

Filed on behalf of Petitioners

Filed: August 5, 2026

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

META PLATFORMS, INC. AND SAMSUNG ELECTRONICS CO., LTD.,
Petitioners,

v.

GAEA, LLC,
Patent Owner.

Case No. IPR2026-00425
U.S. Patent No. 10,776,023

**Petition for *Inter Partes* Review of
U.S. Patent No. 10,776,023**

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
II.	MANDATORY NOTICES.....	2
	A. Real Parties-in-Interest	2
	B. Related Matters.....	3
	C. Notice of Counsel and Service Information	3
	D. Fee for <i>Inter Partes</i> Review	5
	E. Grounds for Standing	5
III.	IDENTIFICATION OF CHALLENGES	5
IV.	BACKGROUND	6
	A. Technology Overview	6
	B. '023 Patent.....	7
	C. The Challenged Claims	11
	D. Prosecution History	12
V.	THE PRIOR ART.....	15
	A. Olster (EX1005)	15
	B. Davis (EX1008).....	18
	C. Hashimoto (EX1006)	19
	D. Kilian (EX1007).....	22
VI.	LEVEL OF ORDINARY SKILL IN THE ART	23
VII.	CLAIM CONSTRUCTION	23

VIII. GROUND 1	24
A. Independent Claims 1, 9	25
1. 1[pre]/9[pre]	25
2. 1[a]/9[a]	26
3. 1[b]/9[b]	27
4. 1[b.i]/9[b.i]	28
5. 1[b.ii]	32
6. 1[b.iii]/9[b.ii]	33
7. 1[b.iv]/9[b.iii]	34
8. 1[b.v]/9[b.iv]	36
9. 1[b.vi]/9[b.v]	38
10. 1[b.vii]/9[b.vi]	39
11. 1[b.viii]/9[b.vii]	42
B. Dependent Claims 7, 14, 15	43
1. Claim 7	43
2. Claim 14	44
3. Claim 15	44
IX. GROUND 2	45
A. Motivation to Combine	45
B. Reasonable Expectation of Success	47
C. Claims 1, 7, 9, 14-15	48

D.	Dependent Claim 2.....	50
E.	Claim 32.....	51
1.	32[pre].....	51
2.	32[a]	51
3.	32[b]	51
4.	32[b.i].....	52
5.	32[b.ii].....	52
6.	32[b.iii].....	52
7.	32[b.iv].....	52
F.	Dependent Claims 34, 36	53
1.	Claim 34.....	53
2.	Claim 36.....	53
X.	GROUND 3.....	54
A.	Independent Claims 1, 9.....	55
1.	1[pre]/9[pre]	55
2.	1[a]/9[a].....	56
3.	1[b]/9[b]	56
4.	1[b.i]/9[b.i]	57
5.	1[b.ii].....	61
6.	1[b.iii]/9[b.ii].....	62
7.	1[b.iv]/9[b.iii].....	64

8.	1[b.v]/9[b.iv]	67
9.	1[b.vi]/9[b.v]	68
10.	1[b.vii]/9[b.vi]	69
11.	1[b.viii]/9[b.vii]	71
B.	Dependent Claim 14:	71
1.	Claim 14	71
XI.	GROUND 4	72
A.	Motivation to Combine	72
B.	Reasonable Expectation of Success	73
C.	Dependent Claims 7, 15	74
D.	Independent Claim 16	76
1.	16[pre]	76
2.	16[a]	76
3.	16[b]	76
4.	16[b.i]	76
5.	16[b.ii]	77
6.	16[b.iii]	77
7.	16[b.iv]	77
8.	16[b.v]	79
9.	16[b.vi]	79
10.	16[b.vii]	79

E.	Dependent Claims 17-20	80
1.	Claim 17.....	80
2.	Claim 18.....	81
3.	Claim 19.....	81
4.	Claim 20.....	81
XII. SECONDARY CONSIDERATIONS.....		82
XIII. CONCLUSION		82

TABLE OF AUTHORITIES

	Page(s)
Cases	
<i>Caption Health, Inc. v. The Univ. of British Columbia</i> , IPR2025-01422, Paper 15, 3 (Director Dec. 18, 2025).....	24
<i>Eunsung Global Corp. v. HydraFacial LLC</i> , IPR2025-00445, Paper 14, 3 (July 10, 2025).....	2
<i>Gaea, LLC v. Meta Platforms, Inc.</i> , 4:26-cv-00348 (N.D. Tex) (filed on March 23, 2026).....	3
<i>Gaea, LLC v. Samsung Electronics Co., Ltd.</i> , 2:26-cv-00264 (E.D. Tex.) (filed on March 31, 2026).....	3
<i>Guardant Health, Inc. v. Univ. of Washington</i> , No. 2024-1129, 2026 WL 184334 (Fed. Cir. Jan. 23, 2026)	25
<i>Realtime Data, LLC v. Iancu</i> , 912 F.3d 1368 (Fed. Cir. 2019)	26, 56
Statutes	
35 U.S.C. §§ 102(a)(1), 102(a)(2), and 102(b)(1).....	5
35 U.S.C. § 103.....	<i>passim</i>
Other Authorities	
37 C.F.R. § 42.10(b)	4
37 C.F.R. § 42.15(a)	5
37 C.F.R. §42.24(a)	6
37 C.F.R. § 42.24(d)	6
37 C.F.R. §42.100(b)	24, 25
37 C.F.R. § 42.104(a)	5

TABLE OF EXHIBITS

Exhibit No.	Description
1001	U.S. Patent No. 10,776,023 to Johnson et al. (“the ’023 Patent”)
1002	File History of U.S. Patent No. 10,776,023
1003	Declaration of Heiner Litz (“Litz Decl.”)
1004	Curriculum Vitae of Heiner Litz
1005	U.S. Patent Application Publication No. 2015/0052167 to Olster et al. (“Olster”)
1006	U.S. Patent Publication No. 2016/0313943 to Hashimoto et al. (“Hashimoto”)
1007	U.S. Patent No. 9,075,851 to Kilian et al. (“Kilian”)
1008	U.S. Patent No. 8,874,835 to Davis et al. (“Davis”)
1009	U.S. Patent Application Publication No. 2012/0166749 to Eleftheriou et al. (“Eleftheriou”)
1010	U.S. Patent Application Publication No. 2012/0246435 to Meir et al. (“Meir”)
1011	U.S. Patent Application Publication No. 2015/0032938 to Salessi (“Salessi”)
1012	U.S. Patent Application Publication No. 2015/0146364 to Chou (“Chou”)
1013	RESERVED

Exhibit No.	Description
1014	RESERVED
1015	U.S. Patent Application Publication No. 2014/0089278 to Lovinger et al. (“Lovinger”)
1016	RESERVED
1017	Chip Walter, ‘Kryder's Law' (2005)’ Scientific American, Aug. 2005.
1018	Timothy Feldman and Garth A. Gibson, ‘Shingled Magnetic Recording’ ACM Queue / CMU, 2013.
1019	Albert Chen and Jim Malina, ‘Addressing Shingled Magnetic Recording Drives with Linear Tape File System’ SNIA Storage Developer Conference, 2013
1020	Jorge Campello, ‘SMR: The Next Generation of Storage Technology’ SNIA Storage Developer Conference, 2015.
1021	Samarth Bhargava, ‘Lowering HAMR Near-Field Transducer Temperature via Inverse Electromagnetic Design’ IEEE Transactions on Magnetics, Apr. 2015.
1022	Yu Cai et al., ‘Read Disturb Errors in MLC Nand Flash Memory: Characterization, Mitigation, and Recovery’ DSN 2015.
1023	Nitin Agrawal et al., ‘Design Tradeoffs for SSD Performance’ USENIX Annual Technical Conference, 2008
1024	Feng Chen, David A. Koufaty, and Xiaodong Zhang; ‘Understanding Intrinsic Characteristics and System Implications of Flash Memory Based Solid State Drive’ SIGMETRICS 2009

Exhibit No.	Description
1025	Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, 'The Google File System' SOSP 2003
1026	Kirk McKusick and Sean Quinlan, 'GFS: Evolution on Fast-Forward,' ACM Queue, Aug. 2009 / Communications of the ACM, Mar. 2010
1027	Konstantin Shvachko et al., 'The Hadoop Distributed File System,' IEEE MSST 2010
1028	Sage A. Weil et al., 'Ceph: A Scalable, High-Performance Distributed File System' OSDI 2006
1029	Sage A. Weil et al., 'CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data' SC 2006
1030	Leslie Ashton et al., 'Two Decades of Policy-Based Storage Management for the IBM Mainframe Computer' IBM Systems Journal, 2003
1031	IBM, 'DFSMSHsm Primer'; IBM Redbooks, Sept. 2015
1032	IBM, 'GPFS: Advanced Administration Guide' including policy-based data management for GPFS, pre-2016 IBM documentation
1033	IBM, 'IBM Spectrum Scale (formerly GPFS)' IBM Redbooks, May 2015
1034	NVM Express, 'NVM Express Revision 1.0' Mar. 1, 2011
1035	NVM Express, 'NVM Express Revision 1.2' Nov. 3, 2014
1036	Open-Channel Solid State Drives, 'NVMe Specification Revision 1.2' Apr. 2016

Exhibit No.	Description
1037	Trusted Computing Group, ‘Storage Security Subsystem Class: Opal Specification Version 2.00’ Feb. 24, 2012
1038	T10, “SCSI Block Commands - 3 (SBC-3),” Working Draft Revision 25, Oct. 27, 2010
1039	Fred Knight and David L. Black, ‘SBC-3 Thin Provisioning Commands’ T10 proposal 08-356r5, Jan. 14, 2009
1040	T10, ‘SCSI Block Commands - 4 (SBC-4)’ draft index, including pre-priority revisions in 2014-2016
1041	David A. Patterson, Garth Gibson, and Randy H. Katz, ‘A Case for Redundant Arrays of Inexpensive Disks (RAID)’ SIGMOD 1988
1042	Peter M. Chen et al., ‘RAID: High-Performance, Reliable Secondary Storage’ ACM Computing Surveys, 1994
1043	John Colgrove et al., 'Purity: Building Fast, Highly-Available Enterprise Flash from Commodity Components,' SIGMOD 2015
1044	EMC Corporation, 'XtremIO Data Protection (XDP): Flash-Specific Data Protection,' White Paper, July 2014
1045	U.S. Patent Application Publication No. 2004/0230795 to Armitano (“Armitano”)
1046	U.S. Patent Application Publication Nos. 2007/0244939 and 2007/0244940 to Devarakonda (“Devarakonda”)
1047	U.S. Patent Application Publication No. 2011/0022642 to deMilo (“deMilo”)

Exhibit No.	Description
1048	U.S. Patent Application Publication No. 2010/0145917 to Bone (“Bone”)
1049	U.S. Patent Application Publication No. 2010/0332401 to Prahlad (“Prahlad”)
1050	U.S. Patent Application Publication No. 2014/0297695 to Dejana (“Dejana”)
1051	IBM, ‘Understanding Hard Drive Media Defects White Paper – Servers’ IBM Product Documentation, Dec. 1997

I. INTRODUCTION

Meta Platforms, Inc. (“Meta”) and Samsung Electronics Co., Ltd. (“Samsung”) (together, “Petitioners”) respectfully request *inter partes* review of claims 1-2, 7, 9, 14-20, 32, 34, and 36 of U.S. Patent No. 10,776,023 (EX1001, “’023 Patent”).

For decades, data storage systems have used storage policies to determine how to store data. In the 1970s, for example, IBM implemented centralized, policy-based rules for its storage systems, which freed administrators from manually managing files. As storage systems advanced, so did storage policies, eventually allowing hosts (such as administrators, users, and client devices) to set storage policies by sending the policy to the storage device as opposed to relying on a policy set at manufacture. By 2016, storage policies, including user-configurable policies, were common and well-known.

Accordingly, during prosecution, the Examiner repeatedly rejected the pending claims of the ’023 Patent, directed at storage devices, such as solid-state drives (“SSDs”) implementing storage policies, as anticipated and obvious in view of U.S. Pat. Appl. Pub. No. 2015/0052167 to Olster et al. (EX1005, “Olster”). But the Applicant continued to argue, and the Examiner eventually allowed the claims after the Applicant grossly mischaracterized Olster and amended the claims to recite receiving “from the host” a storage policy that “indicates a recording method for

writing data to the storage media.” Critically, in allowing the claims, the Examiner conducted a prior art search using the terms “host or server or cpu or client,” (EX1002, 300) but overlooked that Olster, itself, discloses this well-known limitation, teaching “[t]he policies may be...distributed *from a client 102* to the data storage system 100,” including routine policies indicating a record method related to media defect treatment. EX1005, [0030], [0035]; *see also* EX1001, cl. 4 (“the recording method indicated by the at least on storage device policy further comprises instructions related to media defect treatment on the storage media”).

As such, Petitioners respectfully request institution on the challenged patent—a patent that should have never issued. *See Eunsung Global Corp. v. HydraFacial LLC*, IPR2025-00445, Paper 14, 3 (July 10, 2025) (“Although the patent examiner determined that [the asserted art] was pertinent but did not disclose the allowable features, Petitioner persuasively demonstrates that the patent examiner overlooked certain teaching in [the asserted art] that appear to disclose the allowable features of the claims.”).

II. MANDATORY NOTICES

A. Real Parties-in-Interest

The real parties-in-interest are Meta Platforms, Inc., Samsung Electronics Co., Ltd., Samsung Electronics America, Inc., and Samsung Semiconductor, Inc.

B. Related Matters

U.S. Patent Office records indicate that the '023 Patent is assigned to Gaea, LLC (“Patent Owner”), which is currently asserting the '023 Patent in the following concurrent litigations: *Gaea, LLC v. Meta Platforms, Inc.*, 4:26-cv-00348 (N.D. Tex) (filed on March 23, 2026) and *Gaea, LLC v. Samsung Electronics Co., Ltd.*, 2:26-cv-00264 (E.D. Tex.) (filed on March 31, 2026).

There are three patents asserted in each of the concurrent litigations. Petitioners have filed petitions for *inter partes* review against the other asserted patents at substantially the same time: U.S. Patent No. 11,907,553 (IPR2026-00426), and U.S. Patent No. 12,265,715 (IPR2026-00427).

C. Notice of Counsel and Service Information

Petitioners provide the following designation of counsel:

Lead Counsel
Lisa K. Nguyen (Reg. No. 58,018) Paul Hastings LLP 1117 S. California Avenue Palo Alto, CA 94303 Telephone: 650.320.1800 Fax: 650.320.1900 PH-META-Gaea@paulhastings.com

Back-Up Counsel	
David Tennant (Reg No. 48,362) Paul Hastings LLP 2050 M. Street NW Washington, D.C. 20036 Telephone: 202.551.1997 PH-META-Gaea@paulhastings.com	Joshua Nightingale (Reg. No. 67,865) Jones Day 500 Grant Street, Suite 4500 Pittsburgh, PA 15219 Telephone: 412.394.7950 Fax: 412.394.7959 jrnightingale@jonesday.com
Grace I. Wang (Reg. 69,892) Paul Hastings LLP 200 Park Avenue New York, NY 10166 Telephone: 212.318.6000 Fax: 212.319.4090 PH-META-Gaea@paulhastings.com	Stephanie Mishaga (Reg. No. 75,378) Jones Day 4655 Executive Drive, Suite 1500 San Diego, CA 92121 Telephone: 858.703.3140 Fax: 844.345.3178 smishaga@jonesday.com
Sasha Vujcic (Reg. 76,485) Paul Hastings LLP 4655 Executive Drive Suite 350 San Diego, CA 92121 Telephone: 858.458.3000 Fax: 858.458.3005 PH-META-Gaea@paulhastings.com	Robert Breetz (Reg. No. 75,550) Jones Day 901 Lakeside Ave. Cleveland, OH 44114 Telephone: 216-586-7213 Fax: 216.579.0212 rbreetz@jonesday.com
Allison Greubel (Reg. No. 82,758) Paul Hastings LLP 1117 S. California Avenue Palo Alto, CA 94303 Telephone: 650.320.1800 Fax: 650.320.1900 PH-META-Gaea@paulhastings.com	

Pursuant to 37 C.F.R. § 42.10(b), a Power of Attorney accompanies this petition. Petitioners consent to electronic service by e-mail.

D. Fee for *Inter Partes* Review

The Director is authorized to charge the fee specified by 37 C.F.R. § 42.15(a) to Deposit Account 50-2613.

E. Grounds for Standing

Petitioners certify pursuant to 37 C.F.R. § 42.104(a) that the '023 Patent is available for *inter partes* review and that Petitioners are not barred or estopped from requesting an *inter partes* review challenging the patent claims on the grounds identified in this Petition.

III. IDENTIFICATION OF CHALLENGES

Ground	Claims	Statute	Prior Art
1	1, 7, 9, 14-15	35 U.S.C. § 103	Olster ¹
2	1-2, 7, 9, 14-15, 32, 34, 36	35 U.S.C. § 103	Olster-Davis ²
3	1, 9, 14	35 U.S.C. § 103	Hashimoto ³
4	7, 15-20	35 U.S.C. § 103	Hashimoto in view of Kilian ⁴

¹ Olster is prior art under 35 U.S.C. §§ 102(a)(1), 102(a)(2), and 102(b)(1).

² Davis is prior art under 35 U.S.C. §§ 102(a)(1), 102(a)(2), and 102(b)(1).

³ Hashimoto is prior art under 35 U.S.C. §§ 102(a)(1), 102(a)(2), and 102(b)(1).

⁴ Kilian is prior art under 35 U.S.C. §§ 102(a)(1), 102(a)(2), and 102(b)(1).

IV. BACKGROUND

A. Technology Overview

For decades, storage devices have used storage policies to determine how to store data. EX1003, ¶¶25-27. In the 1980s, user-configurable storage policies garnered popularity with the advent of hierarchical storage management. *Id.*, ¶¶35-38. These systems allowed users or administrators to define rules for automatically migrating data. *Id.* For example, IT administrators could direct storage devices to store data in different tiers, storing hot data (frequently used data) in the more expensive disk drives in one tier, while cold data (less frequently used or LFU data) in cheaper, long-term tape archives in another tier. *Id.*

In the late 1980s and 1990s, the invention of NAND memory (the chip inside SSDs) revolutionized data storage, allowing data to be written in small blocks rather than all at once, leading to cheaper, higher-capacity storage. *Id.*, ¶¶25-27. With cheaper storage, more storage policies included rules and instructions related to redundancy schemes like RAID (Redundant Arrays of Independent Disks), which defined how content is stored, protected, and accessed across multiple drives. *Id.*, ¶¶39-40. Specifically, RAID can process content for storage by mirroring content (storing an exact copy of data in another drive), striping data (dividing data into blocks for storage), and calculating parity information (information that can be used

to rebuild the data if a drive fails). *Id.* The advent of SSDs allowed RAID to be implemented more cost-effectively. *Id.*

Thus, by 2016 (the priority date of the '023 Patent), storage policies, including user-configurable policies specifying recording methods for writing content to the storage media, were well-known, routine, and conventional.

