IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

SOUND VIEW INNOVATIONS, LLC, Plaintiff, v. FMR, LLC and FIDELITY BROKERAGE SERVICES LLC, Defendants.	Case No. _____ **JURY TRIAL DEMANDED**

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Sound View Innovations, LLC ("Sound View"), for its Complaint for Patent Infringement against FMR, LLC ("FMR") and Fidelity Brokerage Services LLC ("FBS") (collectively "Fidelity"), alleges as follows:

INTRODUCTION

1. Sound View is an intellectual property licensing company. Sound View's patent portfolio includes more than 850 active and pending patents worldwide, including approximately 450 active U.S. Patents. Sound View's patents were developed by researchers at Alcatel Lucent ("Lucent") and its predecessors. Lucent is home to the world-renowned Bell Laboratories, which has a long and storied history of innovation. Researchers at Lucent's Bell Laboratories have developed a wide variety of key innovations that have greatly enhanced the capabilities and utility of computer systems and networks. This has resulted in benefits such as better and more efficient computer networking, computer security, and user experiences.

2. Patents enjoy the same fundamental protections as real property. Sound View, like any property owner, is entitled to insist that others respect its property and to demand compensation from those who take it for their own use. Fidelity has used, and continues to use, Sound View's

patents. Moreover, despite Sound View's repeated attempts to negotiate, Fidelity refuses to take a license, but continues to use Sound View's property.

NATURE OF THE CASE

3. This action arises under 35 U.S.C. § 271 for Fidelity's infringement of Sound View's United States Patent Nos. 5,806,062 (the "'062 patent"), 6,112,279 ("the '279 patent"), 6,125,371 (the "'371 patent"), 6,502,133 (the "'133 patent"), and 6,912,645 (the "'645 patent") (collectively, the "Patents-In-Suit").

THE PARTIES

4. Plaintiff Sound View is a Delaware limited liability company, with its principal place of business at 2001 Route 46, Waterview Plaza, Suite 310, Parsippany, New Jersey 07054.

5. Defendant FMR is a Delaware limited liability company, with its principal place of business at 235 Summer Street, Boston, Massachusetts 02210. FMR may be served with process by serving its registered agent, The Corporation Trust Company, Corporation Trust Center, 1209 Orange Street, Wilmington, Delaware 19801.

6. Defendant FBS is a Delaware limited liability company, with its principal place of business at 235 Summer Street, Boston, Massachusetts 02210. FBS may be served with process by serving its registered agent, The Corporation Trust Company, Corporation Trust Center, 1209 Orange Street, Wilmington, Delaware 19801. FBS is a wholly owned subsidiary of FMR.

JURISDICTION AND VENUE

7. This action arises under the patent laws of the United States, including 35 U.S.C. § 271 *et seq.* The jurisdiction of this Court over the subject matter of this action is proper under 28 U.S.C. §§ 1331 and 1338(a).

8. This Court has personal jurisdiction over FMR because, among other things: FMR is organized under the laws of the State of Delaware; and FMR has committed, aided, abetted, contributed to and/or participated in the commission of acts (directly, and/or indirectly through its subsidiaries, including FBS) giving rise to this action, including acts within the State of Delaware and this judicial district, and has established minimum contacts within the forum such that the exercise of jurisdiction over FMR would not offend traditional notions of fair play and substantial justice.

9. This Court has personal jurisdiction over FBS because, among other things: FBS is organized under the laws of the State of Delaware; and FBS has committed, aided, abetted, contributed to and/or participated in the commission of acts (directly, and/or indirectly through its subsidiaries) giving rise to this action within the State of Delaware and this judicial district and has established minimum contacts within the forum such that the exercise of jurisdiction over FBS would not offend traditional notions of fair play and substantial justice.

10. Venue is proper in this Court pursuant to 28 U.S.C. §§ 1391(b) and (c) and 1400(b), at least because both FMR and FBS are organized under the laws of the State of Delaware.

THE PATENTS-IN-SUIT

11. Sound View incorporates by reference the preceding paragraphs as if fully set forth herein.

12. The ’062 patent, titled “Data Analysis System Using Virtual Databases,” was duly and properly issued by the United States Patent and Trademark Office (“USPTO”) on September 8, 1998. A copy of the ’062 patent is attached hereto as Exhibit A.

13. Sound View is the owner and assignee of the ’062 patent and holds the right to sue for and recover all damages for infringement thereof, including past infringement.

14. The ’279 patent, titled “Virtual Web Caching System,” was duly and properly issued by the United States Patent and Trademark Office (“USPTO”) on August 29, 2000. On May 24, 2011, the USPTO issued an *ex parte* reexamination certificate for the ’279 patent. A copy of the ’279 patent is attached hereto as Exhibit B.

15. Sound View is the owner and assignee of the ’279 patent and holds the right to sue for and recover all damages for infringement thereof, including past infringement.

16. The ’371 patent, titled “System and Method For Aging Versions of Data in a Main Memory Database,” was duly and properly issued by the USPTO on September 26, 2000. A copy of the ’371 patent is attached hereto as Exhibit C.

