SPENCER HOSIE (CA Bar No. 101777)
shosie@hosielaw.com
BRUCE WECKER (CA Bar No. 078530)
bwecker@hosielaw.com
GEORGE F. BISHOP (CA Bar No. 89205)
gbishop@hosielaw.com
DIANE S. RICE (CA Bar No. 118303)
drice@hosielaw.com
HOSIE RICE LLP
188 The Embarcadero, Suite 750
San Francisco, CA 94105
(415) 247-6000 Tel.
(415) 247-6001 Fax

Attorneys for Plaintiff
IMPLICIT NETWORKS, INC.

UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

IMPLICIT NETWORKS, INC., Plaintiff, v. CITRIX SYSTEMS, INC., Defendant.	Case No. 3:10-cv-3766-SI **FIRST AMENDED COMPLAINT AND DEMAND FOR JURY TRIAL**

Plaintiff Implicit Networks, Inc. ("Implicit" or "Plaintiff") hereby files its complaint against defendant Citrix Systems, Inc. ("Citrix" or "Defendant"), for patent infringement. For its complaint, Plaintiff alleges, on personal knowledge as to its own acts and on information and belief as to all other matters, as follows:

PARTIES

1. Implicit is a corporation organized under the laws of the State of Washington, with its principal place of business in Seattle, Washington.

2. Citrix is a corporation organized under the laws of the State of Delaware, with its principal place of business in Fort Lauderdale, Florida.

JURISDICTION AND VENUE

3. This complaint asserts a cause of action for patent infringement under the Patent Act, 35 U.S.C. § 271. This Court has subject matter jurisdiction over this matter by virtue of 28 U.S.C. § 1338(a). Venue is proper in this Court by virtue of 28 U.S.C. § 1391(b) and (c) and 28 U.S.C. § 1400(b), in that Citrix may be found in this district, have committed acts of infringement in this district, and a substantial part of the events or omissions giving rise to the claim occurred and a substantial part of property that is the subject of the action is situated in this district.

4. This Court has personal jurisdiction over Citrix because Defendant has a place of business in, and provides infringing products and services in, the Northern District of California.

INTRADISTRICT ASSIGNMENT

5. Pursuant to Civil LR 3-2(c), this case should be subject to district-wide assignment because it is an Intellectual Property Action.

I. STATEMENT OF FACTS.

A. Implicit's Dynamic Data Flow Patent Family Patents: Implicit's Inventions, Patents, and Products.

1. The Problem Implicit Solved.

6. In the early 1990's, personal computers were stand-alone devices, just like typewriters before them. Consumers would buy shrink-wrapped software applications, such as Lotus Notes or the Berkeley Systems "Flying Toasters" screensaver. They would install the application, the application would run on the computer, and the consumer would use the computer to perform discreet and well-defined tasks, typically turning on data and document processing. Every computer was an island, unique unto itself.

7. All of this changed with the advent of computer networking, *i.e.*, computers hooked together with other computers and, ultimately, other devices entirely. Suddenly, computers had to be able to *talk* to other computers. With networking, computers moved from being standalone devices for running discreet applications to being constituent parts of much larger linked systems.

8. This physical change brought a corresponding change in use and the content itself. Computers became **communication** devices, allowing their users to exchange real-time text (e-mail), interactive files (conferencing), and multi-media (photos; video). With the Internet, hyperlinks, and the World Wide Web, computer users could shop online, create individual web pages (Facebook), watch movies on demand (the new Netflix), and do all the other on-line activities now commonplace. Instead of resources being applied to isolated data on non-networked machines, computers could be linked together and resources applied to data as it flowed from one system to the next. The shift was from processing **data** (spreadsheet; word processing) to processing the **data flow**, *e.g.*, data in transit.