B. '023 Patent

The '023 Patent describes well-known “data storage devices with configurable policy-based read and write behavior.” EX1001, 1:18-20. The '023 Patent complains that in prior art storage devices, storage decisions were “largely made by the manufacturer at the time of design and/or manufacture” providing users “very little to no influence” on how data is stored. *Id.*, 1:37-41. The '023 Patent purportedly addresses that problem using a routine storage device policy that allows “a user to customize the operation of the storage device. *Id.*, 3:13-15.

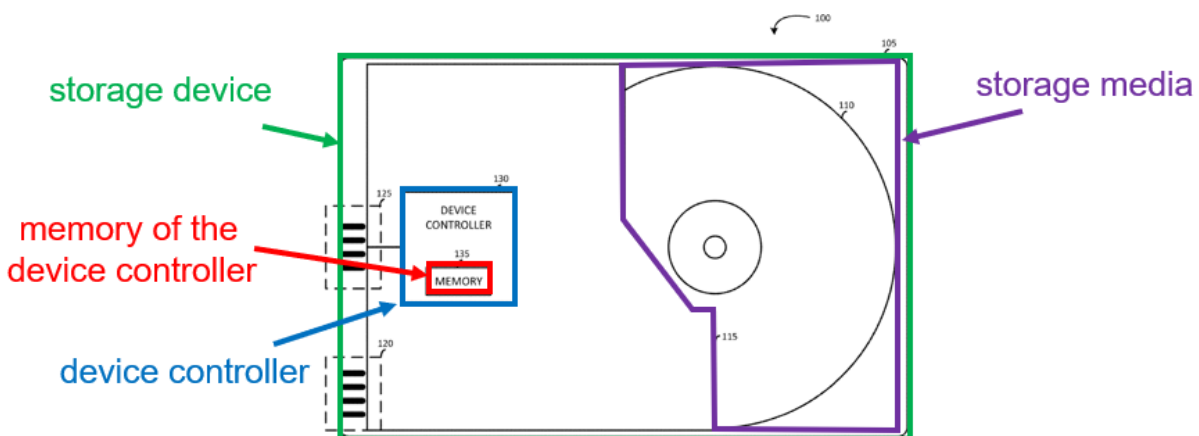
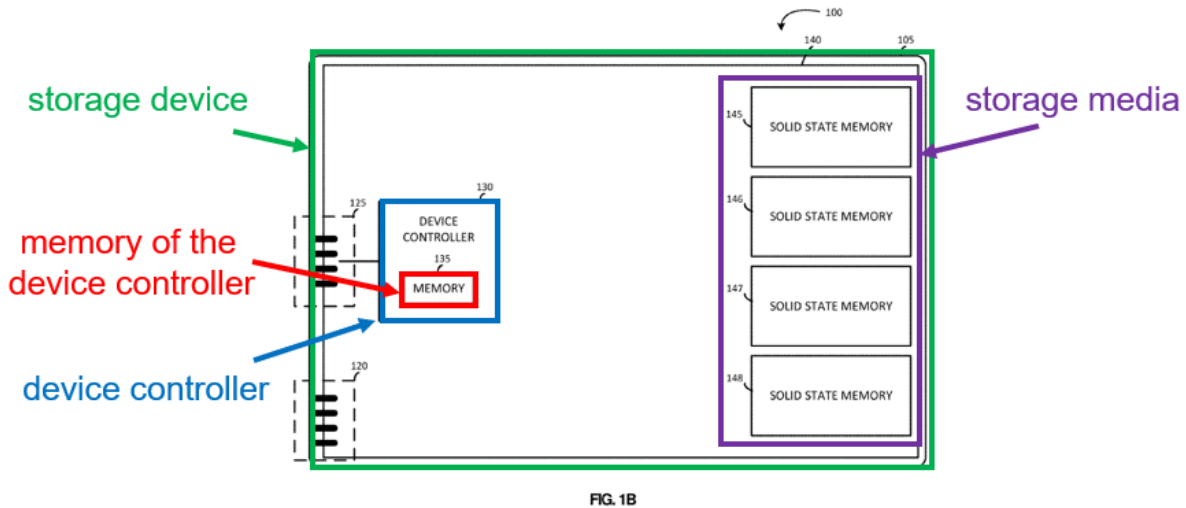


FIG. 1A

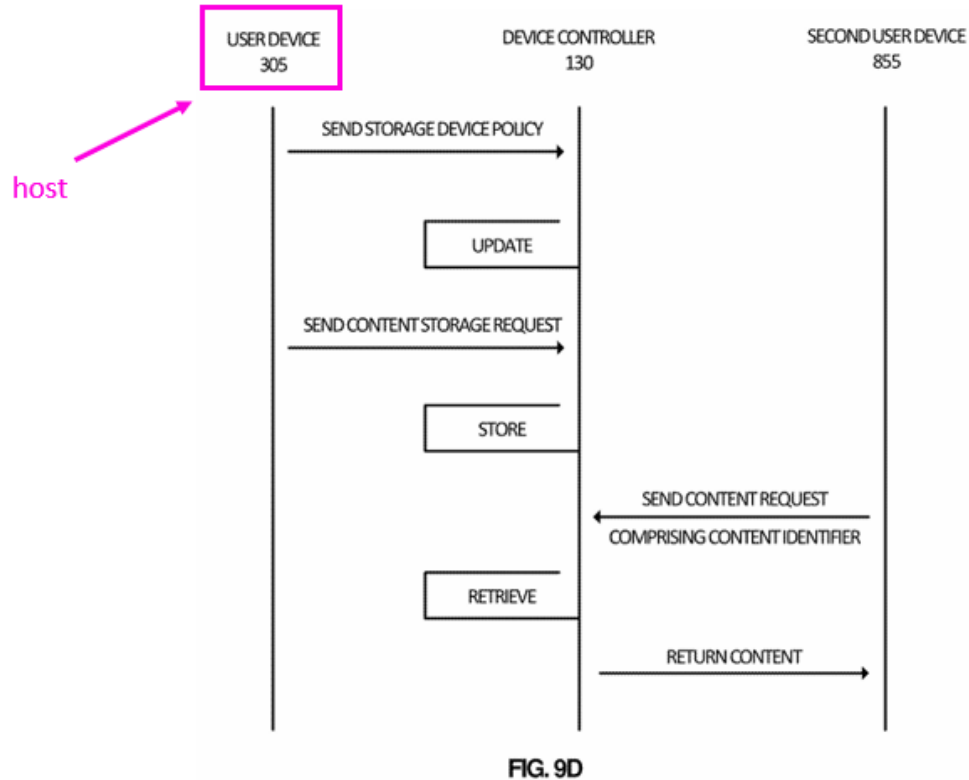
EX1001, Fig. 1A (annotated)



EX1001, Fig. 1B (annotated)

The claimed **storage devices** include a **device controller** and **storage media**, such as hard disk drives (Figure 1A) or solid state memory (Figure 1B). *Id.*, FIGS. 1A, 1B, 1:45-46. The '023 Patent discloses that the device controller 130 can receive a storage device policy from a host, allowing the policy to be user-configurable. “[T]he device controller 130 uses a storage device policy...to determine where to store the content.” *Id.*, 14:42-44. The '023 Patent explains that “[t]he determination can relate partially to the content being stored, and partially to the storage media 110.” *Id.*, 14:44-46. “For example, if the content contains an indication of priority level, such as information indicating that certain content is highly critical, the device controller 130, based on the storage device policy, may determine that the content should be stored on the storage media 110 at a low density, so the likelihood of data corruption will be minimal.” *Id.*, 14:46-52. As another example, “the storage device

policy may instruct the device controller 130 to avoid storing highly critical content in sectors that have defects.” *Id.*, 14:53-56.

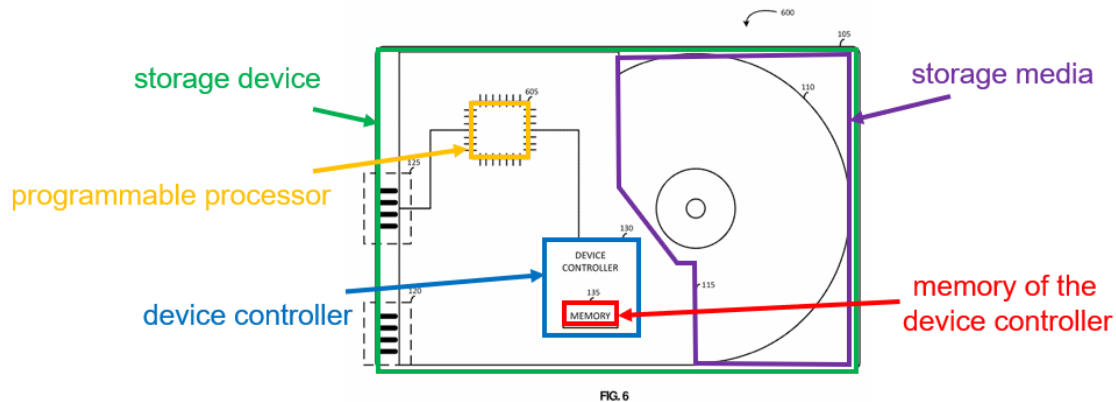


EX1001, Fig. 9D (annotated)

In Figure 9D, the '023 Patent describes one implementation of the operational sequence of this system of writing and reading content according to a storage device policy. *Id.*, FIG. 9D, 23:14-24:22. The device controller 130 receives a storage device policy from the host (here, **user device 305**). *Id.*, 23:15-17. Subsequently, the host “then sends a request to the device controller 130 to store content.” *Id.*, 23:33-35. Based on the storage device policy, the device controller determines “how to store the content” and “writes the content to the storage media.” *Id.*, 23:43-45,

23:52-53.

The device controller 130 “can also store a record of the write (the storage information) in **memory 135**,” which “can include a unique identifier for the content.” *Id.*, 23:54-58. The device controller 130 can then receive “a content request” and “retrieves the requested content according to the storage device policy” and storage information. *Id.*, 23:64-65, 24:19-20, cls. 1, 9.



EX1001, Fig. 6 (annotated)

In another embodiment, the **storage device** additionally includes a **programmable logic device 605**, or programmable processor, that “can be programmed by user of the drive 600 for any operation need that is desired.” *Id.*, 11:21-23. Accordingly, the programmable processor can implement the storage policy, but also other functions such as “custom encryption through hardware” or “compression.” *Id.*, 11:23-41. Code (or program instructions) stored on the **memory of the device controller** can then direct the device controller to store the processed content utilizing the recording method specified by the storage device policy. *Id.*,

11:61-63.

The claimed system can further include other nominal features such as a write-protection policy that prevents data from being deleted upon a delete request. *Id.*, 22:29-43.

C. The Challenged Claims

The challenged claims are entitled to an effective date of no earlier than November 7, 2016.⁵ The '023 Patent has 36 claims, including six independent claims (apparatus claims 1, 9, 16, 21, 26, 32) and 30 dependent claims. This Petition challenges four independent claims and 10 dependent claims.

Independent claims 1 and 9 each include two components: a storage media and device controller. The controller writes and reads content based on a storage policy from a host. Specifically, the controller is configured to (1) receive a storage policy from a host that indicates a recording method for writing content to the storage media, (2) store content on the storage media according to the storage policy, (3) record storage information for the content (including an identifier) in the device controller memory, (4) receive a request for the content, (5) retrieve the storage

⁵ Petitioners do not concede that any challenged claim is entitled to this priority date. For the purpose of this Petition, it is unnecessary to break the priority chain to a later date.

information for the content, (6) recover the content according to the storage information, and (7) return the content to the host. For claim 1, the controller is additionally configured to receive a storage request from the host. For claim 9, the controller additionally recovers the content according to the storage policy.

Independent claim 16 includes the same two components. In addition to indicating a record method, the storage policy also includes a write-protection policy. The controller receives a storage request and a delete request, and in response to the delete requests, indicates that the content could not be deleted in accordance to the write-protection policy.

Independent claim 32 includes the same two components, where the recording method corresponds to a content type.

D. Prosecution History

During the prosecution of the '023 Patent, the Applicant struggled to identify allowable subject matter disclosed in the specification of the '023 Patent. On May 3, 2019, the Applicant faced a non-final rejection of all pending claims on both §§ 102 and 103 grounds over Olster. Specifically, the Examiner found Olster disclosed the “replication and placement component of a policy defines the replication and placement of data objects in the data storage system” anticipating the independent claims, and that the “remaining dependent claim features are expressly or inherently found in the applied prior art.” EX1002, 122-126. The Examiner also

found the dependent claims obvious, stating each of “[t]he dependent claim features, while part of the invention, appear to be well known and their relevance not essential to the main invention found in the independent claim(s).” EX1002, 127.

On October 3, 2019, in an attempt to overcome the prior art, the Applicant amended the claims to recite “at least one storage device policy [that] indicates at least a storage density.” EX1002, 163-64. This amendment failed as storing data in less dense locations (where defects were less likely to occur) was well-known. On October 5, 2019, less than a month later, the Examiner issued a final rejection of all claims on § 103 grounds over Olster in view of US 2002/0036852 to Leow et al. *Id.*, 172-178. Notably, the Examiner acknowledged that recording methods based on density were known in the art, stating “storing or recording data in accordance with densities to manage errors/corruption/error rate on a disk appears to be well known as found in Leow.” *Id.*, 178.

On April 24, 2020, the Applicant decided to take a different tact—mischaracterizing Olster. The Applicant removed all references to storage density in the claims, and amended them to require that the controller receives “from a host” a storage policy that “indicates at least a recording method for writing content to the storage media.” *Id.*, 224. Critically, the Applicant stated incorrectly that Olster merely discloses “a searchable and distributed database” including “a controller for a storage node which distributes storage policies and handles search queries.” *Id.*,

233. Based on this distorted characterization, the Applicant asserted “*Olster* does not contemplate a storage controller or storage system which stores content on a storage media by utilizing a recording method, received from a host, for writing content to a storage media.” *Id.*

In response, the Examiner finally gave in and allowed the claims on May 11, 2020, issuing a notice of allowance with no reasons for allowance. *Id.*, 239-45. In doing so, the Examiner completely overlooked that *Olster* taught a storage policy with two components—not just a searchability component (which stores metadata about the data object in a database) but also a recording method component that “defines the replication and placement of data objects in the ***data storage system***” (not merely storing metadata in a database). *Cf id. to* EX1005, [0031]. The ’023 Patent subsequently issued on September 15, 2020.

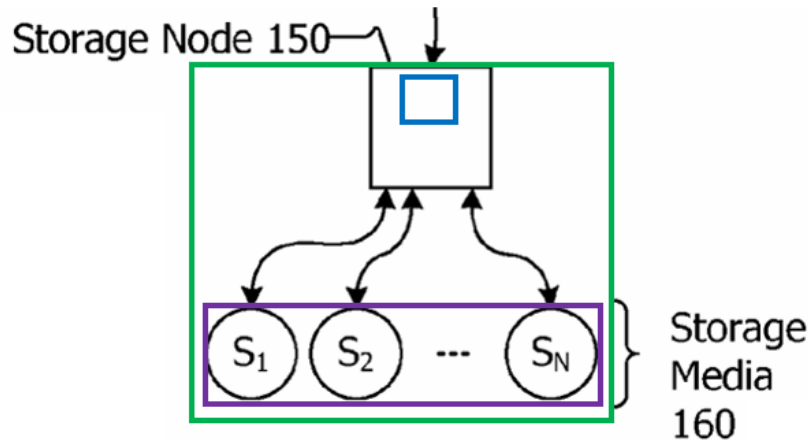
Importantly, there is no evidence in the record confirming that the Examiner considered two critical disclosures in *Olster*. *See generally* EX1002. First, *Olster* discloses that its storage policies are received *from the host*, teaching “[t]he policies may be...distributed from a ***client 102 [i.e., host]*** to the data storage system 100.” EX1005, [0030]. Indeed, the EAST Search History establishes that the Examiner searched for “host” and equivalent terms (specifically “host or server or cpu or ***client***”), but did not reference or acknowledge that *Olster* expressly disclosed receiving a policy “from the host.” Second, *Olster* discloses that its storage policies

indicate a recording method for writing content to the storage media, teaching that the replication component of its storage policy indicates “redundancy such that access to data is possible when a zone goes down or is impaired or unreachable” and may employ “a RAID or other configuration.” EX1005, [0035]. Indeed, the Examiner overlooked that replication and redundancy is a type of media defect treatment that is recited in a dependent claim. EX1001, cl. 4 (“recording method indicated by the at least one storage device policy further comprises instructions related to media defect treatment on the storage media”).

V. THE PRIOR ART

A. Olster (EX1005)

The Examiner materially erred by overlooking Olster’s disclosures. Olster discloses each and every limitation of independent claims 1 and 9, including a storage policy *from a host that indicates at least a recording method for writing content to the storage media*. Like the ’023 Patent, Olster discloses data storage devices with configurable policy-based read and write behavior. EX1005, [0002] (“This disclosure related to data stored in a data storage system”), [0030] (“Various policies may be maintained and distributed or known to the nodes in all zones in the distributed data storage system...and distributed from client 102”).

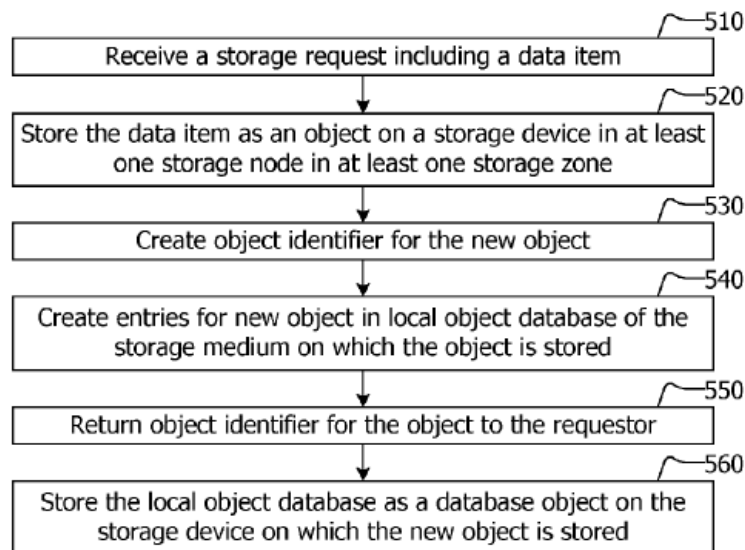


EX1005, Fig. 2 (excerpted and annotated)

Olster teaches a **storage node 150** that includes a **controller** and **storage media**, such as hard disk drives or flash memory. EX1005, FIG. 2, cl. 1 (“each node comprises a plurality of storage devices and a controller, the controller including a processor and memory”), [0024] (“The controller may communicate with and allow for communication with the storage media”), [0029] (“Storage media include magnetic media such as hard disks and tape, flash memory”).