17. Sound View is the owner and assignee of the ’371 patent and holds the right to sue for and recover all damages for infringement thereof, including past infringement.

18. The ’133 patent, titled “Real-Time Event Processing System With Analysis Engine Using Recovery Information,” was duly and properly issued by the USPTO on December 31, 2002. A copy of the ’133 patent is attached hereto as Exhibit D.

19. Sound View is the owner and assignee of the ’133 patent and holds the right to sue for and recover all damages for infringement thereof, including past infringement.

20. The ’645 patent, titled “Method and Apparatus for Archival Data Storage,” was duly and properly issued by the USPTO on June 28, 2005. A copy of the ’645 patent is attached hereto as Exhibit E.

21. Sound View is the owner and assignee of the ’645 patent and holds the right to sue for and recover all damages for infringement thereof, including past infringement.

BACKGROUND FACTS

22. On January 11, 2016, Sound View sent Fidelity a letter, notifying Fidelity of its infringement of five Sound View patents, including the ’133 and ’371 patents. Sound View notified Fidelity of representative Fidelity systems and features that infringe those patents and explained its intention to allow Fidelity to continue to use the inventions covered in those patents through a license from Sound View. Sound View further requested a meeting to discuss the matter in more detail.

23. On May 17, 2016, Sound View met with Fidelity to discuss Sound View and its patent portfolio, including the ’062, ’279, ’133 and ’371 patents. In connection with that meeting, Sound View provided Fidelity with claim charts detailing Fidelity’s infringement of the ’133 and ’371 patents.

24. On November 3, 2016, Sound View notified Fidelity during a teleconference that Fidelity was infringing at least three Sound View patents, including the ’279 patent.

25. On November 16, 2016, in response to a request from Fidelity, Sound View sent Fidelity additional evidence regarding Fidelity’s infringement of at least three Sound View patents, including the ’279 patent.

26. On November 23, 2016, in response to a request from Fidelity, Sound View sent Fidelity claim charts detailing Fidelity’s infringement of three Sound View patents, including the ’279 patent.

27. On December 22, 2016, Sound View had a follow-up teleconference with Fidelity to discuss Fidelity’s infringement of Sound View’s patents.

28. On January 17, 2017, Sound View sent a follow up email to Fidelity with updated information regarding Sound View’s licensing activities.

29. On February 1, 2017, Fidelity responded to Sound View's January 17, 2017 email, indicating that Fidelity had engaged outside counsel to review the patents Sound View had identified to Fidelity.

30. On February 22, 2017, Sound View responded to Fidelity's February 1, 2017 email, in which Sound View endeavored to advance its licensing discussions with Fidelity.

31. On March 7, 2017, Fidelity responded to Sound View's February 22, 2017 email, indicating that Fidelity was still investigating the patents Sound View had identified to Fidelity.

32. On March 9, 2017, Sound View responded to Fidelity's March 7, 2017 email, in which Sound View requested a follow-up call to advance its licensing discussions with Fidelity.

33. On March 17, 2017, Sound View and Fidelity (including Fidelity's outside counsel) had a follow-up teleconference to further discuss Sound View's patent portfolio and Fidelity's infringement.

34. On March 28, 2017, Sound View notified Fidelity in a follow-up email that Fidelity was infringing two additional Sound View patents, including the '645 patent.

35. Despite lengthy correspondence, Fidelity has refused to engage in good faith licensing negotiations with Sound View.

36. Fidelity continues to knowingly, intentionally, and willfully infringe Sound View's patents so as to obtain their significant benefits without paying any compensation to Sound View. Sound View has no other choice but to seek relief through litigation.

COUNT ONE

INFRINGEMENT OF THE '062 PATENT

37. Sound View incorporates by reference the preceding paragraphs as if fully set forth herein.

38. The ’062 patent generally relates to customizable data processing applications that rely on a combination of reusable software operators, such as initial operators, query operators, terminal operators, and/or external operators, to process source information from a virtual database in a particular schema, such as HTML or XML, and transform that source information into another virtual database having the same schema.

39. The ’062 patent is valid and enforceable.

40. Various types of documents may be stored in a computer system, such as word processing files, computer programs, HTML documents, financial files, employee files, etc. When dealing with large or complex files, it is often desirable to analyze or alter the structure and content of the documents; for example, comparing a first version to a second version, or analyzing dependency relationships between various sections of computer code.

41. In order to aid such analysis, a database may be constructed which contains information describing the structure of the documents. Various database queries may be performed to extract and process information describing the structure of the source documents. A collection of source documents, along with an associated database that describes the structure of the documents, is called a repository.

42. To analyze source document information, it is necessary to process information contained in the repository. A computer program that extracts or converts information from a repository is called an operator. Thus, an operator receives a source document and/or a database as input, processes the input, and produces some output. A simple example of an operator is a program that takes a source document as input and counts the number of occurrences of a particular word, and outputs a number containing the number of times the particular word occurs. The overall

function of the analysis—in the above example, a count of the number of occurrences of a particular word—is called an application.

