060E, Hisense 43RE7070E, Hisense 50RE7070E, Hisense 55RE7070E, Hisense 65RE7070E, Hisense 43RE7080E, Hisense 50RE7080E, Hisense 55RE7080E, Hisense 65RE7080E, Hisense R6 Series ROKU TV, Hisense 43R6E, Hisense 50R6E, Hisense 55R6E, Hisense 60R6E, Hisense 65R6E, Philips 4000 series LED-LCD TV 40PFL4662/F7, Philips 4000 series LED-LCD TV 50PFL4662/F7, Philips 4000 series LED-LCD TV 24PFL4664/F7, Hitachi 43RC63, Hitachi 55R82, Hitachi 43R80, Hitachi 65R80, Hitachi 55RH1, Hitachi 32RZ2, Hitachi 32RZ2, Hitachi 43R51, Insignia HDTV Roku TV, Insignia NS-39DR510NA17, Insignia NS-24DR220NA18, Insignia NS-40DR420NA16, Insignia NS-32DR310NA17, Insignia NS-49DR420NA18, Insignia NS-48DR510NA17, Insignia NS-55DR420NA16, Insignia NS-32DR420NA16, Insignia NS-

Nintendo - Ancora Exh. 1077

48DR420NA16, Insignia NS-24ER310NA17, Insignia NS-50DR710NA17, Insignia NS-55DR710NA17, Insignia NS-43DR710NA17, Insignia NS-50DR620NA18, Insignia NS-55DR620NA18, Insignia NS-65DR620NA18, and 4K UHD TV with HDR Roku TV), as well as any predecessor models to such devices, to which Roku sent, or had sent, an OTA update prior to the expiration of the '941 Patent.

30. For example, Claim 1 of the '941 Patent claims "a method of restricting software operation within a license for use with a computer including an erasable, non-volatile memory area of a BIOS of the computer, and a volatile memory area; the method comprising the steps of: [1] selecting a program residing in the volatile memory, [2] using an agent to set up a verification structure in the erasable, non-volatile memory of the BIOS, the verification structure accommodating data that includes at least one license record, [3] verifying the program using at least the verification structure from the erasable non-volatile memory of the BIOS, and [4] acting on the program according to the verification."

31. When Roku transmitted an OTA update like its Roku OS 8.0 updates, Roku performed and/or caused devices like the Roku Ultra to perform each element of Claim 1 as part of its Roku-specified, pre-configured software update process:



https://support.roku.com/article/208755668.

32. In particular, each Roku Ultra contains both erasable, non-volatile memory in the form of flash memory and volatile memory in the form of RAM memory. Such non-volatile memory includes a partition titled "Registry," which—on information and belief—is an example of BIOS memory:

> **roRegistry**
>
> The Registry is an area of non-volatile storage where a small number of persistent settings can be stored.
>
> The Registry provides a means for an application to write and read small amounts of data such as settings, scores, etc. The data persists even if the user exits the application and even if the player is rebooted. Registry data is removed only when the application explicitly removes it, the user uninstalls the application, which remove the registry for the application, or the user performs a factory reset, which removes the registry for all applications.
>
> Access to the registry is available through the roRegistry object. This object is created with no parameters:
>
> `CreateObject("roRegistry")`

https://developer.roku.com/docs/references/brightscript/components/roregistry.md.

33. Further, as detailed above, each Roku Ultra was configured by Roku to repeatedly check to see if a new software update as available, including through the following method:

> 2. **Connect your Roku player to your network and the internet**
> If you are using a wired network, connect an Ethernet cable to your Roku player and set up the wired connection. Otherwise, use the guidelines below to set up the wireless connection.
> - Choose your wireless network from the list of available networks and enter your password. This is most often the same network you use to connect your computer or smartphone to the internet. If you do not see your network, select **Scan again to see all networks**. For more assistance, see help finding your wireless network and password.
> - If you want to see your password as you type, choose **Show password**. This can help prevent entering the wrong password. Remember, passwords are case-sensitive – use the **shift** button ⬆ on the on-screen keyboard to enter a capital letter.
> - Once you enter your password and select **Connect**, your Roku Player will automatically connect to the internet. Alternatively, you can get more help if you are unable to connect to your wireless network.
>
> 3. **Wait for your Roku player to download the latest software**
> - Once connected to the internet, your Roku player will check for a Roku OS software update, download and install the latest version if necessary, and then reboot.

https://support.roku.com/article/223372368.