Like the '023 Patent, Olster discloses that the controller can receive a storage policy from a *host* such as **client 102** or a **system administrator**, allowing the policy to be user-configurable. *Id.*, [0030] (“The policies may be...distributed from a client 102 to the data storage system 100...and to all nodes in the data storage system.”), [0046] (“In another implementation, the specific distribution instructions...may be hard coded or system defined or *earlier set by a system administrator.*”) (emphasis added). “[F]iles and other data may be partitioned into

smaller portions and stored as multiple objects among multiple storage media 160 in a storage node 150...according to [a] storage policy.” *Id.*, [0030]. Olster explains that the policy can include “a specific redundancy scheme identifier.” *Id.*, [0046]. For example, the policy “may operate as a fully replicated distributed data storage system in which all data is replicated.” *Id.*, [0034]. Or the policy may “provide[] for partial replication...to provide a limited amount of redundancy such that access to data is possible when a zone goes down or is impaired or unreachable, without the need for full replication.” *Id.*, [0035]. Olster further discloses that the policy may indicate other “techniques such as by a RAID or other configuration.” *Id.*



EX1005, Fig. 5

Like the '023 Patent, Olster discloses a method of writing and reading data according to a storage policy. *Id.*, FIGS. 5 (storage request), 6 (content request), [0046]-[0051] (“a storage request...may specify the storage policy for the data

item”), [0052]-[0054] (“The server then provides...the objects themselves”). “[T]he system stores the data item as an object according to a policy...provided with the request or [] included in the system such that the data item is stored on at least one storage device.” *Id.*, [0047].

Like the ’023 Patent, “[t]he system creates an object identifier for the new object.” *Id.* The “objects [including the data portion]...may be referenced by object identifier or OIDs.” *Id.*, [0040]. The object identifier includes information that “may be used to map the object to a particular...storage device” and “the system stores a portion of mapping information in memory.” *Id.*, [0041], [0047].

B. Davis (EX1008)

To the extent that Olster does not disclose a storage policy that *indicates at least a recording method for writing content to the storage media*, this would have been obvious in view of Davis.

Like the ’023 Patent, Davis discloses a policy for writing data to flash memory based partially on the content being stored, and partially on the storage media. EX1008, Abstract. For example, Davis teaches that hot data that has priority may be stored on flash memory having a lower number of bits per cell to minimize the likelihood of data corruption. *Id.*, 11:11-15; *see also id.*, 11:7-11

As another example, Davis teaches a policy that instructs the controller to avoid storing critical data more sensitive to corruption, such as text data, in memory

with more defects. *Id.*, 9:37-39 (“One policy is writing text data to a type of flash memory having a lower read error rate and writing image data to a type of flash memory having a higher read error rate.”); *see also id.*, 9:39-45.

C. Hashimoto (EX1006)

Hashimoto discloses each and every limitation of independent claims 1, 9, and 16, except that Hashimoto does not expressly disclose that its storage policy *comprises at least a write-protection policy* and receives and responds to a *delete request* as required by claim 16. Hashimoto teaches a storage device that addresses the demand for “a host of the storage device...to more actively manage and control operations of the storage device.” EX1006, [0004].

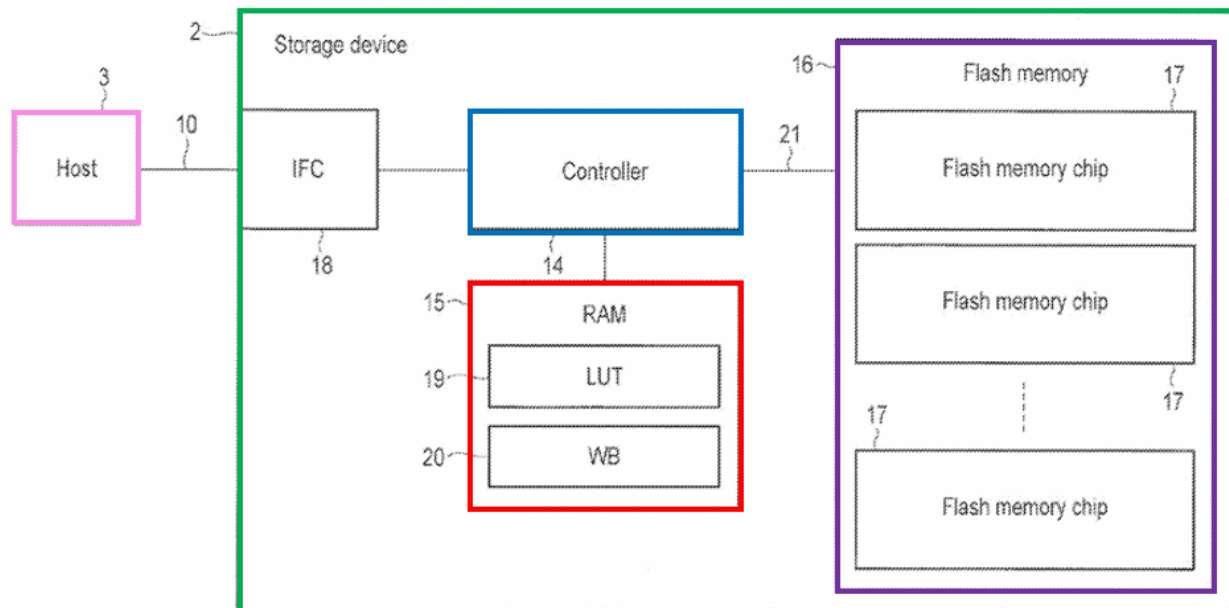


FIG. 4

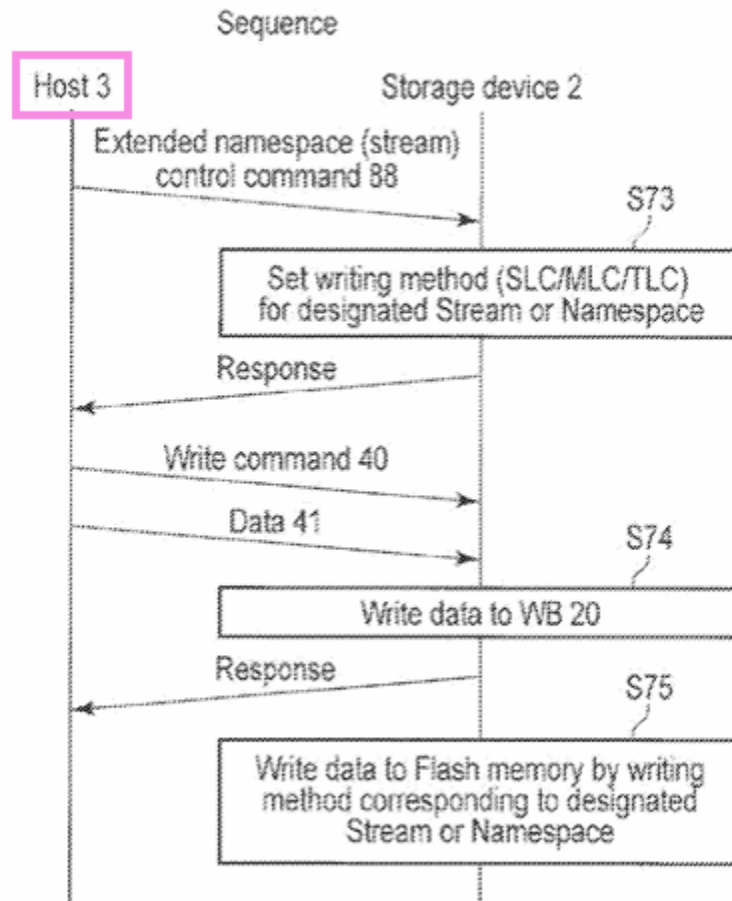
EX1006, Fig. 4 (annotated)

Hashimoto teaches a storage device 2 that includes a controller 14 and flash memory 16. EX1006, [0108] (“The storage device 2 includes a controller 14, a RAM 15, a flash memory 16, and an interface controller (IFC) 18.”), [0109] (“The flash memory 16 functions as a nonvolatile memory” and “includes one or more flash memory chips 17.”), FIG. 4.

Like the '023 Patent, Hashimoto discloses that the controller can receive commands (such as command 88) from host 3 that allows the host to configure and set how the data is stored on memory. *Id.*, [0096] (“The host 3 (host software) can manage and control the storage devices 2 by transmitting various commands to the storage device 2”), FIG. 42 (“set writing method (SLC/MLC/TLC)”). Hashimoto discloses that these host-supplied settings for writing and reading data to flash memory can relate partially to the data being stored, based partially on the flash memory. *Id.*, [0195] (“The controller 14 of the storage device 2 writes the data to the flash memory 16 by employing the writing method designated by the tier attribute.”).

For example, Hashimoto discloses a tier control feature in which the controller 14, based on the host-supplied settings, may determine that the content should be stored on the flash memory with a low density, so the likelihood of data corruption will be minimized. *Id.*, [0193] (“The host 3 can designate a tier attribute which should be used to write the data, by using a command for In-drive tiering control

(the extended write command or the extended namespace (stream) control command).”), [0191]-[0192] (“In the first tier, a 1-bit/cell writing method (SLC writing method) of writing 1-bit data to each memory cell is employed,” and “[i]n the second tier, a writing method of writing data of two or more bits to each memory cell is employed. The writing method employed in the second tier may be a 2-bit/cell writing method (MLC writing method), a 3-bit/cell writing method (TLC writing method), or a 4-bit/cell writing method (QLC writing method).”).



EX1006, Fig. 42 (excerpted and annotated)

Like the '023 Patent, Hashimoto discloses executing write and read operations based on these host-supplied settings. In Figure 42, Hashimoto teaches at step S73, that the controller 14 of storage device 2 receives command 88 from host 3 to apply the tier control feature for a certain data stream and sets the writing method accordingly. *Id.*, [0491]. Subsequently, the host sends write command 40 to write data 41. *Id.*, [0492]. At step S75, the controller 14 “[w]rite data to Flash memory by writing method corresponding to designated Stream or Namespace” according to the host-supplied settings. *Id.*, Fig. 42; [0492].

Like the '023 Patent, Hashimoto discloses that controller 14 stores a record of the write in **RAM 15**. Specifically, the data is identified using a logical block address (LBA), and the LBA to physical mapping (also called L2P address mapping) is stored in the look-up table (LUT) in RAM 15. *Id.*, [0110] (“mapping information indicating relationship between logical addresses and physical addresses of the flash memory 16”). The controller 14 can then receive a read command and retrieve the requested data according to the LBA (to identify the data) and the host-supplied settings (to determine the media type to read from which will dictate processing such as ECC decoding). *Id.*, [0135], [0521], [0555]-[0556].

D. Kilian (EX1007)

Killian discloses a storage system that applies a *write protection policy* and receives and responds to a *delete request* as required by claim 16. Kilian states that

its host provides “a retention period that indicates a period of time for which the data may not be deleted,” and the controller enforces this policy: when it “receives a request to delete the data, it first evaluates the retention period...[and i]f the retention period has not expired, the storage system denies the request.” EX1007, Abstract.

VI. LEVEL OF ORDINARY SKILL IN THE ART

A person of ordinary skill in the art (“POSITA”) in 2016 would have had at least a bachelor’s degree in electrical engineering, computer science, computer engineering, or a similar technical field, with at least two years of experience in the field of data storage systems. Additional experience could substitute for less education, and vice versa. EX1003, ¶¶55-57.

VII. CLAIM CONSTRUCTION

The Board construes the claims “using the same claim construction standard that would be used” in district courts. 37 C.F.R. §42.100(b). This Petition establishes the prior art meets each of the claim limitations under its plain and ordinary meaning.⁶ Thus, the claim terms require no express construction.

⁶ Petitioners reserve the right to provide an express construction consistent with that applied in this proceeding and the Parallel Litigation. *See Caption Health, Inc. v. The Univ. of British Columbia*, IPR2025-01422, Paper 15, 3 (Director Dec. 18, 2025).

VIII. GROUND 1

Olster renders obvious claims 1, 7, 9, 14, and 15. EX1003, ¶¶71-76. Olster describes embodiments of a searchable data storage system that includes storage zones that are autonomous from one another and contain storage nodes, such as solid-state drives, that include storage media. EX1005, Abstract. Each storage node includes a controller comprising a processor and memory. *Id.*, [0024], [0025], cls. 1, 10.

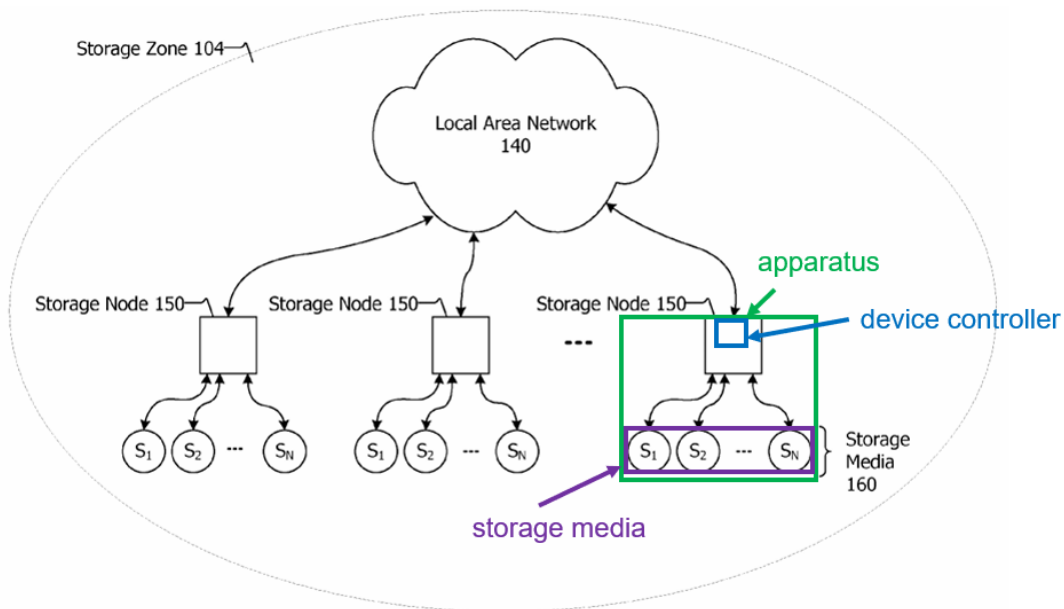


FIG. 2

EX1005, Fig. 2 (annotated)

The data storage system reads and writes data called data objects or data items to “storage nodes...according to the storage policy.” *Id.*, [0030]. The “policies have two components: [1] a replication and placement component and [2] a searchability component.” *Id.*, [0031].

Olster discloses each limitation of claims 1, 7, 9, 14, and 15 as features in a single embodiment. Accordingly, no motivation to combine or reasonable expectation of success is required. *Realtime Data, LLC v. Iancu*, 912 F.3d 1368, 1372-73 (Fed. Cir. 2019); *Guardant Health, Inc. v. Univ. of Washington*, No. 2024-1129, 2026 WL 184334 (Fed. Cir. Jan. 23, 2026). Regardless, a POSITA would have been motivated to combine the disclosed features in Olster in a single system with a reasonable expectation of success. EX1003, ¶¶71-76.

A POSITA would have been motivated to implement Olster’s data storage system to allow an administrator or client to configure or program the storage policy as appropriate. EX1003, ¶¶73-75. A POSITA would have had a reasonable expectation of success as Olster itself teaches that distributing the storage policy through an administrator or client is an option for its data storage system. *Id.*, ¶76. Specifically, because the storage policy is stored in the memory of the controller, the policy can be updated through communications sent to the storage node. *Id.*

A. Independent Claims 1, 9

1. 1[pre]/9[pre]

To the extent limiting, Olster discloses the preamble. EX1003, ¶¶77-78.

Olster discloses a storage node (*apparatus*), such as a solid-state drive. EX1005, [0022] (“[In] FIG. 2, the storage zone 104 includes a plurality of storage nodes 150 which include a plurality of storage media 160.”). Olster teaches that the

collection of components comprising the storage node can be included in an enclosed space with other storage nodes or alone. *Id.*, [0024], [0026]; EX1003, ¶78.