43. At the time of the invention of the ’062 patent, in existing repository analysis systems, operators were designed for single applications. Thus, the user indicated which operator he/she wished to apply to the repository, and the system processed the repository accordingly. The user was presented with the output when the processing was finished. Different operators processed the repository in different manners, but there was no convenient mechanism for combining the various operators to create new applications. Thus, when a new application was desired, a new operator would need to be designed from scratch.

44. Prior art repository analysis systems generally were closed systems, in that all operators were applied within the confines of the system, and all database accesses were performed within the system. For example, a repository analysis system operator may have produced as output a file containing information about the structure of a computer program. In conventional closed systems, this output could not be further processed by, for example, an external graphics program that would format the output in a desired manner. Instead, the output could only be formatted according to operators that were internal to the repository system. There was no convenient mechanism to allow the repository analysis system to communicate with operators that were external to the system.

45. The inventors of the ’062 patent solved these discrete computer-based problems by providing an apparatus and method for creating data analysis applications using reusable software operators. For example, query operators receive data in a particular virtual database format, process the data in the virtual database, and output the results of the processing in another virtual database that has the same format as the original virtual database. A plurality of query operators

can be combined to customize the processing of the data. In addition, initial operators convert source information into the virtual database format so that the query operators can analyze the source data. External operators take an external format as input and create another external format as output. Also, terminal operators are used to convert a virtual database into an external format. A user can combine initial, query, terminal, and external operators to create customizable data processing applications.

46. Creating data analysis applications using reusable software operators, as described in the '062 patent, is particularly useful in that the external format data may be processed in various ways, thus allowing flexible presentation of the analysis results.

47. Fidelity's platforms, web pages, and servers have used the Document Object Model ("DOM") to create and process customizable data analysis and processing applications. The DOM is an application programming interface ("API") that allows documents to be modelled using objects of a variety of data formats, including HTML and XML. It defines the logical structure of documents and the way a document is accessed and manipulated.

48. Using the DOM, the nodes (or objects) of every document are organized in a tree structure, called the "DOM tree," and can be manipulated individually using the DOM methods (or operators). With the DOM, programmers can build documents, navigate their structure, and add, modify, or delete elements and content. Anything found in an HTML or XML document can be manipulated in this way using the DOM, with a few exceptions.

49. As an object model, the DOM identifies: (1) the interfaces and objects used to represent and manipulate a document; (2) the semantics of these interfaces and objects – including both behavior and attributes of the relationships; and (3) collaborations among these interfaces and objects.

50. jQuery is a DOM manipulation library that makes it easier to use JavaScript on a website by taking more complex code needed to manipulate the DOM and wrapping the code into simpler methods that can be called with smaller amounts of JavaScript.

51. Fidelity has used jQuery throughout its products and services, including its webpages such as fidelity.com. For example, when a user logs into his or her account on fidelity.com, jQuery methods are used to create pages during the login process.

52. Fidelity has infringed one or more claims of the ’062 patent under 35 U.S.C. § 271(a), literally and/or under the doctrine of equivalents, by making, using, selling, and/or offering for sale in the United States, and/or importing into the United States, products and/or methods encompassed by those claims, including for example, by making, using, selling, offering for sale, and/or importing its Fidelity systems and platforms, including for example its web pages and servers that use and have used jQuery.

53. On May 17, 2016, Sound View informed Fidelity that at least its processing of data in first and second virtual databases and combining query operators to create applications infringed the ’062 patent.

54. For example, Fidelity has infringed claim 14 by using a method for processing information (such as Fidelity’s applications, web pages, and/or servers that use and have used jQuery) comprising the steps of:

a. providing a plurality of software operators (such as jQuery methods, including, for example, “.append( ),” “.clone( ),” “.attr( ),” and “.wrap( )”) each configured to receive a virtual database (such as DOM nodes (or objects) or web pages, describing the structure of a document) having a first schema (such as HTML or XML), for processing information

contained in said virtual database (such as by applying a jQuery method to a node in the DOM tree), and for outputting a virtual database having said first schema; and

b. combining at least two of said software operators to create an application (such as that used to construct and serve Fidelity's web pages).

55. Sound View has been damaged by Fidelity's infringement of the '062 patent. Sound View is entitled to recover from Fidelity the damages sustained by Sound View as a result of Fidelity's wrongful acts in an amount adequate to compensate Sound View for Fidelity's infringement subject to proof at trial.

COUNT TWO

INFRINGEMENT OF THE '279 PATENT

56. Sound View incorporates by reference the preceding paragraphs as if fully set forth herein.

57. The '279 patent generally relates to an improved caching arrangement for, e.g., web servers.

58. The '279 patent is valid and enforceable.

59. Caching has been widely recognized as one of the solutions to the problem of congestion of traffic on the World Wide Web, because caching can substantially reduce latency and improve bandwidth efficiency.