9. This paradigm shift created a host of new problems, however. In the mid-1990's, for example, there were many different media formats (WAV; mpeg; Windows Media Video), each calibrated to do different things and solve different problems; as the richness of what computers could communicate increased, so too did the number of protocols for **how** to communicate. And, along with media formats, there were formats for other forms of content, *e.g.* HTML, X HTML, DHTML, etc…. More, there were numerous network protocols, including point-to-point ("PTP"), SPX and IPX (proprietary protocols for Novell's Network), Apple Talk, Microsoft's NetBEUI, and the telephony RTP standard. There were also different operating systems on computers, *e.g.* Windows versus Mac vs. Linux, along with different devices (phones; computers; PDA's; etc.) with different protocols, needs, and capabilities. It was a three dimensional problem: ***different devices***, with ***different networks***, sending ***different content*** – the "3D" problem.

2. <u>The "Vertical Application" Fix</u>.

10. The first solution to the 3D problem lay in building greater intelligence into the applications themselves. For example, a media player in 1995 had to be able to digest different types of formats (WAV; mpeg), and work on various operating systems, *e.g.* Windows and Mac OS. The developer of the application had to anticipate who would be using the player, and for which devices and content, and then build-in the ability to handle the anticipated demands. In short, the developer had to anticipate **use** and then **configure** the design accordingly.

11. This model led to ever-increasing complexity, cost, and processing overhead. Given that all anticipated uses had to be preconfigured at build-time, any **unanticipated** new use, *e.g.*, a different format or a different device, would simply break the system. The

developer had to have the foresight to specify explicitly all possible configurations in advance, a difficult task in a rapidly changing world.

12. Given these inherent inadequacies, there was a real need for a new and different approach to solve the 3D problem.

3. __Implicit's Solution__.

13. In 1994, Edward Balassanian was a computer scientist working on networking issues at Microsoft. Microsoft was then promoting proprietary protocols and trying to establish a proprietary standard. But, with the ever more diverse set of devices and demands, Mr. Balassanian did not think that a monolithic, one size fits all approach would ultimately work. In February 1995, he left Microsoft.

14. A year later, he founded Implicit Networks, then known as BeComm (hereafter "Implicit").

15. Mr. Balassanian created Implicit to build a radical new approach to networking – a new solution to the 3D problem. Put simply, instead of stacking intelligence into the application, Mr. Balassanian devised a system where every discrete computer function, *e.g.*, processing http server requests over TCP/IP, streaming a video web-based client, or managing voice-over-IP calls, would be built into a discrete software module, called a "bead." Dynamically, at run-time, a software engine would receive a stream of data --- say video --- determine **what** services were necessary to render that content and **where** the content was to be rendered, and then assemble --- or string together --- the requisite service beads (modules) at run-time. In this fashion, the needs at run-time drove the just-in-time creation of the processing path itself, as against trying to stuff given data into a stack previously hardwired into the application.

16. Any specific service could be encapsulated as a bead, including:

- **hardware** such as a video display, speaker, microphone, mouse, Ethernet, etc.
- **protocols** such as TCP/IP, HTTP, SOAP, email (POP3, SMTP), etc.
- **transformational algorithms** such as audio/video decoders, etc.
- **SDK technologies** such as speech-recognition engines (e.g., IBM's ViaVoice), text-to-speech generators, etc.
- **backend services** such as Database, CRM, and Content Management Systems.

17. Ultimately, Implicit built more than 200 discrete software service beads. Beads were the building blocks for the processing element applied to a data flow.

18. In this new model, services were designed from the outset to process data flows. This meant that the intelligence engine picked the right services for the right data flows, managed the "State" (*e.g.* status) associated with each data flow, and managed the flow across the services. In this new system, the Lego blocks needed to process a particular data flow were assembled when needed and as needed, as against the prior model, where the blocks were immutably glued together at build-time.

19. The benefits of this new approach were significant: services were reusable, processing faster and more efficient, and data that required more CPU involvement got it, when and as needed. Mr. Balassanian called this system "Strings," as discrete functions were strung together at run-time.

20. The concept of breaking up applications into discrete services that could be "strung" together on the fly at runtime was an innovation with profound applicability to real world problems. It applied to media players since it allowed media encoding/decoding/transcoding to happen adaptively at runtime. It applied to network stacks since it allowed network stacks to be responsive to real-time changes in the physical network (*e.g.* QoS), transport (*e.g.* support for new protocols), and application layers (*e.g.* virus threats, firewalls etc.).