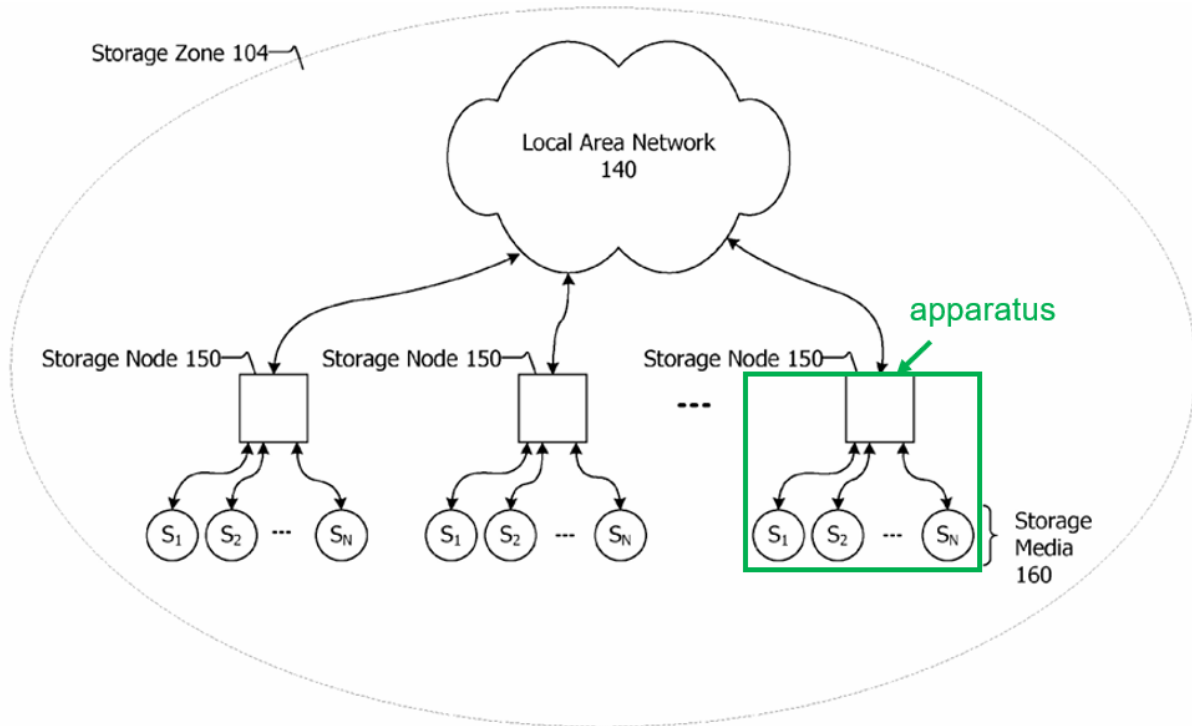


FIG. 2

EX1005, Fig. 2 (annotated)

2. 1[a]/9[a]

Olster discloses this limitation. EX1003, ¶¶79-81. Olster's storage node 150 (*apparatus*) includes a storage media 160 (also referred to as a storage device) (*storage media*). EX1005, FIG. 2, [0022]. Olster explains: “[s]torage media include magnetic media such as hard disks and tape, flash memory, and optical disks.” *Id.*, [0029]; EX1003, ¶80. Olster also refers to storage media as storage devices. EX1005, Abstract (“The nodes include storage devices”), [0023], cl. 1 (“each node comprises a plurality of storage devices and a controller”); *see also id.*, [0024];

EX1003, ¶81.

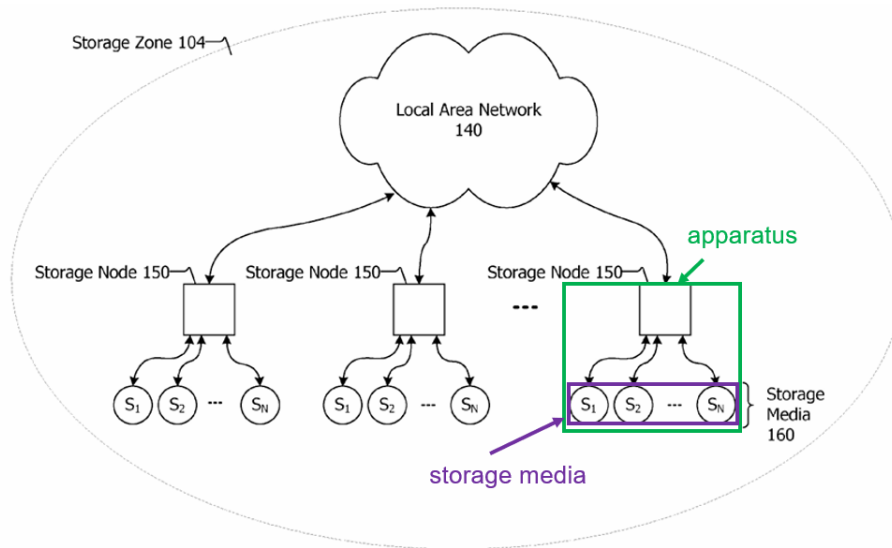


FIG. 2

EX1005, Fig. 2 (annotated)

3. 1[b]/9[b]

Olster discloses this limitation. EX1003, ¶¶82-85. Olster discloses that the storage node 150 (*apparatus*) includes a controller (*device controller*). EX1005, [0025] (“a controller for a node”), cls. 1 (“each node comprises a plurality of storage devices and a controller”), 10. “When [] in a single cabinet or rack, storage nodes and/or constituent storage media may be coupled with a backplane.” *Id.*, [0024]. The controller further “may be coupled with a backplane” and “may communicate with and allow for communications with the storage media” (*device controller operatively coupled with the storage media*). *Id.*, [0024].

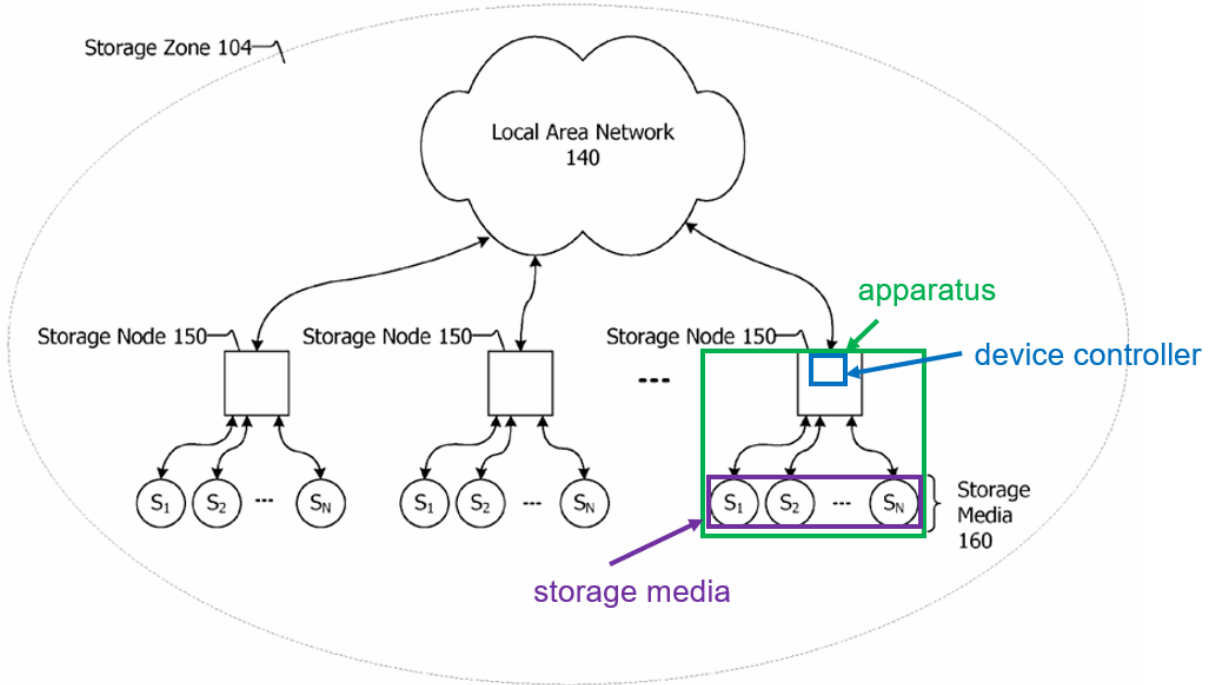


FIG. 2

EX1005, Fig. 2 (annotated)

The controller is configured to store and retrieve the data objects (*content*) to and from storage media 160 (*storage media*) as recited in 1[b.i]/9[b.i]-1[b.viii]/9[b.vii]. See Sections VIII.A.4-11.

4. 1[b.i]/9[b.i]

Olster discloses this limitation. EX1003, ¶¶86-94. Olster discloses that the controller of storage node 150 (*device controller*) can *receive* from client 102 or a system administrator (*from a host*) the storage policy (*at least one storage device policy*) including a replication component that includes instructions for a redundancy scheme such as full replication, partial replication, or RAID, to be applied when storing data portion of the data object to the storage media 160 (*indicates at least a*

recording method for writing content to the storage media). *Id.*

Olster discloses data for storage (*content*) referred to as the data portion of data objects or data items. EX1005, [0006] (“may store large numbers of documents, images, audio, videos and other data as objects.”). Olster further discloses *at least one storage device policy* because Olster teaches “[t]he object including its constituent data portion may be stored among storage nodes and storage zones according to the ***storage policy***.” *Id.*, [0030]; EX1003, ¶87.

The policies (*at least one storage device policy*) indicates *at least a recording method for writing content to the storage media* because they include a “replication...component” that “defines the replication...of data objects in the data storage system.” *Id.*, [0031]; EX1003, ¶88. As Olster explains, “[t]he replication...component may include specific distribution instructions for the data item.” *Id.*, [0046].

The replication component of the policy provides redundancy through “full replication”, “partial replication”, a RAID policy, or other redundancy scheme. *Id.*, [0034]-[0035], [0046]. *Id.*, [0034]-[0035], [0046]. The replicas may be written before, after or during the write of data. *Id.*, [0034]. Olster discloses that the replication policy includes a RAID configuration. *Id.*, [0035] (“resiliency may be provided by using intra-zone techniques such as by a RAID or other configuration.”). The replication component may also include “a specific redundancy scheme

identifier” (*id.*, [0046]), which a POSITA would have understood to include a RAID-level policy, such as RAID 0, RAID 1, RAID 5, and RAID 6, applied to a SSD with multiple types of storage media as disclosed in Olster. EX1003, ¶¶88-89.

A replication of data objects is a *recording method for writing content to the storage media* because it provides rules for how the data is recorded in the storage media. EX1003, ¶¶90-92. Through configurations such as full replication, partial replication, or RAID, the replication component of the policy provides a method of writing the data such as striping the data (dividing it into blocks) and mirroring the data (making a copy of the data to store a redundant copy in other memory units). EX1005, [0034]-[0035]; EX1003, ¶¶90-91.

The replication component of the policy of Olster is also a type of media defect treatment expressly identified as *a recording method for writing data to the storage media* in a dependent claim of the ’023 Patent. Specifically, claim 4 of the ’023 Patent recites *the recording method indicated by the at least on storage device policy further comprises instructions related to the media defect treatment on the storage media*. The replication component of Olster’s policy manages defects in the storage media. EX1003, ¶92. As Olster explains, when a redundancy scheme is applied, “access to data is possible when a zone goes down or is impaired or unreachable.” EX1005, [0035]. If a portion (e.g., block or page) of the SSD is defective and the data is unrecoverable, data recovery and rebuild is done through

RAID or another redundancy scheme . EX1003, ¶92.

Olster discloses that the system receives the storage policies. *Id.*, [0046]. The controller is the part of the system that receives the policy because it is the part of the storage node that executes the storage request to store the data objects according to the storage policy (*at least one storage device policy*). *Id.*, [0026], [0047]; EX1003, ¶93.

Olster discloses that these storage policies (*storage device policy*) “may be...distributed from a client 102,” including “earlier set by a system administrator” (*receive, from a host*). *Id.*, [0030], [0046].

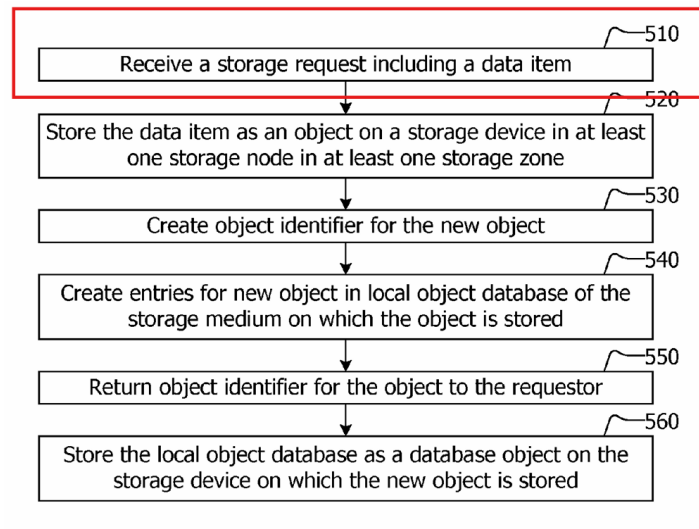
Although the Examiner considered Olster and ultimately allowed the claims after the Applicant amended the claims to recite receiving the storage policy *from the host* and *indicates a recording method for writing content to the storage media*, the Examiner materially erred by overlooking these disclosures in Olster. First, the Examiner did not cite to any Olster disclosure discussing redundancy or distribution from client 102 or the system administrator. EX1002, 119-129, 239-245. Second, the Applicant grossly mischaracterized Olster, arguing that its disclosures related to “a searchable and distributed database.” *Id.*, 233. Specifically, the Applicant made two incorrect distinctions. The Applicant argued that Olster “discloses storing ***an object*** among storage zones according to the storage policies” (emphasis added). *Id.*, 232. Thus, the Applicant incorrectly argued that the system stored the object

including the storage policies, not the content itself. The Applicant also argued that Olster “discloses a controller for a storage node which distributes storage policies and handles search queries.” *Id.*, 233. Thus, the Applicant incorrectly argued that the controller did not store data.

Viewed in context, without the misrepresentations of the Applicant, Olster discloses that its controller (*device controller*) receives a policy “distributed from a client 102” (*receives, from a host, at least one storage device policy*) that includes a replication component related to media defect treatment, such as a redundancy scheme to be applied to the data for storage on the storage node 160 (*indicates a recording method for writing content on the storage media*).

5. 1[b.ii]

Olster discloses this limitation. EX1003, ¶¶95-97. Olster discloses the system “receives a storage request” (*receive a storage request*). EX1005, [0046]. The part of the system that receives the storage request is the controller for the storage node 150 because it is “[t]he controller may communicate with and allow for communications with the storage media.” EX1005, [0026]; EX1003, ¶96.



EX1005, Fig. 5 (annotated)

This storage request is “received from a client application running on a client device” (*receive...from the host*). *Id.* [0046]. Further, the purpose of the storage request is to “store or put a data item” (*store content*) in the storage media, as demonstrated in block 510 in FIG. 5. *Id.*, [0046], FIG. 5; EX1003, ¶97.

6. 1[b.iii]/9[b.ii]

Olster discloses this limitation. EX1003, ¶¶98-101. Olster discloses the controller of storage node 150 (*device controller*) *stores* the data portion of the data objection on the storage media 160 (*store the content on the storage media*) using the redundancy scheme identified in the replication component of the policy (*by utilizing the recording method indicted by the at least one storage policy*).

As discussed, Olster uses “storage media” and “storage device” interchangeably. *See* Section VIII.A.2. Olster describes the controller storing a

“data item” (*content*) “as an object on a storage device” (*storage media*) “according to a policy (representing specific distribution instructions...)” (*recording method indicated by storage device policy*). EX1005, [0030] (“The object including its constituent data portion may be stored among storage nodes and storage zones according to the storage policy”), [0047], [0046]; *see* Sections VIII.A.4-5; EX1003, ¶100.

Specifically, the controller will store the data item in a location indicated by the distribution instruction in the policy, which “may specify...a specific redundancy scheme identifier, and the like.” EX1005, [0046]. By allowing client 102 or a system administrator to define the policy on how the data item is recorded, for example, within a particular node with other copies in different storage media, a user of Olster’s data storage system can balance any competing concerns such as power consumption and cost. EX1003, ¶101.

7. 1[b.iv]/9[b.iii]

Olster discloses this limitation. EX1003, ¶¶102-106. Olster discloses that the controller of the storage node 150 (*device controller*) stores the object specific information including metadata (*record storage information for the content*) on a memory included in the controller of storage node 150 (*memory of the device controller*), including the object identifier (OID) (*comprising at least a content identifier for the content*). *Id.*

Olster discloses that the controller includes memory (*memory of the device controller*). EX1005, cls. 1 (“each node comprises a plurality of storage devices and a controller, the controller including a processor and memory”), 10 (“each storage node comprises a plurality of storage devices and a controller, each controller including a processor and memory”), [0024] (“The controller may include a processor, volatile memory and non-volatile memory.”); EX1003, ¶103.

Olster discloses “object specific information including metadata” (*storage information for the content on the memory*). EX1005, [0040], [0032], [0038] (“metadata (also referred to as object specific information)”), [0039], [0048]; EX1003, ¶103.

Olster further discloses that “objects...may be referenced by object identifier or OIDs” (*content identifier*). *Id.*, [0040], [0049] (“the database entries for objects include an object identifier or OID”). The object specific information or metadata (*storage information*) includes the OID (*content identifier*). *Id.*, [0038]-[0039], [0053]; EX1003, ¶105.

Olster expressly returns the OID after storage and explains that objects may be referenced by OID. EX1005, [0041]. A later request for that object could use the OID as the field/value pair or object reference because the OID is the identifier Olster created and returned for that purpose. EX1003, ¶106. Olster explains that the OID can include a group identifier that “may be used to map the object to a

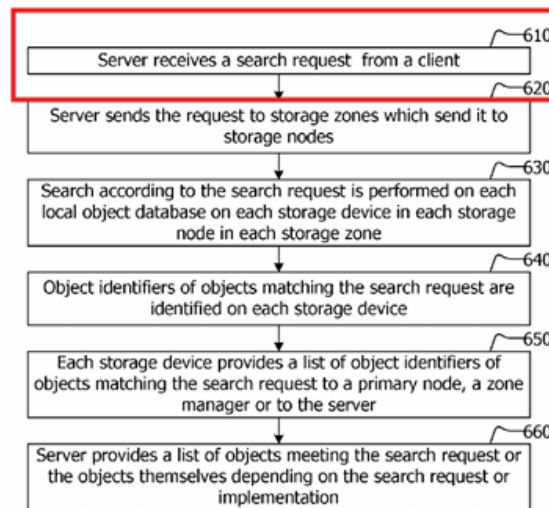
particular storage node or storage device,” and that this mapping information is a part of the object specific information including the metadata (*storage information*). *Id.*, [0041]. Olster further discloses that “the system stores a portion of mapping information” (part of the object specific information) (*storage information*) “in memory” (*memory of the device controller*). *Id.*; compare with *id.* (“the storage nodes hold a portion of the mapping information in their memory”). A POSITA would have understood that the system storing the a portion of mapping information (which is part of the metadata of an object) in the system’s memory is the controller *record[ing] storage information...on a memory of the device controller*. EX1003, ¶¶107-108

8. 1[b.v]/9[b.iv]

Olster discloses this limitation. EX1003, ¶¶109-113. Olster discloses that the controller of storage node 150 (*device controller*) *subsequent to* storing the data item (*storing the content*), can then *receive* a search request for the data item (*content request for the content*), including the object identifier (OID). *Id.*

Olster discloses that the controller (the part of the server in the storage node that processes requests) (*device controller*) *receives a* “search request from a client” (*content request*), as shown in block 610 below, that includes “at least one value [and] at least one field.” EX1005, [0052] (“FIG. 6 is a flow chart of the actions take to search for data [(*content*)] included in the data storage system”), [0022] (“Each of

the storage nodes 150 may include one or more server computers”), cls. 1, 10, FIG. 6; EX1003, ¶¶110-111.