60. At the time the inventors of the '279 patent submitted the application leading to the '279 patent, the inventors recognized that existing caches suffered from a number of problems. One problem was that for each local cache miss, the cache server had to send a query message to all its neighbors and parents, and receive a reply from each of them. In a large cache system with many cache servers or a deep hierarchy, this overhead was high, and represented a particularly

inefficient use of resources when the requested object is small. Additionally, the request/reply interaction introduced latency during the resolution. Another problem of caches that existed at that time is that the search for a cached object was done in a distributed fashion; for each query, all neighbors and parents had to search their caches in parallel, no matter whether the result was a hit or miss. Since a final miss could not be concluded until the slowest cache server responded or the waiting timer expired, a low hit rate at any one of the cache servers slowed down the entire system. Still another problem of then-existing caches was that there was no explicit cache placement policy. An object may have been cached at any of the cache servers; because objects from a same Web site may have been spread over all cache servers, the system needed to search all the cache servers in order to determine whether an object was cached or not.

61. In particular, to solve this discrete computer-centric problem, the ’279 patent teaches an improved caching arrangement by interconnecting a plurality of cache servers, for example, with high speed and high capacity connections. At least some of the cache servers may be connected to a data network, e.g., the Internet. Each cache server may include a selection module that determines whether the cache server can service an incoming request for information, whether the request ought to be directed to another one of the cache servers, or whether the request ought to be routed to the site from whence the information is requested. In making this determination, in accordance with one embodiment, the module consults a table that associates sites or/and sub-sites with specific ones of the cache servers. In another embodiment, the determination is made by translating the address of the site, or sub-site, or web page that is requested; for example, with a hash function.

62. Fidelity uses and has used a caching system known as OpenStack within Fidelity’s data systems, including OpenStack Object Storage (Swift). For example, Fidelity’s use of

OpenStack Object Storage (Swift) has been openly advertised by Fidelity, vendors of OpenStack Object Storage (Swift) such as Rackspace, and others.

63. OpenStack Object Storage (Swift) is a service that provides software storing and retrieving data over HTTP, including proxy and storage services.

64. OpenStack Object Storage (Swift) systems comprise a collection of interconnected proxy servers (or proxy nodes) and storage nodes.

65. Each proxy node in an OpenStack Object Storage (Swift) system has at least one public interface that processes client requests, authenticates those requests, and performs appropriate actions in response to such requests.

66. In an OpenStack Object Storage (Swift) system, client requests include a storage URL of the object being requested, which includes at least the cluster location and storage location.

67. When the proxy node in an OpenStack Object Storage (Swift) system receives a valid request from a client, it will determine the storage node(s) for the requested object based on at least a hash of the object name, and send the request to the storage node(s).

68. Fidelity has infringed one or more claims of the ’279 patent under 35 U.S.C. § 271(a), literally and/or under the doctrine of equivalents, by making, using, selling, and/or offering for sale in the United States, and/or importing into the United States, products and/or methods encompassed by those claims, including for example, by making, using, selling, offering for sale, and/or importing Fidelity systems and platforms that include or use an OpenStack Object Storage (Swift) system, such as Fidelity’s servers for its private cloud.

69. On May 17, 2016, Sound View informed Fidelity that its systems and applications infringe the ’279 patent. However, Fidelity has not stopped infringing.

70. For example, Fidelity infringes claim 25 by using, in an arrangement including a plurality of cache servers interconnected to form a virtual cache, a method for retrieving information from said virtual cache comprising the steps of:

a. receiving at one of said cache servers a request for information (such as by receiving a client request at a proxy node) which specifies an address of one of multiple sites, or a sub-site of said one of multiple sites, which address designates the source of said information (such as by the client request including a storage URL);

b. converting said address destination to a designation that identifies a cache server in said virtual cache (such as by hashing at least a portion of the storage URL to determine the storage node(s) for the object that is the subject of the request); and

c. directing said request for information to the identified cache server (such as by sending request(s) to the storage node(s) for the object that is the subject of the request).

71. Sound View has been damaged by Fidelity's infringement of the '279 patent. Sound View is entitled to recover from Fidelity the damages sustained by Sound View as a result of Fidelity's wrongful acts in an amount adequate to compensate Sound View for Fidelity's infringement subject to proof at trial.

72. In committing these acts of infringement, Fidelity committed egregious misconduct including, for example, acting despite knowing that its actions constituted infringement of a valid patent, or recklessly disregarding the fact that its actions constituted an unjustifiably high risk of infringement of a valid and enforceable patent.

73. Fidelity's infringement of the '279 patent was and is deliberate and willful, entitling Sound View to increased damages under 35 U.S.C. § 284 and to attorney fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

COUNT THREE

INFRINGEMENT OF THE '371 PATENT

74. Sound View incorporates by reference the preceding paragraphs as if fully set forth herein.

75. The '371 patent generally relates to an improved multi-versioned database management system and method that creates multiple versions of data records affected by update transactions and increases capacity of memory by deleting versions of data records in response to associated time stamps and a measurable characteristic of the memory. In the context of the '371 patent, "measurable characteristics of the memory" are a current utilization or capacity of memory, a trend analysis of a utilization or capacity of memory over a time period, or any other applied mathematics- or statistics-based analysis, including a comparison of any of the same with a threshold, ceiling/floor, limit, set point, or the like.

