UNITED STATES DISTRICT COURT
DISTRICT OF DELAWARE

REALTIME DATA LLC d/b/a IXO, Plaintiff, v. PURE STORAGE, INC., Defendant.	C.A. No. 17-cv-01544-GMS **JURY TRIAL DEMANDED**

FIRST AMEMDED COMPLAINT FOR PATENT INFRINGEMENT AGAINST PURE STORAGE, INC.

This is an action for patent infringement arising under the Patent Laws of the United States of America, 35 U.S.C. § 1 *et seq.* in which Plaintiff Realtime Data LLC d/b/a IXO ("Plaintiff," "Realtime," or "IXO") makes the following allegations against Defendant Pure Storage, Inc. ("Pure Storage" or "Defendant"):

PARTIES

1. Realtime is a limited liability company organized under the laws of the State of New York. Realtime has places of business at 5851 Legacy Circle, Plano, Texas 75024, 1828 E.S.E. Loop 323, Tyler, Texas 75701, and 66 Palmer Avenue, Suite 27, Bronxville, NY 10708. Since the 1990s, Realtime has researched and developed specific solutions for data compression, including, for example, those that increase the speeds at which data can be stored and accessed. As recognition of its innovations rooted in this technological field, Realtime holds 50 United States patents and has numerous pending patent applications. Realtime has licensed patents in this portfolio to many of the world's leading technology companies. The patents-in-suit relate to Realtime's development of

advanced systems and methods for fast and efficient data compression using numerous innovative compression techniques based on, for example, particular attributes of the data.

2. On information and belief, Pure Storage is a Delaware corporation with its principal place of business at 650 Castro Street, Mountain View, California 94041. Pure Storage can be served through its registered agent, Corporation Services Company, 251 Little Falls Drive, Wilmington, Delaware 19808.

JURISDICTION AND VENUE

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

4. This Court has personal jurisdiction over Defendant Pure Storage in this action because Pure Storage is incorporated in Delaware and has committed acts within the District of Delaware giving rise to this action and has established minimum contacts with this forum such that the exercise of jurisdiction over Pure Storage would not offend traditional notions of fair play and substantial justice. Pure Storage, directly and through subsidiaries or intermediaries, has committed and continues to commit acts of infringement in this District by, among other things, offering to sell and selling products and/or services that infringe the asserted patents.

5. Venue is proper in this district under 28 U.S.C. § 1400(b). Upon information and belief, Pure Storage is incorporated in Delaware, has transacted business in the District of Delaware, and has committed acts of direct and indirect infringement in the District of Delaware.

COUNT I
INFRINGEMENT OF U.S. PATENT NO. 9,054,728

6. Plaintiff realleges and incorporates by reference paragraphs 1-5 above, as if fully set forth herein.

7. Plaintiff Realtime is the owner by assignment of United States Patent No. 9,054,728 ("the '728 Patent") entitled "Data compression systems and methods." The '728 Patent was duly and legally issued by the United States Patent and Trademark Office on June 9, 2015. A true and correct copy of the '728 Patent is included as Exhibit A.

8. On information and belief, Pure Storage has offered for sale, sold and/or imported into the United States Pure Storage products and services that infringe the '728 patent, and continues to do so. By way of illustrative example, these infringing products and services include, without limitation, Pure Storage's products and services, *e.g.*, Purity Reduce, FlashArray M10, FlashArray M20, FlashArray M50, FlashArray M70, and FlashArray X70, and the system hardware on which they operate, and all versions and variations thereof since the issuance of the '728 Patent ("Accused Instrumentalities").

9. On information and belief, Pure Storage has directly infringed and continues to infringe the '728 Patent, for example, by making, selling, offering for sale, and/or importing the Accused Instrumentalities, and through its own use and testing of the Accused Instrumentalities, which constitute systems for compressing data claimed by Claim 1 of the '728 Patent, comprising: a processor; one or more content dependent data compression encoders; and a single data compression encoder; wherein the processor is configured: to analyze data within a data block to identify one or more parameters or attributes of the data wherein the analyzing of the data within the data block to identify

the one or more parameters or attributes of the data excludes analyzing based solely on a descriptor that is indicative of the one or more parameters or attributes of the data within the data block; to perform content dependent data compression with the one or more content dependent data compression encoders if the one or more parameters or attributes of the data are identified; and to perform data compression with the single data compression encoder, if the one or more parameters or attributes of the data are not identified. Upon information and belief, Pure Storage uses the Accused Instrumentalities, which are infringing systems, for its own internal non-testing business purposes, while testing the Accused Instrumentalities, and while providing technical support and repair services for the Accused Instrumentalities to Pure Storage's customers.

10. On information and belief, Pure Storage has had knowledge of the '728 Patent since at least the filing of the original Complaint in this action, or shortly thereafter, and on information and belief, Pure Storage knew of the '728 Patent and knew of its infringement, including by way of this lawsuit.

11. Pure Storage's affirmative acts of making, using, selling, offering for sale, and/or importing the Accused Instrumentalities have induced and continue to induce users of the Accused Instrumentalities to use the Accused Instrumentalities in their normal and customary way on compatible systems to infringe Claim 1 of the '728 Patent, knowing that when the Accused Instrumentalities are used in their ordinary and customary manner with such compatible systems, such systems constitute infringing systems for compressing data comprising; a processor; one or more content dependent data compression encoders; and a single data compression encoder; wherein the processor is configured: to analyze data within a data block to identify one or more

parameters or attributes of the data wherein the analyzing of the data within the data block to identify the one or more parameters or attributes of the data excludes analyzing based solely on a descriptor that is indicative of the one or more parameters or attributes of the data within the data block; to perform content dependent data compression with the one or more content dependent data compression encoders if the one or more parameters or attributes of the data are identified; and to perform data compression with the single data compression encoder, if the one or more parameters or attributes of the data are not identified. For example, Pure Storage explains to customers the benefits of using the Accused Instrumentalities, such as by touting their performance advantages: "At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss. Best of all, these savings are delivered without requiring any tuning." *See* https://www.purestorage.com/resources/glossary/data-deduplication.html. For similar reasons, Pure Storage also induces its customers to use the Accused Instrumentalities to infringe other claims of the '728 Patent. Pure Storage specifically intended and was aware that the normal and customary use of the Accused Instrumentalities on compatible systems would infringe the '728 Patent. Pure Storage performed the acts that constitute induced infringement, and would induce actual infringement, with the knowledge of the '728 Patent and with the knowledge, or willful blindness to the probability, that the induced acts would constitute infringement. On information and belief, Pure Storage engaged in such inducement to promote the sales of the Accused Instrumentalities, *e.g.*, through Pure Storage's user manuals, product support, marketing materials, and training