34. During this process, one or more OTA servers owned or controlled by Roku set up a verification structure in the erasable, non-volatile memory of the BIOS of the Roku Ultra by

transmitting to the device an OTA update, which the Roku Ultra is configured by Roku to save to the erasable, non-volatile memory of its BIOS:

```
Automatic software update from a TFTP server
============================================

Overview
--------

This feature allows to automatically store software updates present on a TFTP
server in NOR Flash. In more detail: a TFTP transfer of a file given in
environment variable 'updatefile' from server 'serverip' is attempted during
boot. The update file should be a FIT file, and can contain one or more
updates. Each update in the update file has an address in NOR Flash where it
should be placed, updates are also protected with a SHA-1 checksum. If the
TFTP transfer is successful, the hash of each update is verified, and if the
verification is positive, the update is stored in Flash.
```

https://roku.app.box.com/v/RokuOpenSourceSoftware/folder/51885036304.

35. As noted previously, on information and belief, such BIOS areas include what Roku refers to as the "Registry" memory area partition.

36. This OTA update contains a verification structure that includes data accommodating at least one license record.

37. Examples of such a license record include what is known as a Private Key and/or a Public Key, which may be encrypted with an SHA-1 checksum and/or RSA signature:

SetPrivateKey(keyFileName as String) as Integer

Description

Specifies the private key to use for signing.

Parameters

Name	Type	Description
keyFileName	String	Specifies the private key to be used for signing. The file name should specify a path, either in the package or a temp path.

Return Value

- 1 = The key is valid.
- 0 = The file does not contain a valid key.
- -1 = The file was not found.

SetPublicKey(keyFileName as String) as Integer

Specifies the public key to be used for verification.

Parameters

Name	Type	Description
keyFileName	String	Specifies the public key to be used for signing. The file name should specify a path, either in the package or a temp path.

Return Value

- 1 = The key is valid.
- 0 = The file does not contain a valid key.
- -1 = The file was not found.

https://developer.roku.com/docs/references/brightscript/interfaces/ifrsa.md.

38. Other examples include x509 and/or root certificate authority. https://developer.roku.com/docs/references/brightscript/interfaces/ifhttpagent.md.

39. Once the verification structure has been set up in the BIOS, the Roku Ultra is configured by Roku to reboot, load the OTA update into its volatile memory (e.g., RAM), and then use the at least one license record from the BIOS to verify the OTA update as part of its secure boot process:

Nintendo - Ancora Exh. 1077

System security

The Roku platform has been designed to be hardened against unauthorized attack. This process starts at the Roku factory as each system is individualized and uniquely keyed as a foundation for robust security. The platform supports a secure key store and hardware encryption engine. The core set of system software has been encrypted and is protected by a secure boot process and the use of signed binaries.

SSL is the primary method provided for developers to implement content and/or communications security for their application. The device supports both client and server authentication via SSL to provide a secure communications channel between trusted end-points.

https://developer.roku.com/docs/features/security.md.