21. Implicit made and sold products and technology to numerous large and sophisticated customers. Implicit first had its Strings and Beads platform ready for commercial sale in January 2000, at the Consumer Electronics Show ("CES") held that month in Las Vegas. From this date forward, Implicit met with real success in the marketplace. For example, in 2000, Implicit signed a contract to develop all the media processing code for Intel's web tablet, a device very similar to Apple's new iPad. By 2001, Implicit had built the code, and Intel began to manufacture the device.

22. In January 2001, Intel signed a second contract with Implicit, under which Implicit was to build all the software for the Intel equivalent of iTunes. As per this signed contract, Implicit received $850,000, plus a 5% revenue share going forward of all the Intel Consumer Products Division related revenue.

23. In 2004, Intel hired Implicit to use its streaming technology to build the Intel media player, a device that synchronized multiple computers in a home to play music and video, both locally and over a network.

24. Along with these Intel contracts, in 2004, Implicit signed a contract with chip maker AMD to develop a media player referenced design for AMD. Implicit built the media player, using its technology, finishing in 2004.

25. Along the same lines, Thompson Multimedia hired Implicit to build all of the media processing software for the first Thompson digital set-box that allowed for streaming of HD content into the home. The resulting Implicit-Thompson set-box won Best of Show at the annual Consumer Electronics Show ("CES") in 2005.

26. Along the same lines, in 2003, Implicit built a distributed knowledge management solution for Raytheon, using Strings technology. The solution allowed disparate databases to be connected to a single user interface such that data was normalized

on the fly by software components. The system was used as part of a Raytheon product for knowledge discovery in the defense sector.

27. In addition to these specific contractual relationships, Implicit, through its CEO and others, met with numerous large technology companies to introduce them to the novel Implicit technology. These companies included Cisco Systems, 3Com, Motorola, and numerous others. All such technical discussions were conducted pursuant to respective NDA's.

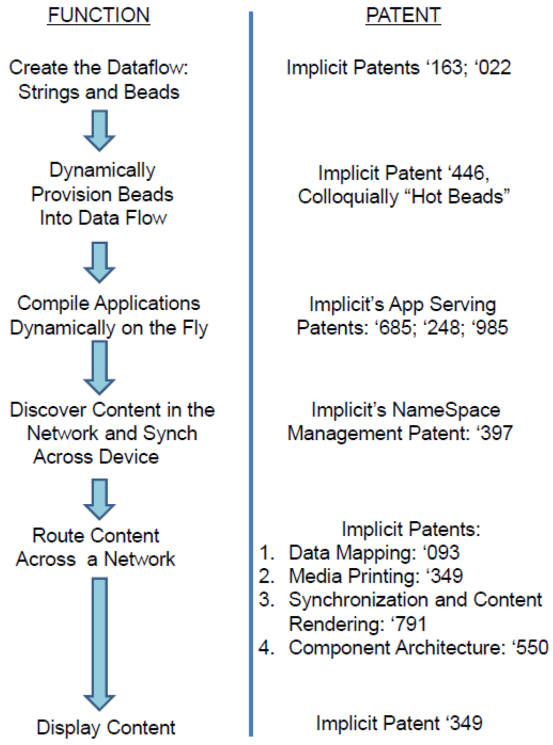
28. Implicit's work, inventions, and patents were the subject of numerous articles in the trade press. For example, in March, 2001, the EETimes reported on Implicit's work with the Intel Tablet, and specifically called out the Implicit patent portfolio, as follows:

> Intel intends to introduce the tablet in North America later this year. One technology that will make the Web Tablet stand out among other Internet appliances is BeComm's Strings. And by extension, Strings could weave disparate distributed appliances into a global peer-to-peer communications architecture.
>
> ***
>
> **Bead-dazzled**
>
> While the Strings core has many similarities to traditional operating systems, it is also significantly different. Strings defines a new middleware layer of software focused on delivering digital media to end users, rather than relying on hardware or networks to deliver that media. To address the fluid nature of Internet appliances, every Strings-based appliance is able to dynamically generate the feature set needed to enable instant access to content. Strings achieves this by leveraging highly discrete software objects called Beads. Any Strings-enabled appliance can instantly string together a series of Beads to dynamically enable the required functionality. Since an appliance can string Beads together across a network of appliances, the functionality required to manage any given type of media can be distributed across a network.