materials to actively induce the users of the accused products to infringe the ’728 Patent. Accordingly, Pure Storage has induced and continues to induce end users of the accused products to use the accused products in their ordinary and customary way with compatible systems to make and/or use systems infringing the ’728 Patent, knowing that such use of the Accused Instrumentalities with compatible systems will result in infringement of the ’728 Patent.

12. The Accused Instrumentalities include a system for compressing data, comprising a processor. For example, main board for a FlashArray controller includes a “12-core processor complex that runs the Purity software … .” FlashArray User's Guide, at 17.

13. The Accused Instrumentalities include a system for compressing data, comprising one or more content dependent data compression encoders. For example, the Accused Instrumentalities perform block-level deduplication, which is a content dependent data compression encoder. *See, e.g.*, https://www.purestorage.com/resources/glossary/data-deduplication.html (“At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss. Best of all, these savings are delivered without requiring any tuning.”); FlashArray User's Guide, at 36 (“For each sector of data that enters an array, Purity computes a hash checksum which it compares against the checksums of already-stored sectors. If it finds a match, the software reads the stored sector and compares it with the new one to eliminate the possibility of ‘hash collisions’. Purity replaces duplicate blocks with pointers to the single

copy of their contents"). Performing deduplication results in compression by representing data with fewer bits.

14. The Accused Instrumentalities comprise a single data compression encoder. *See, e.g.,* FlashArray User's Guide, at 36 ("Purity attempts to compress the data in blocks that remain after pattern elimination and deduplication, choosing among several well-known compression algorithms that balance compression speed against compactness of the result. The software stores compressed rather than original host-written data in NVRAM, in write unit buffers, and ultimately on solid state storage."); https://www.purestorage.com/products/purity/purity-reduce.html ("Inline compression reduces data to use less capacity than the original format. Append-only write layout and variable addressing optimize compression savings by removing the wasted space that fixed-block architectures introduce. Combined with Deep Reduction, compression delivers 2 - 4x data reduction, and is the primary form of data reduction for databases.").

15. The Accused Instrumentalities analyze data within a data block to identify one or more parameters or attributes of the data, for example, whether the data is duplicative of data previously transmitted and/or stored, where the analysis does not rely only on the descriptor. *See, e.g.,* https://www.purestorage.com/resources/glossary/data-deduplication.html ("At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss. Best of all, these savings are delivered without requiring any tuning."); FlashArray User's Guide, at 36 ("For each sector of data that enters an array, Purity computes a hash checksum which it compares against the checksums of already-stored sectors. If it finds a

match, the software reads the stored sector and compares it with the new one to eliminate the possibility of 'hash collisions'. Purity replaces duplicate blocks with pointers to the single copy of their contents").

16. The Accused Instrumentalities perform content dependent data compression with the one or more content dependent data compression encoders if the one or more parameters or attributes of the data are identified. *See, e.g.*, https://www.purestorage.com/resources/glossary/data-deduplication.html ("At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss. Best of all, these savings are delivered without requiring any tuning."); FlashArray User's Guide, at 36 ("For each sector of data that enters an array, Purity computes a hash checksum which it compares against the checksums of already-stored sectors. If it finds a match, the software reads the stored sector and compares it with the new one to eliminate the possibility of 'hash collisions'. Purity replaces duplicate blocks with pointers to the single copy of their contents").

17. The Accused Instrumentalities perform data compression with the single data compression encoder, if the one or more parameters or attributes of the data are not identified. *See, e.g.*, FlashArray User's Guide, at 36 ("Purity attempts to compress the data in blocks that remain after pattern elimination and deduplication, choosing among several well-known compression algorithms that balance compression speed against compactness of the result. The software stores compressed rather than original host-written data in NVRAM, in write unit buffers, and ultimately on solid state storage.");

https://www.purestorage.com/products/purity/purity-reduce.html ("Inline compression reduces data to use less capacity than the original format. Append-only write layout and variable addressing optimize compression savings by removing the wasted space that fixed-block architectures introduce. Combined with Deep Reduction, compression delivers 2 - 4x data reduction, and is the primary form of data reduction for databases.").

18. Pure Storage also infringes other claims of the '728 Patent, directly and through inducing infringement and contributory infringement, for similar reasons as explained above with respect to Claim 1 of the '728 Patent.

19. On information and belief, use of the Accused Instrumentalities in their ordinary and customary fashion results in infringement of the methods claimed by the '728 Patent.

20. By making, using, offering for sale, selling and/or importing into the United States the Accused Instrumentalities, and touting the benefits of using the Accused Instrumentalities' compression features, Pure Storage has injured Realtime and is liable to Realtime for infringement of the '728 Patent pursuant to 35 U.S.C. § 271.

21. As a result of Pure Storage's infringement of the '728 Patent, Plaintiff Realtime is entitled to monetary damages in an amount adequate to compensate for Pure Storage's infringement, but in no event less than a reasonable royalty for the use made of the invention by Pure Storage, together with interest and costs as fixed by the Court.

COUNT II
INFRINGEMENT OF U.S. PATENT NO. 9,667,751

22. Plaintiff realleges and incorporates by reference paragraphs 1-21 above, as if fully set forth herein.

23. Plaintiff Realtime is the owner by assignment of United States Patent No. 9,667,751 ("the '751 Patent") entitled "Data feed acceleration." The '751 Patent was duly and legally issued by the United States Patent and Trademark Office on May 30, 2017. A true and correct copy of the '751 Patent is included as Exhibit B.