76. The '371 patent is valid and enforceable.

77. Database managers ("DBMs") have long been used in computer systems to manage large amounts of data. A DBM is a control application that supervises or manages interactions between application tasks and a database. The '371 patent inventors recognized that two important DBM functions are to ensure (i) data recovery (in response to a database crash caused by, for example, a power outage or a program crash), and (ii) data integrity. Data recovery involves rebuilding at least part of a database after all or part of its data is corrupted or lost, based on the last known valid or uncorrupted state. With respect to data integrity, latency in DBMs was largely intolerable. Latency refers to the time differential between a request for data and subsequent receipt of data. Latency is largely impacted by the type of computer memory on which the database is stored. There are two classifications of computer memory, volatile memory and non-volatile

memory. Volatile memory is memory which does not retain data after power is lost, and is typically characterized by fast access to data. Non-volatile memory is memory that retains data after power is lost and is typically characterized by slower access to data. As a general matter, volatile memory is more expensive than non-volatile memory. Early computer database systems were divided among main (volatile) memory and disk (non-volatile memory). Those disk-based DBMs frequently failed to meet the performance requirements of contemporary information management systems because of the latencies inherent with non-volatile memory transactions.

78. One popular method to solve that latency problem was to map the entire database into the main memory. For data integrity purposes, however, those conventional main memory DBMs had to delay the processing of update transactions. For example, the conventional main memory DBMs had to prevent an update transaction from modifying a data record while another process was simultaneously relying on that data record. In order to reduce conflicts between update transactions and read-only transactions, contemporary databases created multiple versions of data records, known as multi-versioning. In those multi-version DBMs, read-only transactions were given consistent, but out-of-date views of certain data records or data record types.

79. Although those multi-versioning techniques reduced "waits" and conflicts among transactions, they conflicted with DBM efforts to utilize main memory capacity efficiently because main memory continuously expended processing resources collecting data record versions that were no longer needed. The '371 patent solved this computer-based problem—that of lacking an efficient means to reclaim memory space no longer used by multi-version techniques—by aging data record versions in the database based on timestamps and measurable characteristics of the memory. The '371 patent inventions extend to, and provide benefits to, DBMs that utilize secondary or mass storage as opposed to main memory.

80. In particular, to solve this discrete computer-centric problem, the ’371 patent teaches a system that includes each of a time stamping controller, a versioning controller and an aging controller. The time stamping controller assigns a time stamp to transactions to be performed on the database; the time stamp may be assigned as a function of a time stamp counter. The versioning controller creates multiple versions of data records of the database that are affected by update transactions. The aging controller, which may be associated, directly or indirectly, with each of the time stamping and versioning controllers, monitors at least one measurable characteristic and deletes prior ones of the multiple data record versions in response to the time stamp and the at least one measurable characteristic to thereby increase the memory capacity.

81. The monitoring of memory utilization as embodied in the ’371 patent allows DBMs to avoid continuously expending processing resources collecting and aging older, no longer needed data record versions.

82. Fidelity uses and has used a distributed database known as Cassandra in its data systems. For example, current and former employees of Fidelity have openly advertised Fidelity’s use of Cassandra.

83. The Cassandra database is stored in a memory comprising a combination of “memtable” and “SSTable.” A memtable is a Cassandra table-specific, in-memory data structure that resembles a write-back cache. A sorted string table (SSTable) is an immutable data file to which Cassandra writes memtables periodically. SSTables are stored on disk sequentially and maintained for each Cassandra table.

84. During a write transaction, a timestamp is assigned to the transaction performed on the Cassandra database.

85. Cassandra databases utilize periodic compaction to manage the accumulation of SSTables.

86. Cassandra databases have configurable parameters (such as min_threshold and max_threshold parameters) that control when a minor compaction occurs.

87. Fidelity also uses and has used a distributed database known as HBase in its data systems. For example, Fidelity's use of HBase has been publicly advertised by current and former employees of Fidelity, as well as by market research organizations such as Infosys.

88. HBase is a column-oriented database management system that runs on top of a Hadoop Distributed File System. Applications store data into HBase tables that are made up of rows and columns. Table cells—the intersection of row and column coordinates—are versioned. When something is written into one of Fidelity's HBase databases, it is first written to an in-memory store (memstore), and then is flushed into a store file. When Fidelity puts data into HBase, a timestamp is required and is generated by HBase. Performing a "put" operation to HBase creates a new version of a cell.

89. Fidelity controls the number of versions stored in HBase.

90. During major compaction, excess versions are deleted from the store file. The number of versions to be deleted is determined by comparing the number of versions stored to the MaxVersions. If the number of stored versions of the store files is greater than the Max Versions, then the excess versions are deleted. The versions that are deleted are selected based on timestamps.

91. During minor compactions, a configurable number of smaller store files are combined into fewer, but larger store files. The store files to be compacted in a minor compaction

are determined at least in part by configurable store file number, size, and/or ratio parameters. During a minor compaction, versions are also deleted based on timestamps.