40. Examples of such verification functions include the following:

```c
int aimage_v1_sanity_check (aimage_v1_header_t *header, unsigned int type, unsigned int length)
{
    return aimage_v1_process (header, NULL, type, length, SANITY_CHECK);
}

int aimage_v1_verify_hash (aimage_v1_header_t *header, unsigned int type, unsigned int length)
{
    return aimage_v1_process (header, NULL, type, length, VERIFY_HASH);
}

int aimage_v1_verify_signature (aimage_v1_header_t *header, unsigned int type, unsigned int length, const aimage_keypub_t *key)
{
    return aimage_v1_process (header, key, type, length, VERIFY_SIGNATURE);
}

int aimage_v1_verify_signature_dk (aimage_v1_header_t *header, unsigned int type, unsigned int length)
{
    aimage_keypub_t *key = platform_firmware_rsa_public_key(header->release_id);

    return aimage_v1_process (header, key, type, length, VERIFY_SIGNATURE);
}
```

```c
static int aimage_v1_process (aimage_v1_header_t *header, const aimage_keypub_t *key, unsigned int type, unsigned int length, aimage_process_mode_t mode)
{
    int i;
    SHA_CTX context;
    unsigned int t[2][RSA_MOD_WORDS_1024 + 1];
    unsigned int *reference_hash;
    unsigned int platform_mask = ~0xF;

    if (((unsigned long) header) & 0x03)            /* header pointer must be aligned... */
        return -2;

    if (mode == GENERATE_HASH) {
        header->magic = IMG_MAGIC;
        header->magic2 = IMG_MAGIC2;
        header->type = type;
        header->length = ((length + 3) & ~0x03);    /* image must be a multiple of 4 bytes, even if data is not */
        header->data_start_offset = 0;              /* prepend not supported when generating headers here... */
        header->data_length = length;               /* prepend not supported when generating headers here... */
    }
    else {
        if (header->magic != IMG_MAGIC || header->magic2 != IMG_MAGIC2 ||
            (type && (header->type != type)) ||
            (header->platform_id & platform_mask) != (platform_platform_id() & platform_mask) ||
            header->length > length || header->length < sizeof(aimage_v1_header_t))
        {
            return -2;
        }
        if (mode == SANITY_CHECK)
            return 0;                               /* our work here is done... */
    }
    SHA1_Init (&context);
    SHA1_Update (&context, (unsigned char *) header, offsetof (aimage_v1_header_t, signature));
    reference_hash = header->hash;
    if (mode == VERIFY_SIGNATURE) {
        SHA1_Update (&context, (unsigned char *) header->hash, sizeof(header->hash));
        abn_mont_exp (t[1], t[0], header->signature, key);
        for (i = 0; i < 5; i++)
            t[1][i] ^= t[1][RSA_MOD_WORDS_1024 - 5 + i];
        reference_hash = t[1];
    }
```

```
    t[0][0] = 0;
    SHA1_Update (&context, (unsigned char *) t[0], 4);
    SHA1_Update (&context, (unsigned char *) header->reserved, header->length - offsetof (aimage_v1_header_t, reserved));
    SHA1_Final ((unsigned char *) t[0], &context);
    for (i = 0; i < 5; i++) {
        if (mode == GENERATE_HASH)
            reference_hash[i] = t[0][i];
        else
            if (t[0][i] != reference_hash[i])
                return -1;
    }

    return 0;       /* success */
}
```

https://roku.app.box.com/v/RokuOpenSourceSoftware/folder/51885036304.

41. Examples of the use of such functions to verify an OTA Update include the following:

```
    if (hdr->type == IMG_TYPE_CUSTOM_PKG_TOKEN) {
        ret = aimage_v1_verify_signature_dk ((aimage_v1_header_t *) buf, type, hdr->length);
    } else if (hdr->type == IMG_TYPE_BOOTDATA) {
        // bootdata is hash-locked to u-boot.
        uint8_t digest[SHA_DIGEST_LENGTH];
        SHA_CTX ctx;
        SHA1_Init(&ctx);
        SHA1_Update(&ctx, buf, hdr->length);
        SHA1_Final(digest, &ctx);
        if (memcmp(digest, _bootdata_digest, sizeof(digest))) {
            printf("bootdata digest failure!\n");
            if (config_noaccess() && !custom_pkg_sideload)
                ret = -1;
            printf("ignore bootdata digest failure\n");
        }
```

42. If the OTA update is verified, the Roku Ultra is further configured to load and execute the update.

43. Further, during the infringing time period, Roku performed or caused to be performed each of the Claim 1 steps identified above by providing an OTA update to each Accused Product.

44. Further, Roku expressly conditions participation in the OTA update process and the receipt of the benefit of a software update on the performance of each of the above steps.

45. Primarily, as described above, Roku pre-configures/programs each Accused Product to perform the above described steps upon receiving an OTA update from Roku.

46. Further, Roku not only set the time and conditions under which a user could receive and install an OTA update, but Roku required all users to accept and install such updates.