24. On information and belief, Pure Storage has offered for sale, sold and/or imported into the United States Pure Storage products and services that infringe the '751 Patent, and continues to do so. By way of illustrative example, these infringing products and services include, without limitation, Pure Storage's products and services, *e.g.*, Purity Reduce, FlashArray M10, FlashArray M20, FlashArray M50, FlashArray M70, and FlashArray X70, and the system hardware on which they operate, and all versions and variations thereof since the issuance of the '751 Patent ("Accused Instrumentalities").

25. On information and belief, Pure Storage has directly infringed and continues to infringe the '751 Patent, for example, through its own use and testing of the Accused Instrumentalities, which in the ordinary course of their operation form a system for compressing data claimed by Claim 25 of the '751 Patent, including: a data server implemented on one or more processors and one or more memory systems; the data server configured to analyze content of a data block to identify a parameter, attribute, or value of the data block that excludes analysis based solely on reading a descriptor; the data server configured to select an encoder associated with the identified parameter, attribute, or value; the data server configured to compress data in the data block with the selected encoder to produce a compressed data block, wherein the compression utilizes a state machine; and the data server configured to store the compressed data block; wherein the time of the compressing the data block and the storing the compressed data block is

less than the time of storing the data block in uncompressed form. Upon information and belief, Pure Storage uses the Accused Instrumentalities, which are infringing systems, for its own internal non-testing business purposes, while testing the Accused Instrumentalities, and while providing technical support and repair services for the Accused Instrumentalities to Pure Storage's customers.

26. On information and belief, Pure Storage has had knowledge of the '751 Patent since at least the filing of the original Complaint in this action, or shortly thereafter, and on information and belief, Pure Storage knew of the '751 Patent and knew of its infringement, including by way of this lawsuit.

27. Upon information and belief, Pure Storage's affirmative acts of making, using, and selling the Accused Instrumentalities, and providing implementation services and technical support to users of the Accused Instrumentalities, have induced and continue to induce users of the Accused Instrumentalities to use them in their normal and customary way to infringe Claim 25 of the '751 Patent by making or using a data server implemented on one or more processors and one or more memory systems; the data server configured to analyze content of a data block to identify a parameter, attribute, or value of the data block that excludes analysis based solely on reading a descriptor; the data server configured to select an encoder associated with the identified parameter, attribute, or value; the data server configured to compress data in the data block with the selected encoder to produce a compressed data block, wherein the compression utilizes a state machine; and the data server configured to store the compressed data block; wherein the time of the compressing the data block and the storing the compressed data block is less than the time of storing the data block in uncompressed form. For example, Pure

Storage explains to customers the benefits of using the Accused Instrumentalities, such as by touting their efficiency: "At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss. Best of all, these savings are delivered without requiring any tuning." *See* https://www.purestorage.com/resources/glossary/data-deduplication.html. For similar reasons, Pure Storage also induces its customers to use the Accused Instrumentalities to infringe other claims of the '751 Patent. Pure Storage specifically intended and was aware that these normal and customary activities would infringe the '751 Patent. Pure Storage performed the acts that constitute induced infringement, and would induce actual infringement, with the knowledge of the '751 Patent and with the knowledge, or willful blindness to the probability, that the induced acts would constitute infringement. On information and belief, Pure Storage engaged in such inducement to promote the sales of the Accused Instrumentalities. Accordingly, Pure Storage has induced and continues to induce users of the accused products to use the accused products in their ordinary and customary way to infringe the '751 Patent, knowing that such use constitutes infringement of the '751 Patent.

28. The Accused Instrumentalities include a system for compressing data. *See, e.g.,* https://www.purestorage.com/resources/glossary/data-deduplication.html ("At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss. Best of all, these savings are delivered without requiring any tuning."); FlashArray User's Guide, at 36

("For each sector of data that enters an array, Purity computes a hash checksum which it compares against the checksums of already-stored sectors. If it finds a match, the software reads the stored sector and compares it with the new one to eliminate the possibility of 'hash collisions'. Purity replaces duplicate blocks with pointers to the single copy of their contents"); FlashArray User's Guide, at 36 ("Purity attempts to compress the data in blocks that remain after pattern elimination and deduplication, choosing among several well-known compression algorithms that balance compression speed against compactness of the result. The software stores compressed rather than original host-written data in NVRAM, in write unit buffers, and ultimately on solid state storage."); https://www.purestorage.com/products/purity/purity-reduce.html ("Inline compression reduces data to use less capacity than the original format. Append-only write layout and variable addressing optimize compression savings by removing the wasted space that fixed-block architectures introduce. Combined with Deep Reduction, compression delivers 2 - 4x data reduction, and is the primary form of data reduction for databases.").

29. The Accused Instrumentalities include a data server implemented on one or more processors and one or more memory systems. For example, main board for a FlashArray controller includes a "12-core processor complex that runs the Purity software … ." FlashArray User's Guide, at 17. The Accused Instrumentalities also use one or more memory systems, including solid state drives (SSDs). *See, e.g.*, FlashArray User's Guide, at ix. ("SSDs mounted in each storage shelf. An array's first two shelves contain 22 SSDs, with two bays reserved for NVRAMs; each additional shelf contains 24

SSDs."). On information and belief, all of the Accused Instrumentalities use one or more memory systems in substantially the same way.

30. The Accused Instrumentalities include a data server configured to analyze content of a data block to identify a parameter, attribute, or value of the data block that excludes analysis based solely on reading a descriptor. *See, e.g.*, https://www.purestorage.com/resources/glossary/data-deduplication.html ("At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss. Best of all, these savings are delivered without requiring any tuning."); FlashArray User's Guide, at 36 ("For each sector of data that enters an array, Purity computes a hash checksum which it compares against the checksums of already-stored sectors. If it finds a match, the software reads the stored sector and compares it with the new one to eliminate the possibility of 'hash collisions'. Purity replaces duplicate blocks with pointers to the single copy of their contents").