EX1005, Fig. 6 (annotated)

Olster discloses that “metadata fields [] may be stored in the object database as object specific information” and that “[t]he field or entries in the object database may be stored as field/value pairs.” *Id.*, [0039]. The “at least one field [and] at least one value” that are included in Olster’s *content request* are object specific information/metadata “field/value pairs.” *Id.*, [0052]; EX1003, ¶112. The field/value pair in the *content request* is the OID (*content identifier*) because the OID is a type of object specific information/metadata. *See* Section VIII.A.7; EX1003, ¶112.

Olster discloses that the OID was previously returned to the requester during

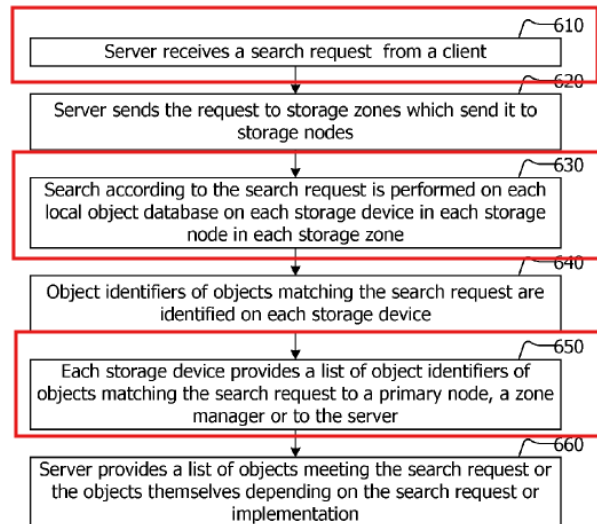
execution of the storage request. EX1005, [0051] (“After the object is stored and the OID is created, the OID may be returned to the requester as shown in block 550.”). A POSITA would have understood that the content must have been stored before a search request (*content request*) can return its OID (*content identifier*). *Id.*; EX1003, ¶113. Thus, the controller receives, as part of the search request (*content request*), the OID (*content identifier*) subsequent to storing the content. EX1003, ¶113.

9. 1[b.vi]/9[b.v]

Olster discloses this limitation. EX1003, ¶¶114-115. Olster discloses that the controller of storage node 150 (*device controller*) in response to the search request (*content request*), will retrieve from the memory of the controller (*memory of the device controller*), the object specific information include metadata that include the OID and mapping information (*storage information for the content*). *Id.*

As discussed above, Olster discloses the controller (the part of the server processing the request) receiving a “search request” (*content request*) including the OID (*content identifier*). See Section VIII.A.8. As discussed above, Olster discloses that the OID further includes a group ID “to map the object to a particular storage node or storage device.” EX1005, [0041]; see Section VIII.A.7. As discussed above, Olster discloses the mapping information derived from the group ID in the OID (*storage information for the content*) is stored on the system memory (*memory*

of the controller device); see Section VIII.A.7. A POSITA would have understood that *in response to the content request* (see FIG. 6, block 610), the controller *retrieves from the system memory (memory of the controller device)* (see FIG. 6, block 630), the mapping information (*storage information for the content*) (see FIG. 6, block 650) because the OID would be needed by the controller to locate the data item to be retrieved. EX1003, ¶115; see Section VIII.A.8 (explaining that the controller is the part of the server processing the request).



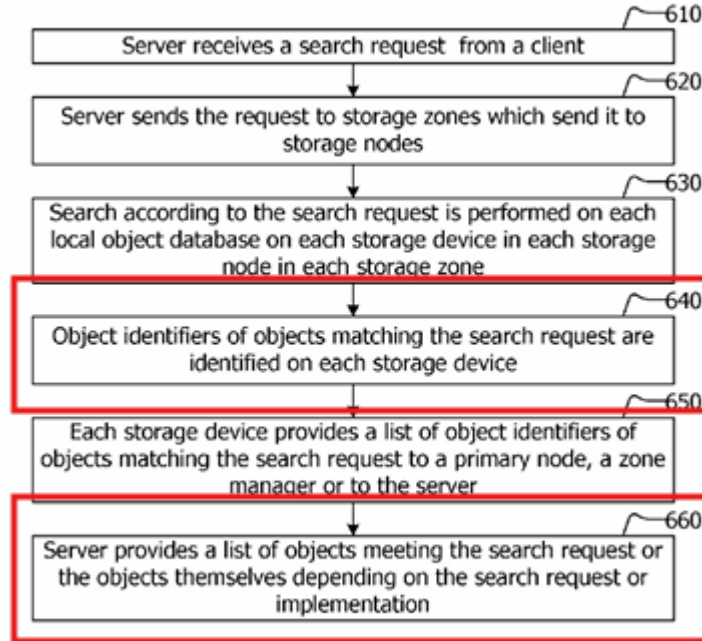
EX1005, Fig. 6 (annotated)

10. 1[b.vii]/9[b.vi]

Olster discloses this limitation. EX1003, ¶¶116-119. Olster discloses that the controller of storage node 150 (*device controller*) will *recover the object* including the data portion (*content*) *from storage media 160 (storage media) according to the* OID (part of the object specific information including metadata) and the redundancy

schemed indicated in the replication portion of the policy (*at least one storage device policy*). *Id.*

Olster discloses that the controller (*device controller*) stores the data objects (*content*) on the storage media 160 (*storage media*). *See* Section VIII.A.6. Olster discloses finding the “[o]bject identifiers of objects matching the search request...as shown in block 640.” EX1005, [0053]. After identifying the object on the storage device (used interchangeably with storage media, *see* Section VIII.A.2) (*storage media*) using the object identifier (*according to the storage information*), the controller (the part of the server that processes the requests), “then provides...the objects themselves,” including the data portion. *Id.*, [0054], FIG. 6. A POSITA would have understood the controller *recovers the* data portion of the objects (*content*) *from the* storage device (*storage media*) because this is necessary in order to provide the objects responsive to the search request to the host. EX1003, ¶117; *see* Section VIII.A.8 (explaining that the controller is the part of the server in the storage node that processes requests).



EX1005, Fig. 6 (annotated)

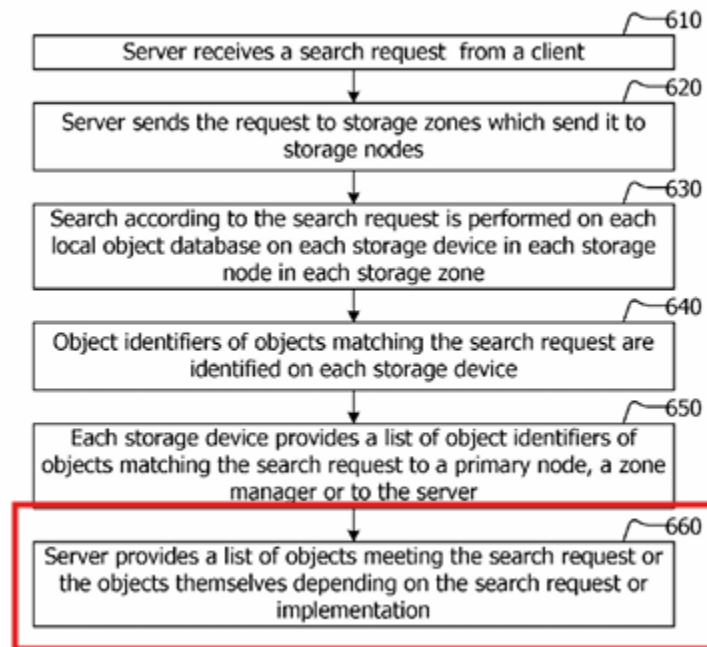
Olster discloses that the controller retrieves the object specific information including mapping information (*storage information*) using the group ID portion of the OID (*content identifier*). See Section VIII.A.9. As discussed above, Olster discloses that the mapping information (*storage information*) “map[s] the object to a particular storage node or storage device.” EX1005, [0041]. A POSITA would have understood that the controller *recovers* the data objects (*content*) from the storage device (*storage media*) *at least in part according to* the mapping information (*storage information*) because the OID would be needed by the controller to locate the data item to be retrieved. EX1003, ¶118.

For claim 9, Olster also discloses that the content is recovered *according to* the replication component of the policy (*the at least one storage device policy*) because the data will need to be recomputed or rebuilt as indicated by the redundancy policy. EX1005, [0034]-[0035], [0052]; EX1003, ¶119.

11. 1[b.viii]/9[b.vii]

Olster discloses this limitation. EX1003, ¶¶120-1211. Olster discloses that the controller of storage node 150 (*device controller*) can *respond to the* search request (*content request*) *by at least returning* the data object including the data portion (*content*) to the client.

In Figure 6, Olster depicts a “block 660” where the controller can provide “the objects themselves.” EX1005, FIG. 6, [0054]. Olster discloses that step 610 of Figure 6 includes the server (specifically, the controller of the server) *receiving* a search request (*content request*) *from the host*. See Section VIII.A.8. A POSITA would have understood that the controller *responds to the* search request (*content request*) *at least by returning the* objects (*content*) *to the host*. EX1003, ¶121; see Section VIII.A.8.



EX1005, Fig. 6 (annotated)

B. Dependent Claims 7, 14, 15

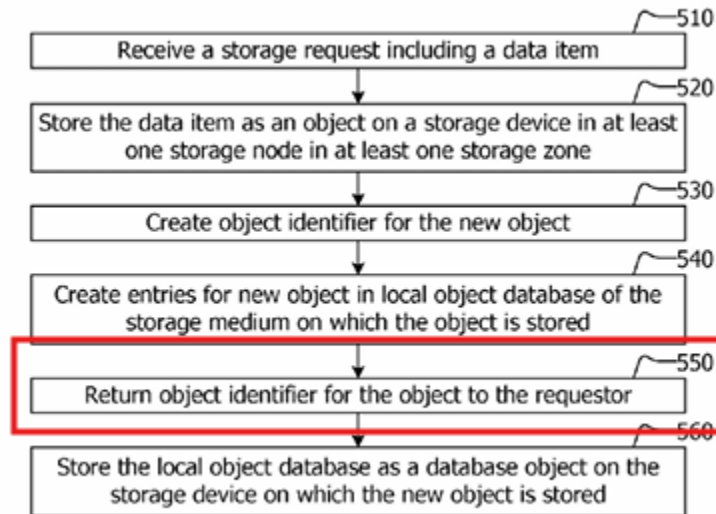
1. Claim 7

Olster discloses this limitation. EX1003, ¶¶122-124.

Olster discloses an “object identifier” or “OID” (*content identifier*). EX1005, [0051]. Olster discloses that the unique identifier portion of the OID (*content identifier*) may be “assigned by the controller” (*device controller*). EX1005, [0042]; *see also id.* [0040] (“The object identifier 400 includes a location identifier 404 and a unique identifier 406.”).

Olster discloses that “OID” (*content identifier*) is “returned to the requester.” *Id.*, [0051] (“After the object is stored and the OID is created, the OID may be

returned to the requester, as shown in block 550.”), FIG. 5. Olster discloses returning the OID (*content identifier*) to the requester (block 550) after receiving the storage request (block 510). *Id.*, [0046], [0051], FIG. 5.



EX1005, Fig. 5 (annotated)

2. Claim 14

Olster discloses this limitation. EX1003, ¶¶125-126. Olster discloses the storage media 160 includes “flash memory” and “silicon storage devices,” which are types of *solid-state memory devices*. EX1005, [0029]; EX1003, ¶126.

3. Claim 15

Olster discloses this limitation. *See* Section VIII.B.1; EX1003, ¶¶122-124, 127.

IX. GROUND 2

The Olster-Davis storage system would have included a modified version of the replication and placement component in Olster's storage policy that incorporate the data placement policies taught by Davis. EX1003, ¶¶128-129. Specifically, the replication component would include policies that write data to a type of flash memory based on the data type. EX1005, [0031]; EX1008, 1:48-51, 9:37-11:15 (describing various data placement policies), FIG. 4A; EX1003, ¶129.

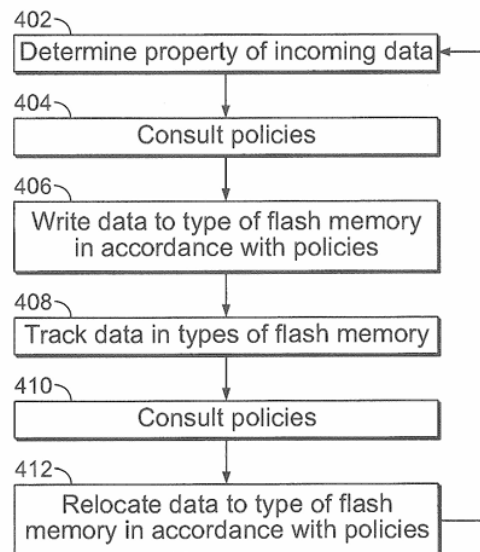


FIG. 4A

EX1008, Fig. 4A

A. Motivation to Combine

A POSITA would have been motivated to modify the replication and placement component of the storage policy of Olster (*at least one storage device policy*) to include the data placement policies taught by Davis, which records data to

a type of flash memory based on the data type (*at least a recording method for writing content to the storage media*). EX1003, ¶¶130-133.

First, both Olster and Davis are directed to data storage systems using solid-state memory that implement storage policies. EX1003, ¶131. A POSITA implementing the data storage system of Olster would naturally look at other data storage systems that apply storage policies such as Davis to improve upon that system. EX1005, [0030], [0031]; EX1008, 3:32-34; EX1003, ¶131.

Second, a POSITA implementing the replication component of Olster's policy would have looked at other more detailed policies, like those described in Davis. EX1003, ¶132. A POSITA would look at other systems with detailed replication and placement components like Davis to implement Olster's policies. EX1005, [0046]; EX1003, ¶132.

Third, a POSITA would have been motivated to implement the data placement policies of Davis to leverage and optimize the different types of storage media and storage policies that are taught by Olster. EX1003, ¶133. Specifically, Davis teaches that “[t]rade-offs in cost per bit, bit density per unit of area in an integrated circuit, error rates, long-term storage reliability, read disturb rates, read wear rates and write wear rates, among other considerations, affect the choice of which type of flash memory to use in a given application.” EX1008, 1:33-38. The

policies of Davis provide one known approach to increase reliability and prioritize data. EX1003, ¶133.

B. Reasonable Expectation of Success

A POSITA would have been able to modify the replication and placement component of Olster’s policy to include the data placement policies of Davis with a reasonable expectation of success. EX1003, ¶¶134-138.

First, the data storage systems of Olster and Davis have similar structures and components. EX1005, [0030]; EX1008, 3:32-34; EX1003, ¶135. Both teach a controller that can be directed using a software application to execute memory access commands based on a storage policy. EX1005, [0020]; EX1008, 20:12-20; EX1003, ¶135. Further, both store and retrieve data from storage media that includes flash memory. EX1005, [0020]; EX1008, Abstract, 1:29-30; EX1003, ¶135.

Second, both Olster and Davis utilize storage policies directed towards data placement. EX1005, [0046]; EX1008, 1:33-38. If anything, the Davis’s policies expand upon Olster’s replication instructions, providing necessary details that are assumed or left to design choice by Olster. EX1005, [0031], [0046]; EX1008, 1:48-51, 9:37-11:15; EX1003, ¶¶136-137.

Third, given the similar structures and components in Olster and Davis, the modification would simply be a change of programming. EX1003, ¶138. Indeed, Olster teaches that these policies can be “distributed from a client 102,” “system

defined or early set by a system administrator,” effectively explaining to a POSITA how to modify the policies of Olster. EX1005, [0030], [0046]; EX1003, ¶138.

C. Claims 1, 7, 9, 14-15

As discussed in Ground 1, Olster discloses each limitation of claims 1, 7, 9, and 14-15. *See* Section VIII. To the extent that Olster does not expressly disclose a *recording method for writing content to the storage media*, Olster-Davis renders obvious this limitation. EX1003, ¶¶139-140. Specifically, a POSITA would have been motivated to modify the replication component of Olster’s policy to include the policies taught by Davis. *Id.*

The specification of the ’023 Patent only uses the term “recording method” once, and only related to recording density. EX1001, 6:4-8; *see also id.*, 5:64-6:3. Accordingly, the ’023 Patent contemplates that the *recording method* includes storing data based on at least recording frequency and priority level. EX1003, ¶141. This is confirmed by the dependent claims reciting that the recording method can comprise “one or more of a recording path density or a recording frequency” (claim 5) or “instructions to determine a priority level for the content wherein the priority level indicates how densely the content is stored” (claim 6).

Davis discloses *recording method for writing content to the storage media* because it discloses the recording method embodiments disclosed in the ’023 Patent—namely both policy comprising a recording density and instructions to

determine a priority level for the content indicating how densely the content is stored. EX1008, 3:1-4.

As to recording frequency, Davis discloses writing data based on types of flash memory, which “includes at least two types of flash memory having differing numbers of bits per cell.” EX1008, 1:48-53; EX1003, ¶¶143-144. Davis explains that “[t]he single level cell type of flash memory has one bit per cell...and the multilevel cell flash memory types have more than one bit per cell.” *Id.*, 1:12-17. “[T]he single level cell type of flash memory has the lowest density, the highest cost per bit, the longest data retention time, the fastest average write time, the lowest read error rate, the lowest susceptibility to read disturb and program disturb (write disturb) events and errors, the lowest read wear rate and the lowest write wear rate.” *Id.*, 3:8-13. In contrast, “[h]igher numbers of bits per cell reverse these trends.” *Id.*, 3:13-14.

Accordingly, like '023 Patent, Davis discloses storage policies that indicate a method for writing data to a specific type of flash memory based on its recording density (*a recording method for writing content to the storage media*). EX1008, 9:31-36 (“data placement policies 326 applying to types of flash memory with differing numbers of levels per cell may be interpreted as or have corresponding data placement policies applying to types of flash memory with differing characteristics relative to data, and vice versa”); EX1003, ¶144.

As to priority level, Davis discloses “[o]ne policy is writing text data to a type of flash memory having a lower read error rate and writing image data to a type of flash memory having a higher read error rate.” EX1008, 9:37-39. The text data, which is more sensitive to read errors than image data, is given priority. *Id.*; EX1003, ¶145. The text data with higher priority is stored less densely in the memory with lower bits per cell. EX1008, 9:39-43.

Accordingly, like the ’023 Patent, Davis discloses storage policies that indicate a method for writing data that determines a priority level that indicates how densely the data is stored. *Id.* (“This policy could be interpreted as...a policy of writing text data to a type of flash memory having a lower number of bits per cell and writing image data to a type of flash memory having a higher number of bits per cell.”); EX1003, ¶¶145-146.