> Strings provides an environment where users have instant access to any type of content from any appliance. For example, a handheld device with a screen, speaker and microphone could provide access any content that can be rendered in audio or video formats. This handheld could morph into an MP3 player, serve as an Internet telephone, or function as a universal remote control. That requires managing not only the appliance's user interface, but also its interface to multimedia content as well, and to the appliance's interface to the network.
>
> **Complete infrastructure**
>
> **<u>To make this possible, Strings leverages a patented technology that allows Beads to be strung together on the fly to provide the precise functionality required by</u>** the end user. Since Beads can encapsulate everything from device drivers and user interface components to multimedia codecs and network protocols, Strings is able to provide a complete infrastructure for intelligent appliances.

Emphasis added.

29. Implicit indeed did patent all of the core aspects of its String architecture. Captured graphically by function, below is the portfolio:


FUNCTION
PATENT
Create the Dataflow: Strings and Beads
Implicit Patents '163; '022
Dynamically Provision Beads Into Data Flow
Implicit Patent '446, Colloquially "Hot Beads"
Compile Applications Dynamically on the Fly
Implicit's App Serving Patents: '685; '248; '985
Discover Content in the Network and Synch Across Device
Implicit's NameSpace Management Patent: '397
Route Content Across a Network
Implicit Patents:
1. Data Mapping: '093
2. Media Printing: '349
3. Synchronization and Content Rendering: '791
4. Component Architecture: '550
Display Content
Implicit Patent '349

30. As particularly germane to this Complaint, on September 30, 2003, United States Patent No. 6,629,163 ("the '163 patent") entitled "Method and System for Demultiplexing a First Sequence of Packet Components to Identify Specific Components Wherein Subsequent Components are Processed Without Re-Identifying Components," was duly and legally issued, and assigned to Plaintiff. On December 18, 2008, the '163 patent was put in re-exam. The '163 patent emerged from re-examination on June 22, 2010, carrying U.S. Patent No. 6,629,163. In its Reasons For Allowance, the PTO called out the novelty of the Implicit Dynamic Data Flow technology. It is assigned to Plaintiff, Implicit. True and correct copies of the '163 patent and the Ex Parte Reexamination Certificate are attached as Exhibit A and Exhibit B.

31. On October 31, 2007, Edward Balassanian filed a continuation application, which on May 4, 2010, issued as U.S. Patent No. 7,711,857 ("'857"). Mr. Balassanian assigned the patent to Implicit and Implicit is the sole owner of the patent. *See* Exhibit C.

B. <u>Citrix's Infringement of the Implicit Dynamic Data Flow Patents</u>.

1. <u>Citrix NetScaler</u>.

32. Citrix NetScaler, a web application delivery appliance, provides various data processing features such as network load balancing, content switching, data compression, and security threat detection, for accelerating the delivery of applications on the Internet. The NetScaler system inspects data packets to determine the type of traffic and performs requisite operations accordingly. NetScaler performs demultiplexing operations such as IP defragmentation, TCP flow reassembly, flow blocking and flow state tracking on incoming/outgoing packets at layer 2-7 of the TCP stack. That is, NetScaler relies on Deep Packet Inspection. As Citrix describes in its NetScaler Policy Configuration and Reference Guide:

> **Benefits of Using Advanced Policies**
>
> Advanced policies use a powerful expression language that is built on a class-object model, and they offer several options that enhance your ability to configure the behavior of various NetScaler features. With advanced policies, you can do the following:
>
> - Perform fine-grained analyses of network traffic from layers 2 through 7.
> - Evaluate any part of the header or body of an HTTP or HTTPS request or response.

33. Citrix NetScaler inspects data packets and compares them to existing policies, defined in the system, to dynamically determine a sequence of actions to be performed on the corresponding data packet flow.