31. The Accused Instrumentalities include a data server configured to select an encoder associated with the identified parameter, attribute, or value. For example, the Accused Instrumentalities select between deduplication or other compression. *See, e.g.*, https://www.purestorage.com/resources/glossary/data-deduplication.html ("At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss. Best of all, these savings are delivered without requiring any tuning."); FlashArray User's Guide, at 36

("For each sector of data that enters an array, Purity computes a hash checksum which it compares against the checksums of already-stored sectors. If it finds a match, the software reads the stored sector and compares it with the new one to eliminate the possibility of 'hash collisions'. Purity replaces duplicate blocks with pointers to the single copy of their contents"); FlashArray User's Guide, at 36 ("Purity attempts to compress the data in blocks that remain after pattern elimination and deduplication, choosing among several well-known compression algorithms that balance compression speed against compactness of the result. The software stores compressed rather than original host-written data in NVRAM, in write unit buffers, and ultimately on solid state storage."); https://www.purestorage.com/products/purity/purity-reduce.html ("Inline compression reduces data to use less capacity than the original format. Append-only write layout and variable addressing optimize compression savings by removing the wasted space that fixed-block architectures introduce. Combined with Deep Reduction, compression delivers 2 - 4x data reduction, and is the primary form of data reduction for databases.").

32. The Accused Instrumentalities include a data server configured to compress data in the data block with the selected encoder to produce a compressed data block, wherein the compression utilizes a state machine. *See, e.g.*, FlashArray User's Guide, at 36 ("Purity attempts to compress the data in blocks that remain after pattern elimination and deduplication, choosing among several well-known compression algorithms that balance compression speed against compactness of the result. The software stores compressed rather than original host-written data in NVRAM, in write unit buffers, and ultimately on solid state storage.");

https://www.purestorage.com/products/purity/purity-reduce.html ("Inline compression reduces data to use less capacity than the original format. Append-only write layout and variable addressing optimize compression savings by removing the wasted space that fixed-block architectures introduce. Combined with Deep Reduction, compression delivers 2 - 4x data reduction, and is the primary form of data reduction for databases.").

33. The Accused Instrumentalities include a data server configured to store the compressed data block. For example, the Accused Instrumentalities have storage devices, such as SSDs, that are managed by controllers. *See, e.g.*, FlashArray User's Guide, at ix ("SSDs mounted in each storage shelf. An array's first two shelves contain 22 SSDs, with two bays reserved for NVRAMs; each additional shelf contains 24 SSDs."); FlashArray User's Guide, at 17 ("Main board: Contains the 12-core processor complex that runs the Purity software, DRAM used to hold Purity code and for data buffering and staging."). On information and belief, all of the Accused Instrumentalities include a data server configured to store the compressed data block in substantially the same way.

34. The time of the compressing the data block and the storing the compressed data block in the Accused Instrumentalities is less than the time of storing the data block in uncompressed form. Due to the data reduction and acceleration features of the specific compression algorithms used, the time of the compressing the data block and the storing the compressed data block is less than the time of storing the data block in uncompressed form. *See, e.g.*, https://www.purestorage.com/resources/glossary/data-deduplication.html ("At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss. Best of all, these

savings are delivered without requiring any tuning."); FlashArray User's Guide, at 36 ("For each sector of data that enters an array, Purity computes a hash checksum which it compares against the checksums of already-stored sectors. If it finds a match, the software reads the stored sector and compares it with the new one to eliminate the possibility of 'hash collisions'. Purity replaces duplicate blocks with pointers to the single copy of their contents"); FlashArray User's Guide, at 36 ("Purity attempts to compress the data in blocks that remain after pattern elimination and deduplication, choosing among several well-known compression algorithms that balance compression speed against compactness of the result. The software stores compressed rather than original host-written data in NVRAM, in write unit buffers, and ultimately on solid state storage."); https://www.purestorage.com/products/purity/purity-reduce.html ("Inline compression reduces data to use less capacity than the original format. Append-only write layout and variable addressing optimize compression savings by removing the wasted space that fixed-block architectures introduce. Combined with Deep Reduction, compression delivers 2 - 4x data reduction, and is the primary form of data reduction for databases.").

35. On information and belief, Pure Storage also infringes, directly and through induced infringement, and continues to infringe other claims of the '751 Patent, for similar reasons as explained above with respect to Claim 25 of the '751 Patent.

36. On information and belief, use of the Accused Instrumentalities in their ordinary and customary fashion results in infringement of the methods claimed by the '751 Patent.

37. By making, using, offering for sale, selling and/or importing into the United States the Accused Instrumentalities, and touting the benefits of using the Accused Instrumentalities' compression features, Pure Storage has injured Realtime and is liable to Realtime for infringement of the '751 Patent pursuant to 35 U.S.C. § 271.

38. As a result of Pure Storage's infringement of the '751 Patent, Plaintiff Realtime is entitled to monetary damages in an amount adequate to compensate for Pure Storage's infringement, but in no event less than a reasonable royalty for the use made of the invention by Pure Storage, together with interest and costs as fixed by the Court.

COUNT III
INFRINGEMENT OF U.S. PATENT NO. 8,717,203

39. Plaintiff realleges and incorporates by reference paragraphs 1-38 above, as if fully set forth herein.

40. Plaintiff Realtime is the owner by assignment of United States Patent No. 8,717,203 ("the '203 Patent") entitled "Data compression systems and methods." The '203 Patent was duly and legally issued by the United States Patent and Trademark Office on May 6, 2014. A true and correct copy of the '203 Patent is included as Exhibit C.

41. On information and belief, Pure Storage has offered for sale, sold and/or imported into the United States Pure Storage products and services that infringe the '203 Patent, and continues to do so. By way of illustrative example, these infringing products and services include, without limitation, Pure Storage's products and services, *e.g.*, Purity Reduce, FlashArray M10, FlashArray M20, FlashArray M50, FlashArray M70, and FlashArray X70, and the system hardware on which they operate, and all versions and variations thereof since the issuance of the '203 Patent ("Accused Instrumentalities").