D. Dependent Claim 2

Olster-Davis renders obvious this limitation. EX1003, ¶¶147-150.

Davis discloses *at least one storage device policy further comprises instructions that direct the device controller to differentiate the content by one or more content types*, which policies are implemented by a controller executing instructions to identify data properties. EX1008, Abstract (“The method includes determining at least one property of a data and determining to which type of a plurality of types of flash memory to write the data, based on the at least one property

of the data.”), 20:17-20; EX1003, ¶149. For example, Davis teaches differentiating between text data and image data. EX1008, 9:37-45.

Davis further discloses *at least one storage device policy further comprises instructions...to assign a customized recording method to each of the one or more content types* because Davis discloses writing data to a type of memory based on data properties. EX1008, 9:39-43 (“This policy could be interpreted as...a policy of writing text data to a type of flash memory having a lower number of bits per cell and writing image data to a type of flash memory having a higher number of bits per cell.”); EX1003, ¶150.

E. Claim 32

1. 32[pre]

To the extent limiting, Olster discloses the preamble as discussed for 1[pre]/9[pre] in Ground 1. EX1003, ¶¶77-78, 269; *see* Section VIII.A.1.

2. 32[a]

Olster discloses this limitation. EX1003, ¶¶79-81, 270-271. Olster describes that storage 150 (*apparatus*) includes storage media 160 (*a storage media*). *See* Section VIII.A.2.

3. 32[b]

Olster discloses this limitation. EX1003, ¶¶82-85, 272-273. Olster explains that storage 150 (*apparatus*) also includes a controller for a node, which may

communicate with and allow for communications with (*operatively coupled with*) storage media 160 (*a storage media*). *See* Section VIII.A.3.

4. 32[b.i]

Olster-Davis renders obvious this limitation. EX1003, ¶¶274-277. Davis teaches differentiating between text data and image data by assigning a method of writing to lower density type memory to text data and assigning a method of writing to higher density type memory to image data. Ex. 1008, 9:37-45; *see* Section IX.D.1; EX1003, ¶¶147-150, 277. Therefore, the data properties (the content type) determine the method of writing, such as the density type of the memory. *See* Section IX.D.1; EX1003, ¶¶147-150, 277.

5. 32[b.ii]

Olster-Davis renders obvious this limitation. EX1003, ¶¶278-279. Davis teaches identifying text data as a first content type and image data as a second content type. Ex. 1008, 9:37-45; *see* Section IX.D.1; EX1003, ¶¶147-150, 278-279.

6. 32[b.iii]

Olster-Davis renders obvious this limitation. EX1003, ¶¶280-282. For example, Davis teaches assigning a method of writing to lower density type memory (*first recording method*) to text data (*the first content*). Ex. 1008, 9:37-45; *see* Section IX.D.1; EX1003, ¶¶147-150, 281-282.

7. 32[b.iv]

Olster-Davis renders obvious this limitation. EX1003, ¶¶283-285.

Davis teaches assigning a method of writing to higher density type memory (*second recording method*) to image data (*the second content*). Ex. 1008, 9:37-45; *see* Section IX.D.1; EX1003, ¶¶147-150, 285. Davis further teaches that the two recording methods differ – one is writing to higher density type memory, and the other is writing to lower density type memory. *Id.*; EX1003, ¶285.

F. Dependent Claims 34, 36

1. Claim 34

Olster-Davis renders obvious this limitation. EX1003, ¶¶286-287. Olster discloses that the “[s]torage media include magnetic media such as...flash memory.” EX1005, [0029]. Davis teaches that “the flash memory 132 includes a single level cell type of flash memory 210, [and] a multilevel cell type of flash memory 212.” EX1008, 4:39-40. In the combination, the text data (*first content*) and the image data (*second content*) would have been stored in the SLC memory 210 and MLC memory 212, respectively, in the Olster’s storage media 160 (*both stored on the storage media*).

2. Claim 36

Olster-Davis renders obvious this limitation. EX1003, ¶¶288-289. As discussed for claim 32, the text data (*first content*) and image data (*second content*) are different types of data with different characteristics (*content type*). EX1008, 9:39-43; EX1003, ¶¶278-279.

X. GROUND 3

Hashimoto discloses each limitation of claims 1, 9, and 14 as features in a single embodiment. EX1003, ¶¶151-159. Accordingly, no motivation to combine or reasonable expectation of success is required. *Realtime Data*, 912 F.3d at 1372-73; *Guardant Health*, No. 2024-1129, 2026 WL 184334 at *4. Regardless, a POSITA would have been motivated to combine the disclosed features in Hashimoto in a single system with a reasonable expectation of success. EX1003, ¶¶157-158.

A POSITA would have been motivated to implement Hashimoto's data storage system with in-drive tier control because it would have maximized performance and endurance and minimized cost. EX1003, ¶158. Hot data requiring higher read and write speed can be stored using first tier writing methods whereas large amounts of cold data can be stored at a lower cost using second tier writing methods. EX1006, [0199]. Similarly, data that may be more sensitive to errors (like text data) can be stored using more reliable first tier writing methods whereas less sensitive data (like image data). *Id.*, [0201]. This customization provides optimization of the entire system. EX1003, ¶158.

A POSITA would have had a reasonable expectation of success as Hashimoto itself teaches that in-drive tiering control is an option for its data storage system. EX1003, ¶159; EX1006, [0451]-[0477], FIGS. 40-41. Indeed, Hashimoto teaches different ways the host-supplied settings could be implemented to include the in-

drive tiering control. *Id.*, [0478]-[0506]. Such a system would have involved a combination of known technologies (data storage systems) according to known methods (communications) to yield the predictable result of a data storage system that implements host-supplied settings. EX1003, ¶159.

A. Independent Claims 1, 9

1. 1[pre]/9[pre]

To the extent limiting, Hashimoto discloses the preamble. EX1003, ¶¶160-162. Hashimoto discloses a storage drive 2 (*apparatus*), such as a solid state drive (SSD). EX1006, [0106] (“FIG. 4 shows a configuration example of the storage device 2.”), [0107] (“In the present embodiment, the storage device 2 is the SSD including the NAND flash memory (hereinafter called flash memory).”), FIG. 4; EX1003, ¶161.

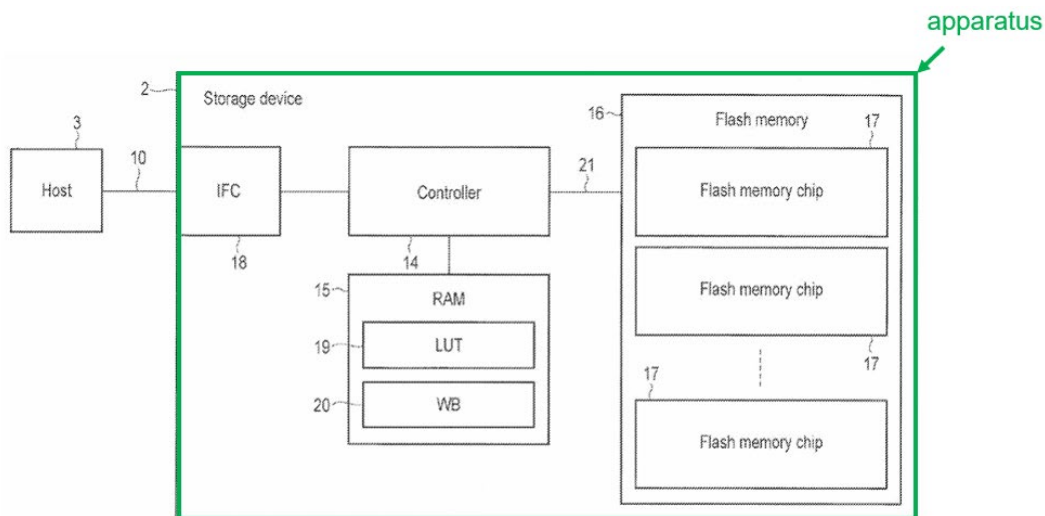


FIG. 4

EX1006, Fig. 4 (annotated)

As discussed below, storage device 2 (*apparatus*) comprises all the elements recited in the claims 1 and 9. See Sections X.A.2-11; EX1003, ¶¶164-211.

2. 1[a]/9[a]

Hashimoto discloses this limitation. EX1003, ¶¶164-166.

Hashimoto discloses storage device 2 (*apparatus*) includes flash memory 16 (*storage media*) to store data. EX1006, [0109] (“The flash memory 16 functions as a nonvolatile memory.”), [0195] (“The controller 14 of the storage device 2 writes the data to the flash memory 16 by employing the writing method designated by the tier attribute.”), FIG. 4.

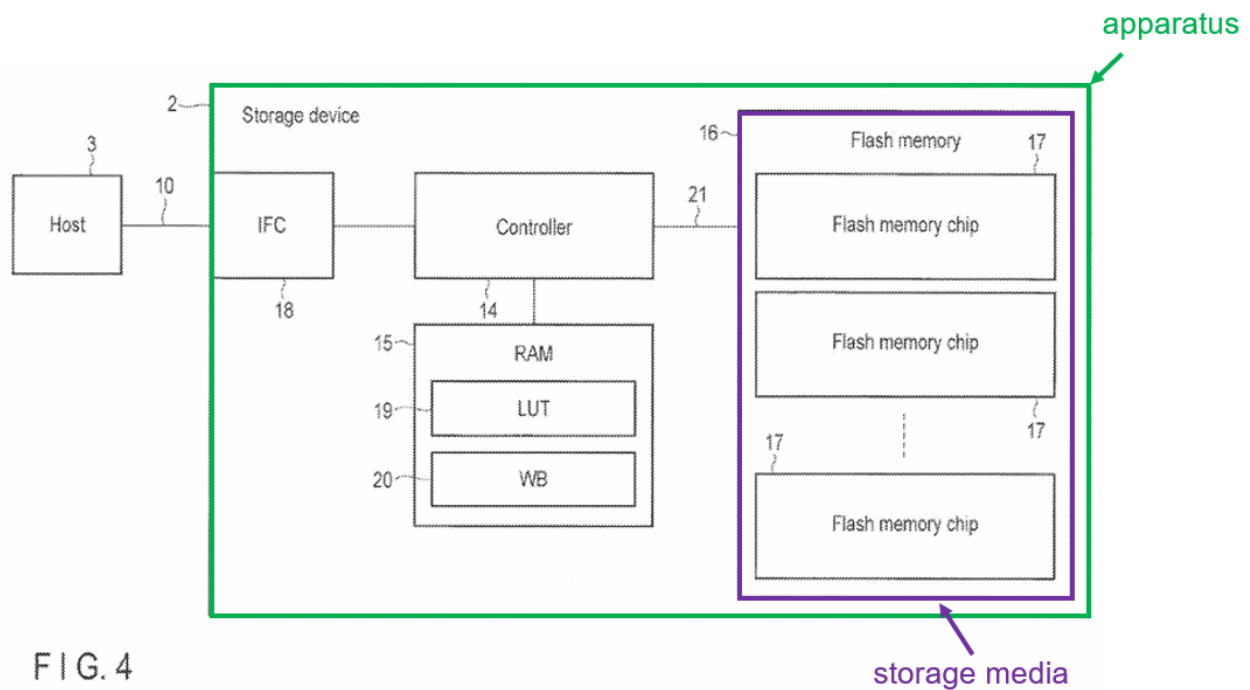


FIG. 4

EX1006, Fig. 4 (annotated)

3. 1[b]/9[b]

Hashimoto discloses this limitation. EX1003, ¶¶167-171.

Hashimoto discloses that storage device 2 (*apparatus*) includes controller 14 (*device controller*). EX1006, [0035], FIG. 4. The controller 14 is *operatively coupled* with the flash memory 16 (*storage media*) because “[t]he controller 14 is electrically connected to the flash memory 16 via a flash memory interface 21.” *Id.*, [0114].

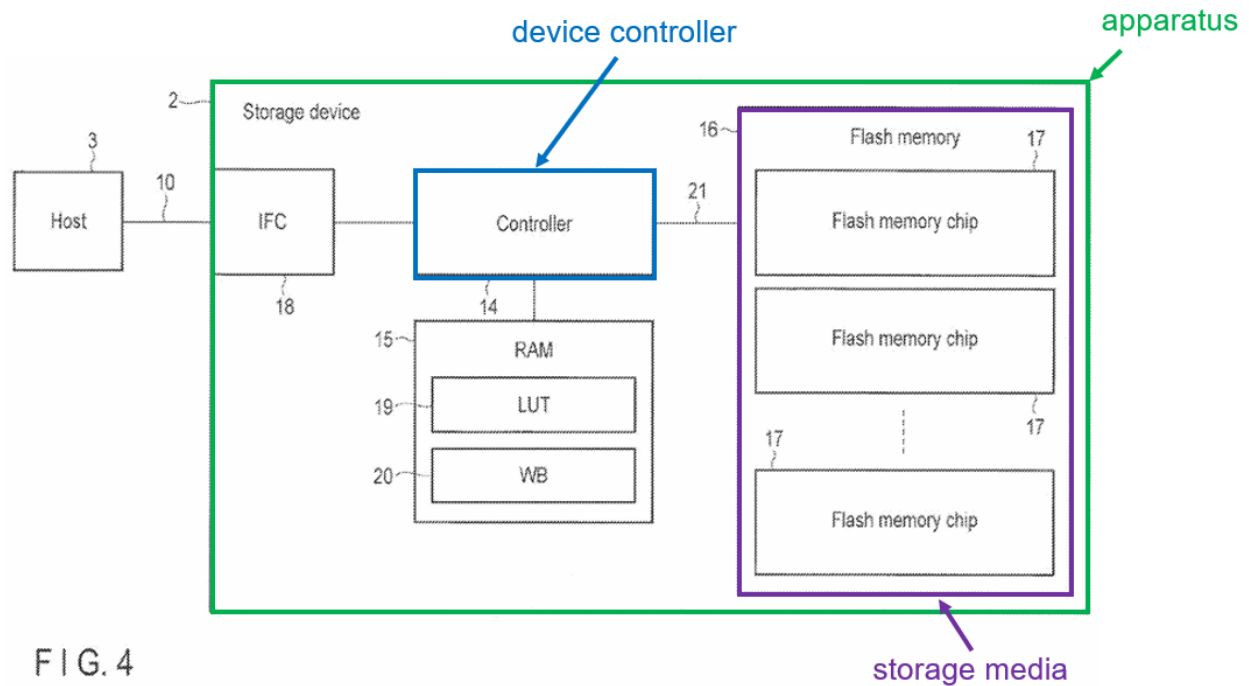


FIG. 4

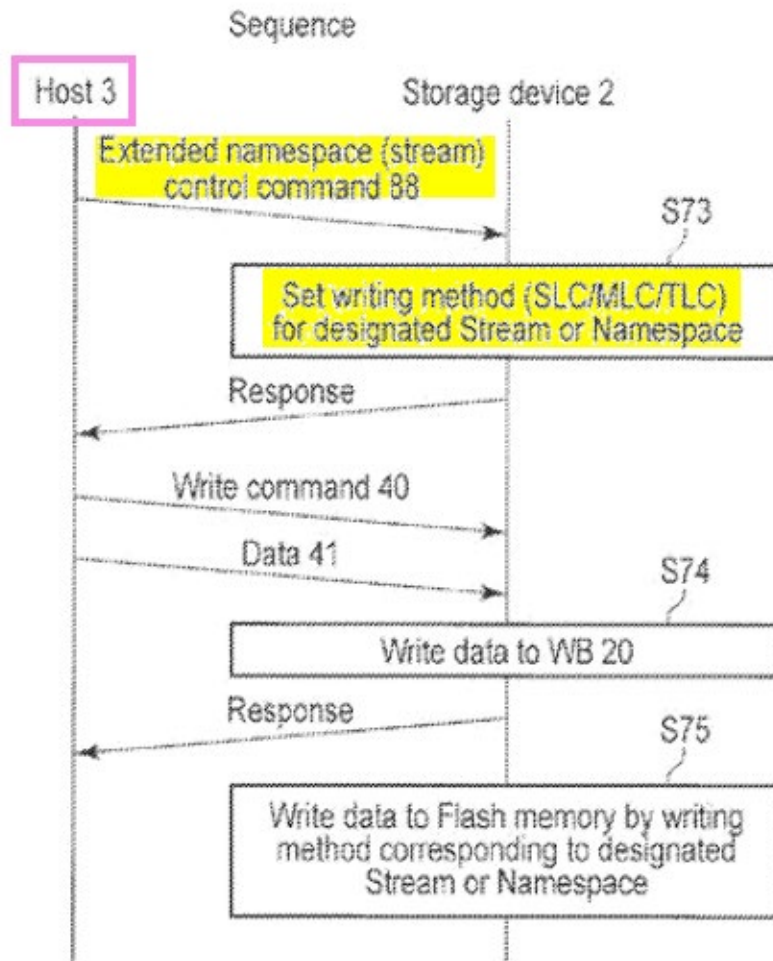
EX1006, Fig. 4 (annotated)

As discussed below, the controller 14 is *configured to at least* receive host-supplied settings (*receive, from a host, at least one storage device policy*) and store and retrieve the data (*content*) from the flash memory 16 (*storage media*) according to the host-supplied settings as recited in 1[b.i]-1[b.ix] and 9[b.1]-9[b.viii].

4. 1[b.i]/9[b.i]

Hashimoto discloses this limitation. EX1003, ¶¶172-184.

Hashimoto discloses that controller 14 (*device controller*) receives from **host 3** commands (such as command 88) to set writing and reading methods for certain data (*receive, from a host, at least one storage device policy*). EX1006, [0491] (“The host 3 transmits the extended namespace (stream) control command 88 to the storage device 2 [and] [t]he controller 14 of the storage device 2 sets the writing method (SLC/MLC/TLC) for the designated target stream or target namespace (step S73).”), FIG. 42 (“[s]et writing method (SLC/MLC/TLC) for designated Stream or Namespace”). Specifically, Hashimoto discloses that the host provides parameters that control how the SSD write content to the flash memory, such as identifying the writing method (SLC/MLC/QLC) for a stream or namespace that the controller later applies whenever writing data associated with the stream or namespace. *Id.*; EX1003, ¶¶173-174.



EX1006, Fig. 42 (excerpted and annotated)

Hashimoto further discloses that these host-supplied settings (*at least one storage policy*) can designate (*indicates*) a tier attribute to write the data to the flash memory 16 (*at least a recording method for writing data to the storage media*). EX1006, [0193] (“The host 3 can designate a tier attribute which should be used to write the data, by using a command for In-drive tiering control (the extended write command or the extended namespace (stream) control command).”), [0195] (“The controller 14 of the storage device 2 writes the data to the flash memory 16 by

employing the writing method designated by the tier attribute.”); EX1003, ¶¶174-175.