34. NetScaler maintains several policies for every feature it implements. Each policy contains a rule, and an associated response action. Rules are the logical expression based criterion which is used to evaluate (or match) the data packet flows. Policy rules have associated response action which is performed on a data packet flow if it matches the rule. NetScaler evaluates all requests, responses, and other traffic flows on the network against all the policies defined in the system and dynamically determines the sequence of actions to be performed on the data packet flow, based on the matching Rules.

35. NetScaler also performs demultiplexing of data packets by reassembling datagrams fragmented over multiple packets.

36. NetScaler dynamically identifies a sequence of actions to be performed on a data packet flow by analyzing the first packets and then applies the identified processes to all the packets in that flow; in so doing, it infringes the claims of Implicit's Dynamic Data Flow Patents.

2. Citrix's WANScaler/Branch Repeater Products.

37. Citrix's WANScaler/Branch Repeater is a WAN optimization and acceleration platform. It provides high-performance application delivery to branch office users. WANScaler/Branch Repeater is a symmetric solution, wherein a WANScaler/Branch Repeater appliance is deployed both in a central data center as well as in the branch office.

38. WANScaler/Branch Repeater performs TCP optimization by performing actions like Windows scaling, selective acknowledgement and linked optimization, and bandwidth optimization. Specific layer seven protocol optimizations help reduce the number of round trips on chatty protocols such as CIFS over the WAN. The WANScaler system is "built upon the AutoOptimizer Engine, which automatically and dynamically applies to each data flow the best combination of performance boosting techniques depending on the application, the data, and the network conditions." Its transparent architecture "ensur[es] that the system can automatically configure and dynamically tune itself." Data flows are also separately identified to provide Fair Queuing so that bottlenecks in one flow do not impact all the other flows of the same class. "The AutoOptimizer Engine automatically applies the right mix of WAN acceleration techniques based on network conditions, data flows, and application mix - and dynamically tunes the system as these variables change, ensuring optimal network performance at all times."

39. In these actions, WANScaler/Branch Repeater infringes the claims of the Implicit Dynamic Data Flow Patents.

COUNT I

PATENT INFRINGEMENT

40. On September 30, 2003, United States Patent No. 6,629,163 ("the '163 patent") entitled "Method and System for Demultiplexing a First Sequence of Packet Components to Identify Specific Components Wherein Subsequent Components are

Processed Without Re-Identifying Components" was duly and legally issued. A true and correct copy of the '163 patent is attached as Exhibit A. On June 22, 2010, an Ex Parte Reexamination Certificate was duly and legally issued. A true and correct copy of the Reexamination Certificate is attached as Exhibit B.

41. On May 4, 2010, a continuation patent, United States Patent No. http://patft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.htm&r=1&f=G&l=50&s1=7711857.PN.&OS=PN/7711857&RS=PN/7711857 - h0#h0http://patft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.htm&r=1&f=G&l=50&s1=7711857.PN.&OS=PN/7711857&RS=PN/7711857 - h2#h27,711,857 ("the '857 patent") entitled "Method and System for Data Demultiplexing" was duly and legally issued. A true and correct copy of the '857 patent is attached as Exhibit C.

42. Edward Balassanian is the sole inventor of the '163 and '857 patents (collectively "Patents-in-Suit"). The Patents-in-Suit have been assigned to Plaintiff. Plaintiff Implicit is the sole legal and rightful owner of the Patents-in-Suit.

43. Citrix makes, uses, and sells products that infringe the Patents-in-Suit, such products including, NetScaler and WANScaler/Branch Repeater, as alleged above.

44. In addition, Citrix has infringed and is still infringing the Patents-in-Suit in this country, through, *inter alia*, its active inducement of others to make, use, and/or sell the systems, products and methods claimed in one or more claims of the patents. Citrix's customers of NetScaler and WANScaler/Branch Repeater products directly infringed the Patents-in-Suit, and were induced to do so by Citrix. Citrix knows of the Patents-in-Suit and

their contents, based upon, *inter alia*, Citrix's actual notice of the patents. Citrix actively and knowingly encouraged, aided and abetted its customers to directly infringe the Patents-in-Suit. Citrix offered its infringing products for sale with the intent of promoting their use to infringe, and with that object, Citrix intentionally encouraged its customers to infringe the Patents-in-Suit by advertising its products for infringing uses, and instructing its customers how to use the products to engage in infringement. Citrix specifically intended that its customers infringe the Patents-in-Suit. Citrix knew of the Patents-in-Suit and of their contents, based upon, its actual notice of the patents. Citrix had specific intent to encourage customers to infringe the Patents-in-Suit, and knew or should have known that its actions would encourage customers to actually infringe the Patents-in-Suit. This conduct constitutes infringement under 35 U.S.C. § 271(b).