42. On information and belief, Pure Storage has directly infringed and continues to infringe the ’203 Patent, for example, through its own use and testing of the Accused Instrumentalities, which in the ordinary course of their operation form a system, claimed by Claim 14 of the ’203 Patent, for decompressing one or more compressed data blocks included in one or more data packets using a data decompression engine, the one or more data packets being transmitted in sequence from a source that is internal or external to the data decompression engine, wherein a data packet from among the one or more data packets comprises a header containing control information followed by one or more compressed data blocks of the data packet. The claimed system includes: a data decompression processor configured to analyze the data packet to identify one or more recognizable data tokens associated with the data packet, the one or more recognizable data identifying a selected encoder used to compress one or more data blocks to provide the one or more compressed data blocks, the encoder being selected based on content of the one or more data blocks on which a compression algorithm was applied; one or more decompression decoders configured to decompress a compressed data block from among the one or more compressed data blocks associated with the data packet based on the one or more recognizable data tokens; wherein: the one or more decompression decoders are further configured to decompress the compressed data block utilizing content dependent data decompression to provide a first decompressed data block when the one or more recognizable data tokens indicate that the data block was encoded utilizing content dependent data compression; and the one or more decompression decoders are further configured to decompress the compressed data block utilizing content independent data decompression to provide a second decompressed data block when the one or more

recognizable data tokens indicate that the data block was encoded utilizing content independent data compression; and an output interface, coupled to the data decompression engine, configured to output a decompressed data packet including the first or the second decompressed data block. Upon information and belief, Pure Storage uses the Accused Instrumentalities, which are infringing systems, for its own internal non-testing business purposes, while testing the Accused Instrumentalities, and while providing technical support and repair services for the Accused Instrumentalities to Pure Storage's customers.

43. On information and belief, Pure Storage has had knowledge of the '203 Patent since at least the filing of the original Complaint in this action, or shortly thereafter, and on information and belief, Pure Storage knew of the '203 Patent and knew of its infringement, including by way of this lawsuit.

44. Upon information and belief, Pure Storage's affirmative acts of making, using, and selling the Accused Instrumentalities, and providing implementation services and technical support to users of the Accused Instrumentalities, have induced and continue to induce users of the Accused Instrumentalities to use them in their normal and customary way to infringe Claim 14 of the '203 Patent by making or using a system for decompressing, one or more compressed data blocks included in one or more data packets using a data decompression engine, the one or more data packets being transmitted in sequence from a source that is internal or external to the data decompression engine, wherein a data packet from among the one or more data packets comprises a header containing control information followed by one or more compressed data blocks of the data packet the system claimed by Claim 14 of the '203 Patent,

including: a data decompression processor configured to analyze the data packet to identify one or more recognizable data tokens associated with the data packet, the one or more recognizable data identifying a selected encoder used to compress one or more data blocks to provide the one or more compressed data blocks, the encoder being selected based on content of the one or more data blocks on which a compression algorithm was applied; one or more decompression decoders configured to decompress a compressed data block from among the one or more compressed data blocks associated with the data packet based on the one or more recognizable data tokens; wherein: the one or more decompression decoders are further configured to decompress the compressed data block utilizing content dependent data decompression to provide a first decompressed data block when the one or more recognizable data tokens indicate that the data block was encoded utilizing content dependent data compression; and the one or more decompression decoders are further configured to decompress the compressed data block utilizing content independent data decompression to provide a second decompressed data block when the one or more recognizable data tokens indicate that the data block was encoded utilizing content independent data compression; and an output interface, coupled to the data decompression engine, configured to output a decompressed data packet including the first or the second decompressed data block. For example, Pure Storage explains to customers the benefits of using the Accused Instrumentalities, such as by touting their performance advantages: "At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss. Best of all, these savings are delivered without requiring

any tuning." *See* https://www.purestorage.com/resources/glossary/data-deduplication.html. For similar reasons, Pure Storage also induces its customers to use the Accused Instrumentalities to infringe other claims of the '203 Patent. Pure Storage specifically intended and was aware that these normal and customary activities would infringe the '203 Patent. Pure Storage performed the acts that constitute induced infringement, and would induce actual infringement, with the knowledge of the '203 Patent and with the knowledge, or willful blindness to the probability, that the induced acts would constitute infringement. On information and belief, Pure Storage engaged in such inducement to promote the sales of the Accused Instrumentalities. Accordingly, Pure Storage has induced and continues to induce users of the accused products to use the accused products in their ordinary and customary way to infringe the '203 Patent, knowing that such use constitutes infringement of the '203 Patent.

45. The Accused Instrumentalities form a system for decompressing one or more compressed data blocks included in one or more data packets using a data decompression engine, the one or more data packets being transmitted in sequence from a source that is internal or external to the data decompression engine. The Accused Instrumentalities utilize multiple formats of compression to compress data for backup. *See, e.g.,* https://www.purestorage.com/resources/glossary/data-deduplication.html ("At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss. Best of all, these savings are delivered without requiring any tuning."); FlashArray User's Guide, at 36 ("For each sector of data that enters an array, Purity computes a hash checksum which it

compares against the checksums of already-stored sectors. If it finds a match, the software reads the stored sector and compares it with the new one to eliminate the possibility of 'hash collisions'. Purity replaces duplicate blocks with pointers to the single copy of their contents"); FlashArray User's Guide, at 36 ("Purity attempts to compress the data in blocks that remain after pattern elimination and deduplication, choosing among several well-known compression algorithms that balance compression speed against compactness of the result. The software stores compressed rather than original host-written data in NVRAM, in write unit buffers, and ultimately on solid state storage."); https://www.purestorage.com/products/purity/purity-reduce.html ("Inline compression reduces data to use less capacity than the original format. Append-only write layout and variable addressing optimize compression savings by removing the wasted space that fixed-block architectures introduce. Combined with Deep Reduction, compression delivers 2 - 4x data reduction, and is the primary form of data reduction for databases."). To recover data from backup, the Accused Instrumentalities decompress the data.

46. The data packets from among the one or more data packets in the Accused Instrumentalities include a header containing control information followed by one or more compressed data blocks of the data packet. The header containing control information contains information used to determine which compression format was used to compress the data. *See, e.g.*, https://www.purestorage.com/resources/glossary/data-deduplication.html ("At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss.