As discussed in Ground 2, the specification of the ’023 Patent only uses the term “recording method” once to describe recording methods that write data based on density in one embodiment and based on priority level in another embodiment. *See* Section IX.C; EX1003, ¶¶139-146, 176. Hashimoto’s host-supplied settings (*at least one storage policy*) indicates the tier attribute (*at least a recording method for writing data to the storage media*) teaches both these ’023 Patent embodiments of a *recording method for writing data to the storage media*. EX1006, [0193], [0195], [0614]-[0619]; EX1003, ¶176.

For density, Hashimoto discloses that the tier attribute designates different write methods, where one tier uses less dense write methods (such as SLC write methods) and another tier uses more dense write methods (such as MLC write methods). EX1003, ¶177. As Hashimoto explains, “[i]n the first tier, a 1-bit/cell writing method (SLC writing method) of writing 1-bit data to each memory cell is employed.” EX1006, [0191]. “In the second tier, a writing method of writing data of two or more bits to each memory cell is employed.” *Id.*; *see also id.* (“The writing method employed in the second tier may be a 2-bit/cell writing method (MLC writing method), a 3-bit/cell writing method (TLC writing method), or a 4-bit/cell writing method (QLC writing method).”). Thus, the tier attribute determines which

writing method (such as SLC, MLC, TLC, or QLC writing method) is used to write data to the flash memory 16. EX1003, ¶177.

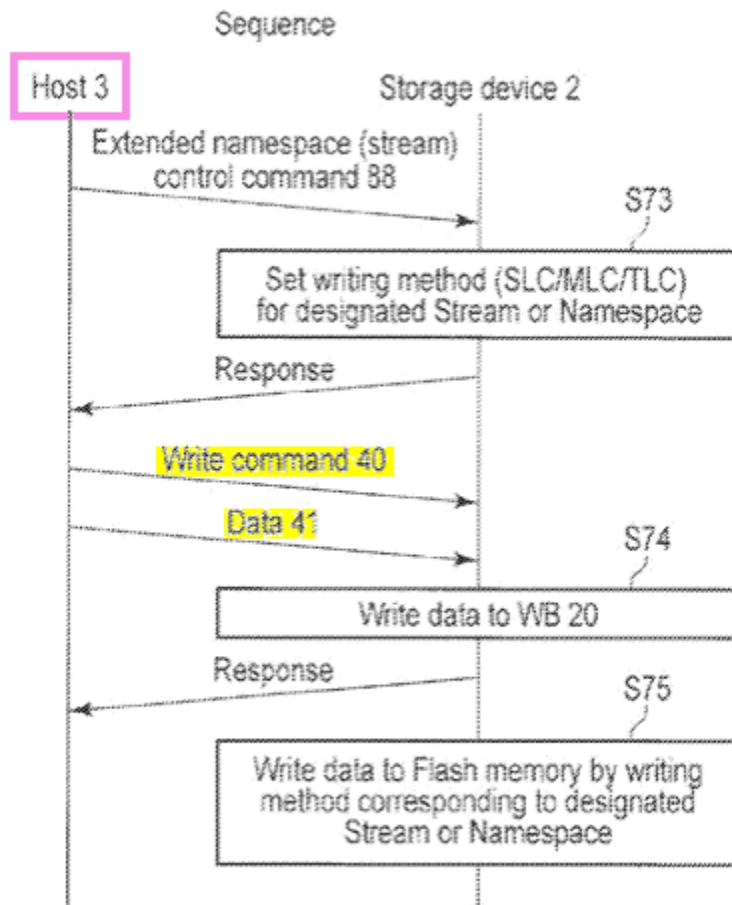
For priority level, Hashimoto discloses that the tier attribute can also determine a priority level for the data to indicate how densely the data should be stored. EX1006, [0198] (“The host 3 can designate a tier attribute giving a higher priority to the write speed than to data retention or a tier attribute giving a higher priority to the data retention than to the write speed, by using a command for the NAND program control (Extended write command, etc.).”). For example, Hashimoto teaches that tier attribute determines priority based on whether the data is hot or cold. EX1006, [0199] (“[W]hen data (hot)...the host 3 may designate the tier attribute giving a higher priority to the write speed than to the data retention. In contrast, when data (cold)...the host 3 may designate the tier attribute giving a higher priority to the data retention than to the write speed.”), EX1003, ¶¶183-184. As another example, “when data is required to have high reliability, the host 3 may designate a tier attribute giving a higher priority to the data reliability than to the read speed.” EX1006, [0202].

5. 1[b.ii]

Hashimoto discloses this limitation. EX1003, ¶¶185-188.

Hashimoto discloses that controller 14 (*device controller*) receives write command 40 (*a storage request*), from **host 3** (*from the host*) to store data 41 (*to*

store content). EX1006, [0492], FIG. 42. The controller 14 is the part of storage device 2 that receives write command 40 because it executes the command. *Id.* (“[T]he controller 14 writes the write data to the block in the flash memory 16 by the writing method for the designated target stream or target namespace (step S75).”); see Section X.A.3. EX1003, ¶¶167-171, 186-188.



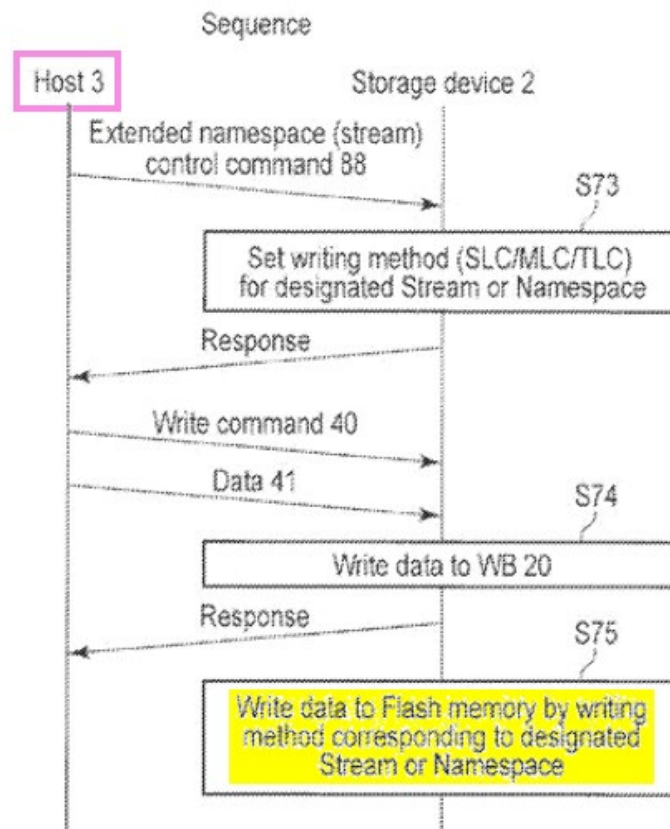
EX1006, Fig. 42 (excerpted and annotated)

6. 1[b.iii]/9[b.ii]

Hashimoto discloses this limitation. EX1003, ¶¶189-191.

Hashimoto discloses that controller 14 (*device controller*) writes the data

(store the content) on flash memory 16 (on the storage media) based on the writing method indicated by the tier attribute (by utilizing the recording method) designated in the host-supplied settings (indicated by the at least one storage device policy). EX1003, ¶190. Specifically, Hashimoto discloses “controller 14 writes the write data to the block in the flash memory 16 by the writing method for the designated target stream or target namespace (step S75).” EX1006, [0492]. As discussed for 1[b.i]/9[b.i], the writing method for the designated target stream is determined by the tier attribute set in step S73. See Section X.A.4; EX1006, [0492], FIG. 42; EX1003, ¶¶172-184, 191.



EX1006, Fig. 42 (excerpted and annotated)

7. 1[b.iv]/9[b.iii]

Hashimoto discloses this limitation. EX1003, ¶¶192-197.

Hashimoto discloses that the “RAM 15” is a *memory of* “controller 14” (*device controller*). For example, CPU 61 of controller 14 exclusively controls **RAM 15**. EX1006, [0150] (“[t]he controller 14 includes a CPU 61, a RAM interface 62”), [0152] (“The RAM interface 62 is a RAM controller configured to control the RAM 15 under control of the CPU 61.”), FIGS. 4, 8. “The RAM 15 includes a storage region to store a lookup table (LUT) 19.” *Id.*, [0109].

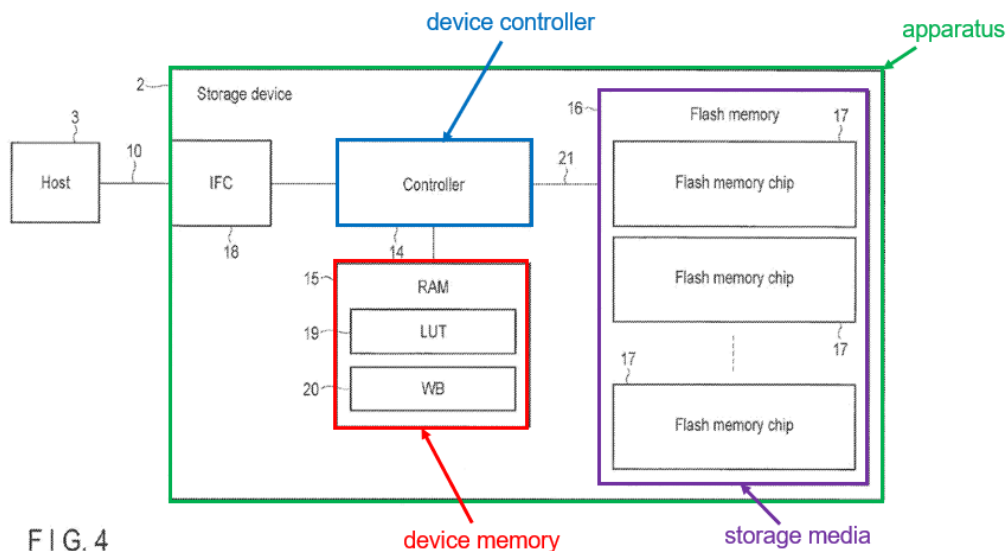


FIG. 4

EX1006, Fig. 4 (annotated)

Hashimoto discloses that controller 14 (*device controller*) stores (*record*) mapping and processing information associated with data 41 (*storage information for the content*) in LUT 19 in RAM 15 of the device controller (*a memory of the device controller*). EX1003, ¶193. Hashimoto teaches “mapping information

indicating relationship between logical addresses and physical addresses of the flash memory 16.” *Id.*, [0110]. This logical-to-physical (L2P) address mapping (part of the *storage information*) is *for* write data 41 (*content*) because the controller updates the LUT 19 to map logical addresses (LBAs) associated with the data to physical addresses after storing that data in flash memory 16, including in the application of the in-drive tiering. *Id.*, [0121], [0254], [0492]; EX1003, ¶194.

Hashimoto also discloses that processing information about how data 41 is stored (part of *storage information for the content*) is included in LUT 19. EX1006, FIG. 4; EX1003, ¶195. For example, as part of the tier attribute, “host 3 can designate either a tier attribute of compressing data and writing the compressed data to the block of the flash memory 16 or a tier attribute of writing the write data to the block without compressing the data.” EX1006, [0222]. The associated processing information, such as compression, is also stored in LUT 19. EX1006, [0662] (“The controller 14 updates the lookup table 19, maps the LBA of the write data to the physical address of the physical storage location at which the write data has been written, and further associates a compression flag to the LBA of the write data”), [0663]-[0665], FIG. 61; EX1003, ¶195.

LUT 19

Logical addresses (LBA)	Physical addresses	Compression flags	Others

FIG. 61

EX1006, Fig. 61 (annotated)

Hashimoto discloses that the L2P address mapping stored on RAM 15 *compris[es] at least* the LBA (*a content identifier*) for the data (*for the content*). EX1003, ¶196. Hashimoto discloses that the LUT 19 of RAM 15 records LBAs as part of its L2P address mapping. *Id.*, [0416], [0434], [0662] (“lookup table 19, maps the LBA of the write data to the physical address of the physical storage location at which the write data has been written”). An LBA uniquely identifies data because it is used by the host to subsequently retrieve that data from the flash memory, and is *thus a content identifier for the content*. *Id.*, [0104], [0121], [0416], [0417], [0434], [0437]; *see* Section X.A.8; EX1003, ¶197.

8. 1[b.v]/9[b.iv]

Hashimoto discloses this limitation. EX1003, ¶¶198-201. Hashimoto discloses that after the data 41 is written to flash memory 16 (*subsequent to storing the content*), the controller 14 (*device controller*) can then *receive* a read command (*content request*) for data 41 (*content*). EX1006, [0663] (“controller 14 receives the read command from the host 3”); *see also id.* [0082], [0098], cl. 17 (“a host configured to send commands to write data to or read data from the storage device”).

The read command for data 41 (*content request for the content*) includes the LBA (*content identifier*) for the data 41: “The command such as a write command and a read command includes a starting logical address (Starting LBA: Logical block Address) corresponding to the data (data to be written or data to be read) and a sector count (transfer length).” EX1006, [0104]; EX1003, ¶200.

Hashimoto also teaches that the read command is received *subsequent to storing the content* because the controller uses the same LBA for both reads and writes (*id.*) and which means that the controller can only satisfy the request by consulting the L2P address mapping established for that data at write time. EX1003, ¶201. For example, “[w]hen the controller 14 receives the read command from the host 3, the controller 14 determines whether or not the data requested to be read is compressed, by referring to the lookup table 19,” and if so, “de-compresses the data read from the flash memory 16.” *Id.*, [0663]. This determination is only possible

because the L2P address mapping of the data requested by the read command (*content request for the content*) was previously recorded in the LUT 19 in the context of a write command (storage request). EX1003, ¶201.

9. 1[b.vi]/9[b.v]

Hashimoto discloses this limitation. EX1003, ¶¶202-205. Hashimoto teaches that *in response to the read command (content request)*, controller 14 *retrieves from* LUT 19 in RAM 15 of the controller 14 (*memory of the device controller*), the L2P mapping and processing information (*storage information for the content*). EX1003, ¶204. As Hashimoto explains, “[w]hen the controller 14 receives the read command” (*content request*) “from the host 3, the controller 14” (*device controller*) “determines whether or not the data requested to be read is compressed, by referring to the lookup table 19” of RAM 15 (*storage information for the content*). EX1006, [0663]. The controller 14 *retrieves the* L2P address mapping and processing information (*storage information*) *from* LUT 19 of RAM 15 (*the memory of the device controller*) because the L2P mapping and processing information was previously stored there and “referring to [the L2P mapping in] the lookup table 19” and other processing information necessitates retrieving it first. *Id.*, [0663]; *see* Section X.A.7. The controller 14 will need the L2P mapping to identify where the data is stored, and the processing information (such as whether the data is compressed to determine whether the data needs to be decompressed and the ECC

applied) to recover the data. EX1003, ¶205.

10. 1[b.vii]/9[b.vi]

Hashimoto discloses this limitation. EX1003, ¶¶206-212. Hashimoto discloses that controller 14 (*device controller*) retrieves data 41 (*recover the content*) *from the* flash memory 16 (*storage media*) *at least in part according to the* L2P mapping and processing information, such as whether the data is compressed (*at least in part according to the storage information*). EX1003, ¶208. Controller 14 can only read the requested data “from the flash memory 16” because the physical address of that data was previously retrieved from the L2P address mapping (*storage information*) stored in LUT 19. EX1006, [0110], [0663]; EX1003, ¶208. Also, after “referring to the lookup table 19[,] [i]f the data requested to be read is compressed, the [controller] 14 de-compresses the data read from the flash memory 16 and transmits the de-compressed data to the host 3.” EX1006, [0663]; EX1003, ¶209. Thus, Hashimoto discloses *recovering* the data (*content*) from flash memory 16 (*the storage media*), *at least in part according to the* L2P mapping and processing information (*storage information*). EX1003, ¶209.

For claim 9, Hashimoto discloses that controller 14 (*device controller*) also retrieves data 41 (*recover the content*) *from the* flash memory 16 (*storage media*) *according to the* host-supplied settings including the tier attribute (*at least one storage device policy*). See Section X.A.4; EX1003, ¶210. As discussed for

1[b.1]/9[b.i], the tier attribute determines the writing method (SLC/MLC/TLC) used to write the data. *See* Section X.A.4. To retrieve the data, Hashimoto explains that controller 14 must also determine the writing method used to read the data from memory. EX1006, [0134]-[0135]; EX1003, ¶¶210-211. “When a write method (SLC: Single Level Cell) for writing 1-bit data per memory cell is employed, the physical sector 35 is equivalent to a page.” EX1006, [0134]. In comparison, “[w]hen a write method (MLC: Multi Level Cell) for writing 2-bit data per memory cell is applied, the physical sector 35 is equivalent to two pages (i.e., an upper page and a lower page).” *Id.* Hashimoto explains that “[o]ne word line and one physical sector are selected based on the physical address like the row address received from the controller 14, during the read operation.” *Id.*, [0135]. To determine whether the address includes one page or two pages, controller 14 uses the host-supplied settings including the tier attribute (*at least one storage device policy*) to determine which writing method was used. EX1003, ¶211.

Hashimoto further discloses controller 14 (*device controller*) retrieves data 41 (*recover the content*) from the flash memory 16 (*storage media*) according to the host-supplied settings including the tier attribute (*at least one storage device policy*) because the tier attribute is used to determine the error correction code (ECC) that controller 14 applies to detect errors in the data before returning the data to the host. EX1003, ¶212. Hashimoto explains that the ECC is determined “in accordance with

the designated tier attribute.” EX1006, [0521]. For example, when controller 14 selects a “writing method giving a higher priority to the data retention than to the write speed” according to the tier attribute, a “a strong ECC is applied to assure the data reliability.” EX1006, [0522]-[0523]. In response to a read command, controller 14 applies the ECC decoding based on host-supplied settings including the tier attribute. EX1003, ¶212.

11. 1[b.viii]/9[b.vii]

Hashimoto discloses this limitation. EX1003, ¶¶213-214. Hashimoto’s controller 14 (*device controller*) responds to the read command (*respond to the content request*) by *at least returning the ECC-decoded data (content) to host 3 (the host)*. EX1006, [0663] (controller 14 transmits requested data to the host 3); *see also id.*, [0445], [0449]; EX1003, ¶214.