45. In addition, Citrix has infringed and is still infringing the Patents-in-Suit in this country through, *inter alia*, providing and selling goods and services including the infringing NetScaler and WANScaler/Branch Repeater products designed for use in practicing one or more claims of the Patents-in-Suit, where the goods and services constitute a material part of the invention and are not staple articles of commerce, and which have no use other than infringing one or more claims of the Patents-in-Suit. Citrix's customers commit the entire act of direct infringement. Citrix has committed these acts with knowledge that the goods and services it provides are specially made for use in a manner that directly infringes the Patents-in-Suit. This conduct constitutes infringement under 35 U.S.C. § 271(c).

46. As a result of the infringement by Citrix, Plaintiff has been damaged, and will continue to be damaged, until this Defendant is enjoined from further acts of infringement.

47. Citrix will continue to infringe unless enjoined by this Court. Plaintiff faces real, substantial and irreparable damage and injury of a continuing nature from infringement for which Plaintiff has no adequate remedy at law.

WHEREFORE, Plaintiff prays for entry of judgment:

A. that the Patents-in-Suit patent are valid and enforceable;

B. that Defendant has infringed one or more claims of the Patents-in-Suit;

C. that Defendant account for and pay to Plaintiff all damages caused by the infringement of the Patents-in-Suit, which by statute can be no less than a reasonable royalty;

D. that Plaintiff be granted pre-judgment and post-judgment interest on the damages caused to them by reason of Defendant's infringement of the Patents-in-Suit;

E. that this Court require Defendant to file with this Court, within thirty (30) days after entry of final judgment, a written statement under oath setting forth in detail the manner in which Defendant has complied with the injunction;

F. that this be adjudged an exceptional case and the Plaintiff be awarded its attorney's fees in this action pursuant to 35 U.S.C. § 285;

G. that this Court award Plaintiff its costs and disbursements in this civil action, including reasonable attorney's fees; and

H. that Plaintiff be granted such other and further relief as the Court may deem just and proper under the current circumstances.

Dated: December 1, 2010

Respectfully submitted,

/s/ Spencer Hosie
SPENCER HOSIE (CA Bar No. 101777)
shosie@hosielaw.com

BRUCE WECKER (CA Bar No. 078530)
bwecker@hosielaw.com
GEORGE F. BISHOP (CA Bar No. 89205)
gbishop@hosielaw.com
DIANE S. RICE (CA Bar No. 118303)
drice@hosielaw.com
HOSIE RICE LLP
188 The Embarcadero, Suite 750
San Francisco, CA 94105
(415) 247-6000 Tel.
(415) 247-6001 Fax

Attorneys for Plaintiff
IMPLICIT NETWORKS, INC.

DEMAND FOR JURY TRIAL

Plaintiff, by its undersigned attorneys, demands a trial by jury on all issues so triable.

Dated: December 1, 2010

Respectfully submitted,

/s/ Spencer Hosie
SPENCER HOSIE (CA Bar No. 101777)
shosie@hosielaw.com
BRUCE WECKER (CA Bar No. 078530)
bwecker@hosielaw.com
GEORGE F. BISHOP (CA Bar No. 89205)
gbishop@hosielaw.com
DIANE S. RICE (CA Bar No. 118303)
drice@hosielaw.com
HOSIE RICE LLP
188 The Embarcadero, Suite 750
San Francisco, CA 94105
(415) 247-6000 Tel.
(415) 247-6001 Fax

Attorneys for Plaintiff
IMPLICIT NETWORKS, INC.