Best of all, these savings are delivered without requiring any tuning."); FlashArray User's Guide, at 36 ("For each sector of data that enters an array, Purity computes a hash checksum which it compares against the checksums of already-stored sectors. If it finds a match, the software reads the stored sector and compares it with the new one to eliminate the possibility of 'hash collisions'. Purity replaces duplicate blocks with pointers to the single copy of their contents"); FlashArray User's Guide, at 36 ("Purity attempts to compress the data in blocks that remain after pattern elimination and deduplication, choosing among several well-known compression algorithms that balance compression speed against compactness of the result. The software stores compressed rather than original host-written data in NVRAM, in write unit buffers, and ultimately on solid state storage."); https://www.purestorage.com/products/purity/purity-reduce.html ("Inline compression reduces data to use less capacity than the original format. Append-only write layout and variable addressing optimize compression savings by removing the wasted space that fixed-block architectures introduce. Combined with Deep Reduction, compression delivers 2 - 4x data reduction, and is the primary form of data reduction for databases.").

47. The Accused Instrumentalities utilize multiple formats of compression to compress data for backup. *See, e.g.*, https://www.purestorage.com/resources/glossary/data-deduplication.html ("At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss. Best of all, these savings are delivered without requiring any tuning."); FlashArray User's Guide, at 36

("For each sector of data that enters an array, Purity computes a hash checksum which it compares against the checksums of already-stored sectors. If it finds a match, the software reads the stored sector and compares it with the new one to eliminate the possibility of 'hash collisions'. Purity replaces duplicate blocks with pointers to the single copy of their contents"); FlashArray User's Guide, at 36 ("Purity attempts to compress the data in blocks that remain after pattern elimination and deduplication, choosing among several well-known compression algorithms that balance compression speed against compactness of the result. The software stores compressed rather than original host-written data in NVRAM, in write unit buffers, and ultimately on solid state storage."); https://www.purestorage.com/products/purity/purity-reduce.html ("Inline compression reduces data to use less capacity than the original format. Append-only write layout and variable addressing optimize compression savings by removing the wasted space that fixed-block architectures introduce. Combined with Deep Reduction, compression delivers 2 - 4x data reduction, and is the primary form of data reduction for databases."). An encoder to compress data is selected based on content of the one or more data blocks on which a compression algorithm is applied. To prepare to decompress the data, the Accused Instrumentalities include a data decompression processor configured to analyze the data packet to identify one or more recognizable data tokens associated with the data packet, the one or more recognizable data identifying a selected encoder used to compress one or more data blocks to provide the one or more compressed data blocks, the encoder being selected based on content of the one or more data blocks on which a compression algorithm was applied.

48. To decompress the data, the Accused Instrumentalities include one or more decompression decoders configured to decompress a compressed data block from among the one or more compressed data blocks associated with the data packet based on the one or more recognizable data tokens. *See, e.g.*, https://www.purestorage.com/resources/glossary/data-deduplication.html ("At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss. Best of all, these savings are delivered without requiring any tuning."); FlashArray User's Guide, at 36 ("For each sector of data that enters an array, Purity computes a hash checksum which it compares against the checksums of already-stored sectors. If it finds a match, the software reads the stored sector and compares it with the new one to eliminate the possibility of 'hash collisions'. Purity replaces duplicate blocks with pointers to the single copy of their contents"); FlashArray User's Guide, at 36 ("Purity attempts to compress the data in blocks that remain after pattern elimination and deduplication, choosing among several well-known compression algorithms that balance compression speed against compactness of the result. The software stores compressed rather than original host-written data in NVRAM, in write unit buffers, and ultimately on solid state storage."); https://www.purestorage.com/products/purity/purity-reduce.html ("Inline compression reduces data to use less capacity than the original format. Append-only write layout and variable addressing optimize compression savings by removing the wasted space that fixed-block architectures introduce. Combined with Deep Reduction,

compression delivers 2 - 4x data reduction, and is the primary form of data reduction for databases.").

49. One of the compression formats in the Accused Instrumentalities is content dependent data decompression. *See, e.g.*, https://www.purestorage.com/resources/glossary/data-deduplication.html ("At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss. Best of all, these savings are delivered without requiring any tuning."); FlashArray User's Guide, at 36 ("For each sector of data that enters an array, Purity computes a hash checksum which it compares against the checksums of already-stored sectors. If it finds a match, the software reads the stored sector and compares it with the new one to eliminate the possibility of 'hash collisions'. Purity replaces duplicate blocks with pointers to the single copy of their contents"). The one or more decompression decoders in the Accused Instrumentalities are further configured to decompress the compressed data block utilizing content dependent data decompression to provide a first decompressed data block when the one or more recognizable data tokens indicate that the data block was encoded utilizing content dependent data compression.

50. One of the compression formats in the Accused Instrumentalities is content independent data decompression. *See, e.g.*, FlashArray User's Guide, at 36 ("Purity attempts to compress the data in blocks that remain after pattern elimination and deduplication, choosing among several well-known compression algorithms that balance compression speed against compactness of the result. The software stores compressed

rather than original host-written data in NVRAM, in write unit buffers, and ultimately on solid state storage."); https://www.purestorage.com/products/purity/purity-reduce.html ("Inline compression reduces data to use less capacity than the original format. Append-only write layout and variable addressing optimize compression savings by removing the wasted space that fixed-block architectures introduce. Combined with Deep Reduction, compression delivers 2 - 4x data reduction, and is the primary form of data reduction for databases."). The one or more decompression decoders in the Accused Instrumentalities are further configured to decompress the compressed data block utilizing content independent data decompression to provide a second decompressed data block when the one or more recognizable data tokens indicate that the data block was encoded utilizing content independent data compression.