92. Fidelity has infringed one or more claims of the ’371 patent under 35 U.S.C. § 271(a), literally and/or under the doctrine of equivalents, by making, using, selling, and/or offering for sale in the United States, and/or importing into the United States, products and/or methods encompassed by those claims, including for example, by making, using, selling, offering for sale, and/or importing servers and systems that include or use applications based on Cassandra, such as Fidelity’s servers and data systems.

93. Fidelity has also infringed one or more claims of the ’371 patent under 35 U.S.C. § 271(a), literally and/or under the doctrine of equivalents, by making, using, selling, and/or offering for sale in the United States, and/or importing into the United States, products and/or methods encompassed by those claims, including for example, by making, using, selling, offering for sale, and/or importing Fidelity systems and platforms that include or use applications based on HBase, such as Fidelity’s servers and data systems.

94. On January 11, 2016, Sound View informed Fidelity that its systems infringe the ’371 patent. However, Fidelity has not stopped infringing.

95. For example, Fidelity infringes claim 1 by using a processing system (such as Fidelity’s servers) for use with a database of data records (such as a Cassandra database), said database stored in a memory, comprising:

a. a time stamping controller that assigns a time stamp to transactions to be performed on said database (such as a timestamp assigned during a write transaction);

b. a versioning controller that creates multiple versions of ones of said data records affected by said transactions that are update transactions (such as the new timestamped version of an updated row in the database); and

c. an aging controller that monitors a measurable characteristic of said memory (such as a measurement associated with a min_threshold or max_threshold parameter) and deletes ones of said multiple versions of said ones of said data records in response to said time stamp and said measurable characteristic thereby to increase a capacity of said memory (such as by performing a compaction process in response to the min_threshold parameter being met or exceeded).

96. Similarly, Fidelity infringes claim 1 by using a processing system (such as Fidelity's servers) for use with a database of data records (such as an HBase database), said database stored in a memory, comprising:

a. a time stamping controller that assigns a time stamp to transactions to be performed on said database (such as a timestamp assigned during a write transaction);

b. a versioning controller that creates multiple versions of ones of said data records affected by said transactions that are update transactions (such as the new timestamped version of an updated cell in the database); and

c. an aging controller that monitors a measurable characteristic of said memory (such as the number of versions being stored in the store file, and/or the store file number, size, and/or ratio parameters) and deletes ones of said multiple versions of said ones of said data records in response to said time stamp and said measurable characteristic thereby to increase a capacity of said memory (such as by deleting a version of the cell based on the measurable characteristic and the timestamp of each version).

97. Sound View has been damaged by Fidelity's infringement of the '371 patent. Sound View is entitled to recover from Fidelity the damages sustained by Sound View as a result of Fidelity's wrongful acts in an amount adequate to compensate Sound View for Fidelity's infringement subject to proof at trial.

98. In committing these acts of infringement, Fidelity committed egregious misconduct including, for example, acting despite knowing that its actions constituted infringement of a valid patent, or recklessly disregarding the fact that its actions constituted an unjustifiably high risk of infringement of a valid and enforceable patent.

99. Fidelity's infringement of the '371 patent was and is deliberate and willful, entitling Sound View to increased damages under 35 U.S.C. § 284 and to attorney fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

COUNT FOUR

INFRINGEMENT OF THE '133 PATENT

100. Sound View incorporates by reference the preceding paragraphs as if fully set forth herein.

101. The '133 patent generally relates to real-time event processing in applications such as telecommunications and computer networks, and more particularly, to a method, apparatus, and system for processing events in a real-time analysis engine, and storing recovery information in a main-memory database system associated with the real-time analysis engine.

102. The '133 patent is valid and enforceable.

103. At the time of the invention of the '133 patent, high performance real-time event processing applications had performance requirements that could not be met by conventional general purpose database management systems. For example, some real-time event processing

applications required the service time for such events to not exceed a few milliseconds. However, with conventional database technology, the service time costs of invoking a structured query language operation over a client-server interface, or the service time costs associated with a single access to secondary storage, could account for hundreds of milliseconds. These limitations led real-time event processing applications instead to rely on the use of custom database systems.

104. These custom database systems had disadvantages: (1) there was a high cost of developing and maintaining custom systems; (2) those high costs could not be amortized across a number of different applications; and (3) custom database systems were generally inflexible and difficult to adapt to unforeseen or evolving requirements.

105. At the time of the invention of the ’133 patent, a need therefore existed for an improved real-time event processing system that could provide the performance benefits of custom database systems, but without sacrificing the flexibility and maintainability typically associated with conventional general-purpose database systems.

106. The inventors of the ’133 patent solved that discrete computer-based problem and improved upon the existing real-time event processing systems by providing a real-time event processing system that avoids the problems associated with custom systems.

107. Using a real-time analysis engine operating in the manner described by the ’133 patent is particularly useful because it can provide transactional access to persistent data, but at the speed of a main-memory system, and it also incorporates a recovery model which stores recovery information in order to facilitate roll-back to a recovery point after a failure.