47. For example, Roku stated the following in its 2015 End User License Agreement:

Software Updates

in its sole discretion, ROKU MAY PROVIDE UPDATES TO YOUR PLAYER VIA the internet, including BUG FIXES AND UPDATES, CHANGES IN THE USER INTERFACE OR HOW YOU ACCESS CONTENT, AND OTHER CHANGES THAT MAY add, ALTER or removE functionalities and features. You acknowledge that these updates: (a) may happen automatically in the background at any time (and that they cannot be disabled by You); AND (b) will require aN internet CONNECTION and You may incur additional data charges from the provider of the INTERNECT CONNECTION. You understand that these updates are necessary to maintain compatibility with other updates to ROKU'S products or services and may be required for security reasons. by using the player, you hereby AGREE to receive such updates.

https://docs.roku.com/api/v1/published/deviceplayereula/en/IE/text.

48. Similarly, Roku stated the following in its 2017 Terms of Service:

Software Updates

Roku reserves the right to AUTOMATICALLY update the ROKU SERVICE, including BUG FIXES AND UPDATES, CHANGES IN THE USER INTERFACE OR HOW YOU ACCESS CONTENT, AND OTHER CHANGES THAT MAY add, ALTER or removE functionalities and features. You acknowledge that these updates may happen automatically in the background at any time (and that they cannot be disabled by You). You understand that these updates are necessary to maintain compatibility with other updates to our products or services and may be required for security reasons. by using the ROKU SERVICE, you hereby AGREE to receive such updates.

https://docs.roku.com/published/usertermsandconditions/en/ie.

49. Further, Roku emphasizes the benefits associated with updating the software of its Accused Products, including to "maintain compatibility" with Roku's products and services.

50. Roku also identified the specific benefits associated with each OTA update it provided: https://support.roku.com/article/228844467.

51. Further, Roku controlled the manner in which each OTA update could be performed, including by pre-configuring each Accused Product such that, upon receiving an OTA update from Roku, the device would automatically perform each remaining step of the claimed method.

52. Roku also controlled the timing of the performance of such method by determining when to utilize its OTA servers/software to set up a verification structure in each Accused Product.

53. Roku also had the right and ability to stop or limit infringement simply by not performing the initial step of using its OTA servers/software to set up a verification structure in each Accused Product. Absent this action by Roku, the infringement at issue in this lawsuit would not have occurred.

Nintendo - Ancora Exh. 1077

54. Roku's infringement has caused damage to Ancora, and Ancora is entitled to recover from Roku those damages that Ancora has sustained as a result of Roku's infringement.

DEMAND FOR JURY TRIAL

55. Ancora hereby demands a jury trial for all issues so triable.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff prays for judgment as follows:

A. Declaring that Roku, Inc. has infringed United States Patent No. 6,411,941 in violation of 35 U.S.C. § 271;

B. Awarding damages to Ancora arising out of this infringement, including enhanced damages pursuant to 35 U.S.C. § 284 and prejudgment and post-judgment interest, in an amount according to proof;

C. Awarding such other costs and relief the Court deems just and proper, including any relief that the Court may deem appropriate under 35 U.S.C. § 285.

Date: July 16, 2021

/s/ Andres Healy

Andres Healy (WA 45578)
SUSMAN GODFREY LLP
1201 Third Avenue, Suite 3800
Seattle, Washington 98101
Tel: (206) 516-3880
Fax: 206-516-3883
ahealy@susmangodfrey.com

Lexie G. White (TX 24048876)
SUSMAN GODFREY LLP
1000 Louisiana Street, Suite 5100
Houston, Texas 77002
Tel: (713) 651-9366
Fax: (713) 654-6666
lwhite@susmangodfrey.com

Nintendo - Ancora Exh. 1077

Charles Ainsworth
State Bar No. 00783521
Robert Christopher Bunt
State Bar No. 00787165
PARKER, BUNT & AINSWORTH, P.C.
100 E. Ferguson, Suite 418
Tyler, TX 75702
903/531-3535
E-mail: charley@pbatyler.com
E-mail: rcbunt@pbatyler.com

COUNSEL FOR PLAINTIFF ANCORA TECHNOLOGIES, INC.

Nintendo - Ancora Exh. 1077