51. The Accused Instrumentalities include an output interface, coupled to the data decompression engine, configured to output a decompressed data packet including the first or the second decompressed data block. For example, the in the Accused Instrumentalities the SSD arrays have Ethernet, Fibre Channel, PCIe, InfiniBand, and/or iSCSI interfaces for communications with the main board, host, and intra-array communications. FlashArray User's Guide, at 17-22. Furthermore, the Accused Instrumentalities have memory, such as NVRAM, into which decompressed data can be written. *See, e.g.,* FlashArray User's Guide, at ix, 15-16. On information and belief, all of the Accused Instrumentalities have network connections that provide an output interface, coupled to the data decompression engine, configured to output a decompressed data packet including the first or the second decompressed data block.

52. On information and belief, Pure Storage also infringes, directly and through induced infringement, and continues to infringe other claims of the ’203 Patent, for similar reasons as explained above with respect to Claim 14 of the ’203 Patent.

53. On information and belief, use of the Accused Instrumentalities in their ordinary and customary fashion results in infringement of the methods claimed by the ’203 Patent.

54. By making, using, offering for sale, selling and/or importing into the United States the Accused Instrumentalities, and touting the benefits of using the Accused Instrumentalities’ compression features, Pure Storage has injured Realtime and is liable to Realtime for infringement of the ’203 Patent pursuant to 35 U.S.C. § 271.

55. As a result of Pure Storage’s infringement of the ’203 Patent, Plaintiff Realtime is entitled to monetary damages in an amount adequate to compensate for Pure Storage’s infringement, but in no event less than a reasonable royalty for the use made of the invention by Pure Storage, together with interest and costs as fixed by the Court.

COUNT IV
INFRINGEMENT OF U.S. PATENT NO. 9,116,908

56. Plaintiff Realtime realleges and incorporates by reference paragraphs 1-55 above, as if fully set forth herein.

57. Plaintiff Realtime is the owner by assignment of United States Patent No. 9,116,908 (“the ’908 Patent”) entitled “System and methods for accelerated data storage and retrieval.” The ’908 Patent was duly and legally issued by the United States Patent and Trademark Office on August 25, 2015, and Claims 1, 2, 4-6, 9, 11, 21, 22, 24, and 25 of the ’908 Patent confirmed as patentable in a Final Written Decision of the Patent Trial

and Appeal Board on October 31, 2017. A true and correct copy of the ’908 Patent is included as Exhibit D.

58. On information and belief, Pure Storage has offered for sale, sold and/or imported into the United States Pure Storage products and services that infringe the ’908 Patent, and continues to do so. By way of illustrative example, these infringing products and services include, without limitation, Pure Storage’s products and services, *e.g.*, Purity Reduce, FlashArray M10, FlashArray M20, FlashArray M50, FlashArray M70, and FlashArray X70, and the system hardware on which they operate, and all versions and variations thereof since the issuance of the ’908 Patent (the “Accused Instrumentality”).

59. On information and belief, Pure Storage has directly infringed and continues to infringe the ’908 Patent, for example, through its own use and testing of the Accused Instrumentality, which constitutes a system comprising: a memory device; and a data accelerator configured to compress: (i) a first data block with a first compression technique to provide a first compressed data block; and (ii) a second data block with a second compression technique, different from the first compression technique, to provide a second compressed data block; wherein the compressed first and second data blocks are stored on the memory device, and the compression and storage occurs faster than the first and second data blocks are able to be stored on the memory device in uncompressed form. Upon information and belief, Pure Storage uses the Accused Instrumentality, an infringing system, for its own internal non-testing business purposes, while testing the Accused Instrumentality, and while providing technical support and repair services for the Accused Instrumentality to Pure Storage’s customers.

60. On information and belief, use of the Accused Instrumentality in its ordinary and customary fashion results in infringement of the systems claimed by the ’908 Patent.

61. On information and belief, Pure Storage has had knowledge of the ’908 Patent since at least the filing of this First Amended Complaint or shortly thereafter, and on information and belief, Pure Storage knew of the ’908 Patent and knew of its infringement, including by way of this lawsuit.

62. Upon information and belief, Pure Storage’s affirmative acts of making, using, and selling the Accused Instrumentalities, and providing implementation services and technical support to users of the Accused Instrumentalities, have induced and continue to induce users of the Accused Instrumentalities to use them in their normal and customary way to infringe Claim 1 of the ’908 Patent by making or using a system comprising: a memory device; and a data accelerator configured to compress: (i) a first data block with a first compression technique to provide a first compressed data block; and (ii) a second data block with a second compression technique, different from the first compression technique, to provide a second compressed data block; wherein the compressed first and second data blocks are stored on the memory device, and the compression and storage occurs faster than the first and second data blocks are able to be stored on the memory device in uncompressed form. For example, Pure Storage explains to customers the benefits of using the Accused Instrumentalities, such as by touting their performance advantages: “At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block

architectures miss. Best of all, these savings are delivered without requiring any tuning." *See* https://www.purestorage.com/resources/glossary/data-deduplication.html. For similar reasons, Pure Storage also induces its customers to use the Accused Instrumentalities to infringe other claims of the '908 Patent. Pure Storage specifically intended and was aware that these normal and customary activities would infringe the '908 Patent. Pure Storage performed the acts that constitute induced infringement, and would induce actual infringement, with the knowledge of the '908 Patent and with the knowledge, or willful blindness to the probability, that the induced acts would constitute infringement. On information and belief, Pure Storage engaged in such inducement to promote the sales of the Accused Instrumentalities. Accordingly, Pure Storage has induced and continues to induce users of the accused products to use the accused products in their ordinary and customary way to infringe the '908 Patent, knowing that such use constitutes infringement of the '908 Patent.