B. Dependent Claim 14:

1. Claim 14

Hashimoto discloses this limitation. EX1003, ¶¶215-217.

Hashimoto discloses flash memory 16 (*storage media*) *comprises a solid-state memory device* in the form of flash memory chip 17. EX1006, [0109], [0084], [0107]. Solid-state memory, also referred to as nonvolatile memory, is a type of memory (such as NAND flash memory) that stores data persistently without moving parts. EX1003, ¶¶216-217.

XI. GROUND 4

A. Motivation to Combine

A POSITA would have been motivated to modify Hashimoto's host-supplied settings to include Kilian's retention policy feature. EX1003, ¶¶218-225.

First, a POSITA implementing Hashimoto would naturally look at other data storage references that apply user-configurable settings to improve the system. EX1003, ¶¶219-221. A POSITA would have recognized that Hashimoto and Kilian are directed to the same basic architecture: a controller within a storage system that receives commands and instructions from a host and, based on those instructions, manages how content is stored, protected, and retrieved on behalf of the host. EX1006, [0082]-[0084], [0096]-[0097], [0107]-[0110]; EX1007, 3:16-22, 3:40-63; EX1003, ¶221.

Second, Kilian identifies benefits of its host-supplied retention feature, explaining that conventional protections are inadequate because "an administrator of the system typically has the ability to change the 'read-only' status of files such that the files may be modified or deleted," and "conventional file systems do not allow the user to specify a period of time for which data cannot be deleted or modified." *Id.*, 1:38-42. Kilian's retention period addresses this issue: "once the retention period has been established for a unit of data, the storage system ensures that the unit of data is retained in an unmodified state for the full retention period."

Id., 3:36-39. A POSITA would have been motivated to add Kilian’s host-specified data retention/write-protection functionality to Hashimoto’s host-programmable controller to obtain these expressly-taught benefits. EX1003, ¶¶219-221.

Third, Kilian’s retention period policy would have made Hashimoto’s system more reliable. EX1003, ¶222. Hashimoto discloses an SSD that already accepts host-supplied settings controlling how data is stored and protected. EX1006, [0189]-[0195]. Notably, Hashimoto provides hosts with the ability to implement settings prioritizing data reliability. *Id.*, [0521]-[0524]. Killian supplies an additional, known host-supplied protection setting that improves reliability—preventing deletion or modification during a retention period. EX1003, ¶222. Particularly, Kilian’s policy improves reliability of read-only data by preventing corruption or malicious overwrites by an attacker. EX1007, 1:38-42; EX1003, ¶222.

B. Reasonable Expectation of Success

A POSITA would have a reasonable expectation of success to modify Hashimoto’s storage device to incorporate the retention period policy taught by Davis. EX1003, ¶¶223-225.

First, Hashimoto’s controller 14 already receives and executes a variety of host-supplied storage device policies and stores the resulting state (*e.g.*, tier attributes, compression flags) in its own onboard memory for use in subsequent read and write operations. EX1006, ¶¶[0110], [0150]-[0155], [0189]-[0195], [0217]-

[0227]; EX1003, ¶223. Because controller 14 is already configured to receive a host-supplied policy, store it, and consult it when executing subsequent commands targeting specific data, adding a further host-supplied policy directing the controller to deny deletion or modification requests for a designated period (as taught by Kilian, EX1007, 3:23-39) requires nothing more than programming the controller with one additional rule of the same general kind it already implements (*i.e.*, recording a retention date in the LUT 19). EX1003, ¶224.

Second, a POSITA would have had a reasonable expectation of success in combining Kilian's retention period policy with Hashimoto's controller, since the combination uses each reference for its own established purpose and does not require any new hardware, new host-device interface, or unpredictable modification to Hashimoto's existing architecture. *Id.*, ¶225. Instead, Kilian's policy would simply direct Hashimoto's controller 14 to check an additional host-supplied parameter (a retention period stored in the LUT) before executing a delete or overwrite operation, in the same manner that controller 14 already checks other host-supplied parameters (*e.g.*, compression attribute) before executing write and read operations. *Id.*, ¶225.

C. Dependent Claims 7, 15

Hashimoto in view of Kilian renders obvious these limitations. EX1003, ¶¶226-231. Kilian discloses that the storage system, rather than the host, generates the identifier used to reference stored content, and communicates that identifier back

to the host so the host can use it to access the data afterward: “the content address is generated by the storage system itself, and is returned to the host.” *Id.*, 9:41-43; *see also id.*, 4:65-5:11. This content address is generated by the storage system’s controller as part of processing the host’s storage request, and the returned address is what the host subsequently uses “for use in the future to retrieve the data from the storage system.” *Id.*, 4:65-5:1.

A POSITA would have been motivated to modify Hashimoto’s controller 14 (which otherwise relies on the host to supply the Starting LBA identifying the write data (EX1006, [0121])) to instead assign its own content identifier and return it to the host in the write command’s completion response, in order to obtain Kilian’s expressly-taught benefit of a trusted, storage-system-authenticated identifier that a host cannot spoof to circumvent the storage device’s data-management policies. EX1007, 9:46-56 (generating and returning the content address thereby “ensuring that retention policies are met, irrespective of the behavior of hosts connected thereto”); EX1003, ¶229. A POSITA would have had a reasonable expectation of success in the combination because it only requires relocating a known identifier-assignment function from the host to a controller already responsible for maintaining an identifier-to-physical address mapping in the LUT 19 and providing this identifier to the host when confirming command completion. EX1003, ¶229.

D. Independent Claim 16

1. 16[pre]

To the extent limiting, Hashimoto discloses the preamble as discussed for 1[pre]/9[pre] in Ground 3. *See* Section X.A.1; EX1003, ¶232.

2. 16[a]

Hashimoto discloses this limitation. EX1003, ¶¶233-234. Hashimoto discloses storage device 2 (*apparatus*) includes flash memory 16 (*a storage media*) as discussed for 1[a]/9[a] in Ground 3. *See* Section X.A.2; EX1003, ¶234.

3. 16[b]

Hashimoto discloses this limitation. EX1003, ¶¶235-236. Hashimoto discloses storage device 2 (*apparatus*) includes controller 14 electrically connected via a flash memory interface 21 (*operatively coupled with*) flash memory 16 (*a storage media*) as discussed for 1[b]/9[b] in Ground 3. *See* Section X.A.3; EX1003, ¶236.

4. 16[b.i]

Hashimoto in view of Kilian renders obvious this limitation. EX1003, ¶¶237-240. The '023 Patent explain that write-protection determines whether data is deleted. EX1001, 22:29-43 (“the content should be stored as *write-protected (or write-once)* such that i[t] will *not be deleted* later or overwritten”). Kilian states that its host provides “a retention period that indicates a period of time for which the data may not be deleted,” and the controller enforces this policy: when it “receives a

request to delete the data, it first evaluates the retention period...[and i]f the retention period has not expired, the storage system denies the request.” EX1007, Abstract. This host-supplied retention period is a *write-protection policy* because it instructs the controller to protect content from deletion until the period elapses. EX1003, ¶240.

5. 16[b.ii]

Hashimoto discloses this limitation. EX1003, ¶¶241-242. Hashimoto discloses that controller 14 (*device controller*) receives write command 40 (*a storage request*) from host 3 (*from the host*) to store data 41 (*to store content*) as discussed for 1[b.ii] in Ground 3. See Section X.A.5; EX1003, ¶242.

6. 16[b.iii]

Hashimoto discloses this limitation. EX1003, ¶¶243-244. Hashimoto discloses controller 14 (*device controller*) writes the data (*store the content*) on flash memory 16 (*on the storage media*) based on the writing method indicated by the tier attribute designated in the host-supplied settings (*indicated by the at least one storage device policy*) as discussed for 1[b.iii]/9[b.ii] in Ground 3. See Section X.A.6; EX1003, ¶244.

7. 16[b.iv]

Hashimoto-Kilian renders obvious this limitation. EX1003, ¶¶245-248.

In Hashimoto-Kilian, the processing information associated with data 41 would include the retention period policy as taught by Kilian in LUT 19. EX1003, ¶247. Kilian teaches that the data store system receives “a retention period that indicates a period of time for which the data may not be deleted.” EX1007, Abstract. Accordingly, “[w]hen the storage system receives a request to delete the data, it first evaluates the retention period associated with that data to determine if the retention period has expired.” *Id.*

LUT 19

Logical addresses (LBA)	Physical addresses	Compression flags	Others

FIG. 61

EX1006, Fig. 61 (annotated)

A POSITA would have been motivated to include the retention period in LUT 19 because it would centralize the processing information associated with the data located at the LBA. EX1003, ¶248. Hashimoto teaches that LUT 19 includes other processing information about the data located at the LBA, and it would have been a

simple modification of the code to include the retention period as taught by Kilian, providing a POSITA with a reasonable expectation of success. EX1006, Fig. 61; EX1007, Fig. 4; EX1003, ¶248.

8. 16[b.v]

Hashimoto-Kilian renders obvious this limitation. EX1003, ¶¶249-251.

As discussed for 16[b.i], Kilian states that the data storage system “receives a request to delete the data” (*receive a delete request*). EX1007, Abstract; EX1003, ¶251; *see also* Section XI.D.4. In Hashimoto-Kilian, the delete request would be for the previously-stored data 41. EX1003, ¶251.

9. 16[b.vi]

Hashimoto-Kilian renders obvious this limitation. EX1003, ¶¶252-254. In Hashimoto-Kilian, the L2P mapping and processing information, including the retention period, would also be retrieved in response to the delete command (*in response to the delete request*). EX1007, Abstract, 3:31-34; FIG. 4; EX1003, ¶¶253-254.

10. 16[b.vii]

Hashimoto-Kilian renders obvious this limitation. EX1003, ¶¶255-258.

Hashimoto discloses that the controller 14 (*device controller*) will respond to commands (*respond to the...request*). EX1003, ¶256. As Hashimoto shows in Fig.

3, the storage device 2 through the controller 14 will send a reply to the host whenever an instruction or request is received. EX1006, [0102]-[0104], FIG. 3.

In Hashimoto-Kilian, controller 14 responds to the delete command (*respond to the delete request*), indicating that data 41 was not deleted due to the retention period (*the content was not deleted in accordance with the write-protection policy*). EX1003, ¶257. The controller denies the delete request if the period has not expired, responding to the host by indicating that the content was not deleted. EX1007, 10:8-20 (“When it is determined... that the retention period has not yet expired, [the] process denies the request to delete the CDF,” with “[t]he storage system ... optionally return[ing] an error message to the requester (e.g., a host).”); EX1003, ¶257.

E. Dependent Claims 17-20

1. Claim 17

Hashimoto-Kilian renders obvious this limitation. EX1003, ¶¶259-261. Kilian’s delete request is received from the host. EX1007, 1:59-60, 2:6-7 (“in response to a request from the host computer to delete the data”). In the Hashimoto-Kilian system, the storage policy and the delete request are thus received from the same source: the host. EX1003, ¶261.

2. Claim 18

Hashimoto-Kilian renders obvious this limitation. EX1003, ¶¶262-263. Kilian discloses a storage system that “communicates with a plurality of hosts 105a, 105b, ..., 105m through a network 103.” EX1007, 3:42-43. Kilian further teaches that the storage system enforces retention policies “irrespective of the behavior of hosts connected thereto,” confirming that any connected host may issue a delete request, not only the host that established the policy. *Id.*, 9:50-54; EX1003, ¶263.

3. Claim 19

Hashimoto-Kilian renders obvious this limitation. EX1003, ¶¶264-265. Kilian’s retention-period policy encompasses modification requests. Kilian defines the retention period as “a period of time for which the data cannot be deleted or modified.” EX1007, 3:23-29; *see also* 3:31-35 (“receives a request from a host computer to delete or modify the data”); EX1003, ¶265.

4. Claim 20

Hashimoto-Kilian renders obvious this limitation. EX1003, ¶¶266-268. Kilian explains that in conventional systems, storing modified data could “overwrite the original data,” and Kilian’s system prevents this by ensuring that modified data “is stored at a different location from original data 201 and does not overwrite the original data.” EX1007, 5:33-36. Kilian’s write-protection policy thus treats an *overwrite request* (*i.e.*, a request to write new data to the location of previously stored

content) as a form of deletion request that is denied when the retention period has not expired. EX1003, ¶¶267-268.

XII. SECONDARY CONSIDERATIONS

There are no secondary considerations known to Petitioners that affect, let alone overcome—this strong case of obviousness.

XIII. CONCLUSION

Petitioners respectfully request institution.

U.S. Patent No. 10,776,023
Petition for *Inter Partes* Review

Dated: August 5, 2026

Respectfully submitted,

By: /s/ Lisa K. Nguyen
Lisa K. Nguyen (Reg. No. 58,018)
PAUL HASTINGS LLP
1117 S. California Avenue
Palo Alto, CA 94304
Telephone: 650-320-1800
lisanguyen@paulhastings.com

Counsel for Petitioner
META PLATFORMS, INC.

CLAIM LISTING APPENDIX

Claim 1

- 1 [pre] An apparatus comprising:
- 1 [a] a storage media;
- 1 [b] a device controller operatively coupled with the storage media
 configured to at least:
- 1 [b.i] receive, from a host, at least one storage device policy, wherein
 the at least one storage device policy indicates at least a
 recording method for writing content to the storage media;
- 1 [b.ii] receive a storage request, from the host, to store content;
- 1 [b.iii] store the content on the storage media by utilizing the recording
 method indicated by the at least one storage device policy;
- 1 [b.iv] record storage information for the content on a memory of the
 device controller, comprising at least a content identifier for the
 content;
- 1 [b.v] subsequent to storing the content, receive a content request for the
 content, comprising at least the content identifier for the
 content;
- 1 [b.vi] in response to the content request, retrieve from the memory of the
 device controller, the storage information for the content;
- 1 [b.vii] recover the content from the storage media, at least in part
 according to the storage information; and
- 1 [b.viii] respond to the content request at least by returning the content to
 the host.

Claim 2

- 2 The apparatus of claim 1, wherein the at least one storage device policy further comprises instructions that direct the device controller to differentiate the content by one or more content types and to assign a customized recording method to each of the one or more content types.

Claim 7

- 7 The apparatus of claim 1, wherein the content identifier is assigned by the device controller and is returned in a response to the storage request.

Claim 9

- 9 [pre] An apparatus comprising:
- 9 [a] a storage media;
- 9 [b] a device controller operatively coupled with the storage media configured to at least:
- 9 [b.i] receive, from a host, at least one storage device policy, wherein the at least one storage device policy indicates at least a recording method for writing content to the storage media;
- 9 [b.ii] store content on the storage media by utilizing the recording method indicated by the at least one storage device policy;

- 9 [b.iii]** record storage information for the content on a memory of the device controller, comprising at least a content identifier for the content;
- 9 [b.iv]** subsequent to storing the content, receive a content request for the content, comprising at least the content identifier for the content;
- 9 [b.v]** in response to the content request, retrieve from the memory of the device controller the storage information for the content;
- 9 [b.vi]** recover the content from the storage media, according to the storage information and according to the at least one storage device policy; and
- 9 [b.vii]** respond to the content request at least by returning the content to the host.

Claim 14

- 14** The apparatus of claim 9, wherein the storage media comprises a solid-state memory device.

Claim 15

- 15** The apparatus of claim 9, wherein the content identifier is assigned by the device controller and is returned in a response to the storage request.

Claim 16

- 16 [pre]** An apparatus comprising:
- 16 [a]** a storage media;

- 16 [b]** a device controller operatively coupled with the storage media configured to at least:
- 16 [b.i]** receive, from a host, a storage policy, wherein the storage policy comprises at least a write-protection policy and recording method for writing content to the storage media;
 - 16 [b.ii]** receive, from the host, a storage request to store content;
 - 16 [b.iii]** store the content on the storage media by utilizing the recording method of the storage policy;
 - 16 [b.iv]** record storage information for the content on a memory of the device controller, comprising at least a content identifier for the content;
 - 16 [b.v]** subsequent to storing the content, receive a delete request for the content, comprising at least the content identifier for the content;
 - 16 [b.vi]** in response to the content request, retrieve from the memory of the device controller the storage information for the content;
 - 16 [b.vii]** respond to the delete request, indicating that the content was not deleted in accordance with the write-protection policy.

Claim 17

- 17** The apparatus of claim 16, wherein the storage policy and the delete request are received from the same source.

Claim 18

- 18** The apparatus of claim 16, wherein the storage policy and the delete request are received from different sources.

Claim 19

- 19** The apparatus of claim 16, wherein the delete request is a modification request.

Claim 20

- 20** The apparatus of claim 16, wherein the delete request is a overwrite request.

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that foregoing
**PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO.
10,776,023** exclusive of the parts exempted as provided in 37 C.F.R.
§42.24(a), contains 13,998 words and therefore complies with the type-volume
limitations of 37 C.F.R. §42.24(a).

Dated: August 5, 2026

Respectfully submitted,

By: /s/ Lisa K. Nguyen
Lisa K. Nguyen (Reg. No. 58,018)
PAUL HASTINGS LLP
1117 S. California Avenue
Palo Alto, CA 94304
Telephone: 650-320-1800
lisanguyen@paulhastings.com

Counsel for Petitioner
META PLATFORMS, INC.

CERTIFICATE OF SERVICE

I hereby certify that a true and correct copy of the foregoing **Petition for *Inter Partes* Review of U.S. Patent No. 10,776,023 and Exhibits** are being served on August 5, 2026 via Federal Express overnight mail on counsel of record for U.S. Patent No. 10,776,023 as addressed below:

**POLSINELLI PC
PO Box 140310
KANSAS CITY, MO 64114-0310**

A courtesy copy is also being served on counsel for the patent holder in the pending litigation:

Rajeev Gupta, raj.gupta@finnegan.com
Karthik Kumar, karthik.kumar@finnegan.com
Daniel F. Roland, daniel.roland@finnegan.com
Daniel F. Klodowski, daniel.klodowski@finnegan.com
C. Collette Corser, collette.corser@finnegan.com
Victor M. Palace, victor.palace@finnegan.com
Jordan M. Cowger, jordan.cowger@finnegan.com
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, LLP
901 New York Avenue, N.W., Washington, D.C. 20001-4413

Dated: August 5, 2026

Respectfully submitted,

By: /s/ Lisa K. Nguyen
Lisa K. Nguyen (Reg. No. 58,018)
PAUL HASTINGS LLP
1117 S. California Avenue
Palo Alto, CA 94304
Telephone: 650-320-1800
lisanguyen@paulhastings.com

Counsel for Petitioner
META PLATFORMS, INC.