108. In accordance with the ’133 patent, recovery information regarding a recovery point for a given real-time analysis engine may be stored in a memory portion of the main-memory database system. This way, the real-time event processing system provides a critical path for event

processing that is specifically designed for high performance, while also retaining many desirable features of conventional database systems, including high-level, declarative programming interfaces, and the transactional correctness properties of atomicity, consistency, isolation and durability. These features of the ’133 patent enhance the reliability, robustness, usability and maintainability of the real-time event processing system and any applications built thereon.

109. Fidelity uses and has used frameworks known as Apache Storm (“Storm”) to perform stream processing of events in real-time and continuous data processing. For example, current and former employees of Fidelity have openly advertised Fidelity’s use of Storm. Those systems’ architecture is composed of three components: (1) “Streams,” which are unbounded sequences of tuples that are processed; (2) “Spouts,” which are sources of streams, and (3) “Bolts,” which are responsible for processing the Streams in real-time.

110. Those systems are integrated with Fidelity’s infrastructure, such as its database systems. Events are generated by various Fidelity system applications, *e.g.*, an enterprise-wide monitoring solution; when these system applications generate events, these events are grouped into Streams.

111. Spouts emit Streams into the topology, so that they can subsequently be processed.

112. Bolts are real-time analysis engines that process the Streams. Bolts are capable of performing simple stream transformations, and multiple Bolts are used for more complex stream transformations.

113. Those systems have the capability to save and retrieve in-memory the state of the Bolts. For example, Storm has a default in-memory based state implementation and also a Redis backed implementation that provides state persistence. This main-memory database within Storm

has the function known as state management, allowing it to automatically and periodically take snapshots of the state of the Bolts.

114. Fidelity has infringed one or more claims of the ’133 patent under 35 U.S.C. § 271(a), literally and/or under the doctrine of equivalents, by making, using, selling, and/or offering for sale in the United States, and/or importing into the United States, products and/or methods encompassed by those claims, including for example, by making, using, selling, offering for sale, and/or importing servers and products, such as Fidelity’s servers used for real-time analytics and real-time processing, that include or use applications based on Storm.

115. On January 11, 2016, Sound View informed Fidelity that its systems and applications infringe the ’133 patent. However, Fidelity has not stopped infringing.

116. For example, Fidelity infringes claim 13 by using a method of processing events (such as Streams) generated by at least one system application (such as Fidelity’s database systems), the method comprising the steps of:

a. processing the events in at least one real-time analysis engine (such as a Bolt); and

b. storing in a main-memory database system (such as Storm’s default in-memory based state implementation) associated with the real-time analysis engine recovery information regarding a recovery point for the real-time analysis engine (such as the state information relating to the Bolt’s state).

117. Sound View has been damaged by Fidelity’s infringement of the ’133 patent. Sound View is entitled to recover from Fidelity the damages sustained by Sound View as a result of Fidelity’s wrongful acts in an amount adequate to compensate Sound View for Fidelity’s infringement subject to proof at trial.

118. In committing these acts of infringement, Fidelity committed egregious misconduct including, for example, acting despite knowing that its actions constituted infringement of a valid patent, or recklessly disregarding the fact that its actions constituted an unjustifiably high risk of infringement of a valid and enforceable patent.

119. Fidelity's infringement of the '133 patent was and is deliberate and willful, entitling Sound View to increased damages under 35 U.S.C. § 284 and to attorney fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

COUNT FIVE

INFRINGEMENT OF THE '645 PATENT

120. Sound View incorporates by reference the preceding paragraphs as if fully set forth herein.

121. The '645 patent generally relates to data storage techniques, and more particularly to data storage techniques that are suitable for use in managing archival data in a network storage system.

122. The '645 patent is valid and enforceable.

123. At the time of the invention of the '645 patent, archival data storage typically involved the regular backup of data from a computer or other client machine to an optical jukebox, redundant array of inexpensive disks (RAID) device, magnetic tape drive or other device in a long-term storage system. A typical scenario involved providing backup as a central service for a number of client machines. Client software interfaced with a file system or database and determined what data to back up. The data was copied from the client to a storage device, often over a network, and a record of what was copied was stored in a catalog database.

124. At that time, substantial increases in the capacity of various storage technologies were making it practical to archive data in perpetuity. However, conventional techniques such as those described above were generally not optimized for providing this type of storage.

125. The techniques described in the ’645 patent solve that discrete computer-based problem and improve upon data storage methods and apparatus for use in storing data in perpetuity in archival data storage applications, but also for use in other applications, including primary storage applications.

126. In particular, the ’645 patent provides novel data storage methods and apparatuses in which a data block is processed to generate an address as a function of the contents of the data block, and the data block is then stored in the system in a memory location identified by the address. This approach provides write-once storage of the data block, in that the contents of the data block are not modifiable without also altering the address of the data block determinable from the above-noted processing operation. As an example, in an illustrative embodiment of the invention, the processing of the data block involves determining a fingerprint or other substantially unique identifier of the data block by applying a collision-resistant hash function to the contents of the data block. The address is subsequently determined from the substantially unique identifier by utilizing the identifier to perform a lookup of the address in an index.

127. Fidelity uses and has used Centera, a content addressable storage system for automated archival storage. For example, current and former employees of Fidelity have openly advertised Fidelity’s use of Centera.