63. The Accused Instrumentality evidently includes a memory device and a data accelerator configured to compress: (i) a first data block with a first compression technique to provide a first compressed data block; and (ii) a second data block with a second compression technique, different from the first compression technique, to provide a second compressed data block. For example, the Accused Instrumentalities also use one or more memory devices, including including solid state drives (SSDs). *See, e.g.*, FlashArray User's Guide, at ix. ("SSDs mounted in each storage shelf. An array's first two shelves contain 22 SSDs, with two bays reserved for NVRAMs; each additional shelf contains 24 SSDs."). The Accused Instrumentality includes a data accelerator configured to compress: (i) a first data block with a first compression technique (e.g. deduplication)

to provide a first compressed data block; and (ii) a second data block with a second compression technique (e.g. compression), different from the first compression technique, to provide a second compressed data block. *See, e.g.*, https://www.purestorage.com/resources/glossary/data-deduplication.html ("At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss. Best of all, these savings are delivered without requiring any tuning."); FlashArray User's Guide, at 36 ("For each sector of data that enters an array, Purity computes a hash checksum which it compares against the checksums of already-stored sectors. If it finds a match, the software reads the stored sector and compares it with the new one to eliminate the possibility of 'hash collisions'. Purity replaces duplicate blocks with pointers to the single copy of their contents"); FlashArray User's Guide, at 36 ("Purity attempts to compress the data in blocks that remain after pattern elimination and deduplication, choosing among several well-known compression algorithms that balance compression speed against compactness of the result. The software stores compressed rather than original host-written data in NVRAM, in write unit buffers, and ultimately on solid state storage."); https://www.purestorage.com/products/purity/purity-reduce.html ("Inline compression reduces data to use less capacity than the original format. Append-only write layout and variable addressing optimize compression savings by removing the wasted space that fixed-block architectures introduce. Combined with Deep Reduction, compression delivers 2 - 4x data reduction, and is the primary form of data reduction for databases.").

64. The Accused Instrumentality stores the compressed first and second data blocks on the memory device. For example, the Accused Instrumentalities have storage media, such as, for example, solid state drives (SSDs). *See, e.g.*, FlashArray User's Guide, at ix. ("SSDs mounted in each storage shelf. An array's first two shelves contain 22 SSDs, with two bays reserved for NVRAMs; each additional shelf contains 24 SSDs."). Also, compressed data blocks are stored temporarily in volatile memory when they are created. The compression and storage occurs faster than the first and second data blocks are able to be stored on the memory device in uncompressed form. *See, e.g.*, https://www.purestorage.com/resources/glossary/data-deduplication.html ("At Pure Storage, high-performance inline deduplication operates on a 512-byte aligned, variable block size range from 4 - 32K. Thus only unique blocks of data are saved on flash – removing even the duplicates that fixed-block architectures miss. Best of all, these savings are delivered without requiring any tuning."); FlashArray User's Guide, at 36 ("For each sector of data that enters an array, Purity computes a hash checksum which it compares against the checksums of already-stored sectors. If it finds a match, the software reads the stored sector and compares it with the new one to eliminate the possibility of 'hash collisions'. Purity replaces duplicate blocks with pointers to the single copy of their contents"); FlashArray User's Guide, at 36 ("Purity attempts to compress the data in blocks that remain after pattern elimination and deduplication, choosing among several well-known compression algorithms that balance compression speed against compactness of the result. The software stores compressed rather than original host-written data in NVRAM, in write unit buffers, and ultimately on solid state storage."); https://www.purestorage.com/products/purity/purity-reduce.html ("Inline

compression reduces data to use less capacity than the original format. Append-only write layout and variable addressing optimize compression savings by removing the wasted space that fixed-block architectures introduce. Combined with Deep Reduction, compression delivers 2 - 4x data reduction, and is the primary form of data reduction for databases.").

65. On information and belief, Pure Storage also infringes, directly and through induced infringement, and continues to infringe other claims of the '908 Patent, for similar reasons as explained above with respect to Claim 1 of the '908 Patent.

66. By making, using, offering for sale, selling and/or importing into the United States the Accused Instrumentalities, and touting the benefits of using the Accused Instrumentalities' compression features, Pure Storage has injured Realtime and is liable to Realtime for infringement of the '908 Patent pursuant to 35 U.S.C. § 271.

67. As a result of Pure Storage's infringement of the '908 Patent, Plaintiff Realtime is entitled to monetary damages in an amount adequate to compensate for Pure Storage's infringement, but in no event less than a reasonable royalty for the use made of the invention by Pure Storage, together with interest and costs as fixed by the Court.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff Realtime respectfully requests that this Court enter:

a. A judgment in favor of Plaintiff that Pure Storage has infringed, either literally and/or under the doctrine of equivalents, the '728 Patent, the '751 Patent, the '203 Patent, and the '908 Patent;

b. A permanent injunction prohibiting Pure Storage from further acts of infringement of the '728 Patent, the '751 Patent, the '203 Patent, and the '908 Patent;

c. A judgment and order requiring Pure Storage to pay Plaintiff its damages, costs, expenses, and prejudgment and post-judgment interest for its infringement of the ’728 Patent, the ’751 Patent, the ’203 Patent, and the ’908 Patent; and

d. A judgment and order requiring Pure Storage to provide an accounting and to pay supplemental damages to Realtime, including without limitation, prejudgment and post-judgment interest;

e. A judgment and order finding that this is an exceptional case within the meaning of 35 U.S.C. § 285 and awarding to Plaintiff its reasonable attorneys’ fees against Defendants; and

f. Any and all other relief as the Court may deem appropriate and just under the circumstances.

DEMAND FOR JURY TRIAL

Plaintiff, under Rule 38 of the Federal Rules of Civil Procedure, requests a trial by jury of any issues so triable by right.

Dated: December 12, 2017

BAYARD, P.A.

/s/ Stephen B. Brauerman
Stephen B. Brauerman (No. 4952)
Sara E. Bussiere (No. 5725)
600 N. King Street, Suite 400
Wilmington, DE 19801
Phone: (302) 655-5000
sbrauerman@bayardlaw.com
sbussiere@bayardlaw.com

Attorneys for Plaintiff Realtime Data LLC d/b/a IXO

OF COUNSEL

Marc A. Fenster (CA SBN 181067)
Reza Mirzaie (CA SBN 246953)
Paul A. Kroeger (CA SBN 229074)
Stanley H. Thompson, Jr. (CA SBN 198825)
RUSS AUGUST & KABAT
12424 Wilshire Boulevard, 12th Floor
Los Angeles, CA 90025
(310) 826-7474
mfenster@raklaw.com
rmirzaie@raklaw.com
pkroeger@raklaw.com
sthompson@raklaw.com