128. Content addressable storage devices such as Centera eliminate the need for applications to manage the physical location of information on storage media; instead, addresses

are calculated based on the content itself and allow applications to find and retrieve stored objects from such content-derived addresses.

129. Centera provides WORM (write once, read many) functionality, which prevents modification or deletion of the object once it is stored.

130. In addition, Fidelity uses and has used OpenStack Object Storage (Swift), an object storage system. For example, Fidelity's use of OpenStack Object Storage (Swift) has been openly advertised by Fidelity, vendors of OpenStack Object Storage (Swift) such as Rackspace, and others.

131. Object storage systems such as OpenStack Object Storage (Swift) use an object's hash to determine a storage location for the object.

132. OpenStack Object Storage (Swift) provides WORM (write once, read many) functionality, which prevents modification or deletion of the object once it is stored.

133. Fidelity has infringed one or more claims of the '645 patent under 35 U.S.C. § 271(a), literally and/or under the doctrine of equivalents, by making, using, selling, and/or offering for sale in the United States, and/or importing into the United States, products and/or methods encompassed by those claims, including for example, by making, using, selling, offering for sale, and/or importing servers and products, such as Fidelity's data systems that include Centera and/or OpenStack Object Storage (Swift).

134. On March 28, 2017, Sound View informed Fidelity that its systems infringe the '645 patent. However, Fidelity has not stopped infringing.

135. For example, Fidelity infringes claim 1 by using a method of storing data in a storage system (such as a Centera system), the method comprising the steps of:

a. processing a data block to generate an address which is determined as a function of the contents of the data block (such as by processing a data block to generate a content-derived address); and

b. storing the data block in the system in a memory location identified by the address (such as by storing the data block in the Centera system in a memory location identified by the content-derived address);

c. wherein the processing and storing steps provide write once storage of the data block in the system such that the contents of the data block are not modifiable without also altering the address of the data block determinable in the processing step (such as by utilizing write once read many (WORM) storage of the data block in the system).

136. As an additional example, Fidelity infringes claim 1 by using a method of storing data in a storage system (such as an OpenStack Object Storage (Swift) system), the method comprising the steps of:

a. processing a data block to generate an address which is determined as a function of the contents of the data block (such as by processing a data block to generate a content-derived address); and

b. storing the data block in the system in a memory location identified by the address (such as by storing the data block in the OpenStack Object Storage (Swift) system in a memory location identified by the content-derived address);

c. wherein the processing and storing steps provide write once storage of the data block in the system such that the contents of the data block are not modifiable without also altering the address of the data block determinable in the processing step (such as by utilizing write once read many (WORM) storage of the data block in the system).

137. Sound View has been damaged by Fidelity's infringement of the '645 patent. Sound View is entitled to recover from Fidelity the damages sustained by Sound View as a result of Fidelity's wrongful acts in an amount adequate to compensate Sound View for Fidelity's infringement subject to proof at trial.

138. In committing these acts of infringement, Fidelity committed egregious misconduct including, for example, acting despite knowing that its actions constituted infringement of a valid patent, or recklessly disregarding the fact that its actions constituted an unjustifiably high risk of infringement of a valid and enforceable patent.

139. Fidelity's infringement of the '645 patent was and is deliberate and willful, entitling Sound View to increased damages under 35 U.S.C. § 284 and to attorney fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

RELIEF REQUESTED

Wherefore, Sound View respectfully requests that this Court enter judgment against Fidelity as follows:

a) that Fidelity has infringed each of the Patents-In-Suit;

b) that Fidelity's infringement of the '279, '371, '133, and '645 patents is and has been willful;

c) that Sound View be awarded damages in accordance with 35 U.S.C. § 284, including trebled damages, and, if necessary to adequately compensate Sound View for Fidelity's infringement, an accounting;

d) that this case is exceptional under 35 U.S.C. § 285;

e) that Sound View be awarded the attorney fees, costs, and expenses that it incurs in prosecuting this action; and

f) that Sound View be awarded such further relief at law or in equity as the Court deems just and proper.

DEMAND FOR JURY TRIAL

Sound View hereby demands trial by jury on all claims and issues so triable.

Dated: October 3, 2017

By: */s/ John C. Phillips, Jr.*

John C. Phillips, Jr. (No. 110)
Megan C. Haney (No. 5016)
PHILLIPS, GOLDMAN, MCLAUGHLIN & HALL, P.A.
1200 North Broom Street
Wilmington, Delaware 19806-4204
Telephone: (302) 655-4200
Facsimile: (302) 655-4210
jcp@pgmhlaw.com
mch@pgmhlaw.com

Of Counsel:

DESMARAIS LLP
Alan S. Kellman
Ameet A. Modi
Richard M. Cowell
C. Austin Ginnings
230 Park Avenue
New York, NY 10169
Tel: (212) 351-3400
Fax: (212) 351-3401
akellman@desmaraisllp.com
amodi@desmaraisllp.com
rcowell@desmaraisllp.com
aginnings@desmaraisllp.com

Attorneys for Plaintiff Sound View Innovations, LLC