

Netskope, Inc. v. Fortinet, Inc.

IPR2023-00458

U.S. Patent No. 9,280,678

Patent Owner's Demonstratives

Hearing: June 21, 2024

Ground	Claim(s) challenged
1. Cidon, Shikfa, Herrmann	1-3, 8-14, 16-19, 21-27
2. Cidon, Shikfa, Herrmann, Fellenstein	6
3. Cidon, Shikfa, Herrmann, Hirai	15
4. Auradkar, Chiueh, Chambers	1-3, 8-19, 21-27
5. Auradkar, Chiueh, Chambers, Inoue	6

Summary of Patent Owner's Arguments

Ground	Patent Owner's Arguments	Applies to Claims
All Grounds	Cidon and Auradkar do not disclose “a policy” that performs all recited functions	All Challenged Claims
All Grounds	Cidon and Auradkar do not “create” or “configure” a namespace	All Challenged Claims
1-3	No motivation to combine Cidon and Shikfa	All Challenged Claims
4-5	No motivation to combine Aurakdar and Chiueh	All Challenged Claims
All Grounds	Cidon and Auradkar do not process a file “to be stored/uploaded”	All Challenged Claims
1	Cidon does not disclose storing a file “across two cloud platforms”	11, 25
5	Inoue's keyword is not content from a file	6

Overview of '678 Patent

Overview of the '678 Patent - Claim 1

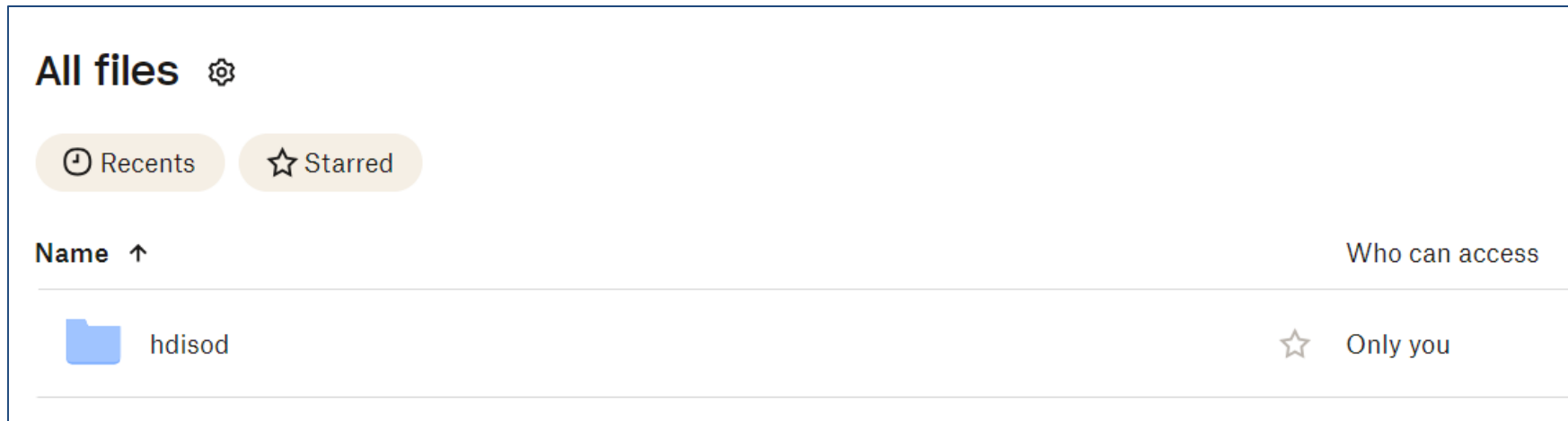
1. A method comprising:
assigning to one or more users, by a gateway device, a policy for managing access to and processing a file to be stored on one or more cloud platforms, wherein the policy defines access rights of the one or more users;
encrypting, by the gateway device, using cryptographic key information defined by the policy, content of the file to produce a **searchable encrypted file** by:
dividing the file into a plurality of chunks;
creating namespaces for one or more of the plurality of chunks; and
configuring the namespaces of the one or more chunks such that content of the file is encrypted in a manner that makes it searchable;
storing, by the gateway device, the searchable encrypted file on the one or more cloud platforms based on the policy; and
managing access to the searchable encrypted file by the one or more users based on the policy.

Overview of the '678 Patent – Storing searchable, encrypted file on cloud service

- File encryption process: Create a folder named with the file's encrypted filename

“In the present example, the top-most portion of each container may represent a folder within which the other portions are included in the form of files. The name of the folder may be the encrypted version of the original file name and the names of the files may include one or more searchable indices.”

Ex. 1001 (678 Patent) at 13:2-7



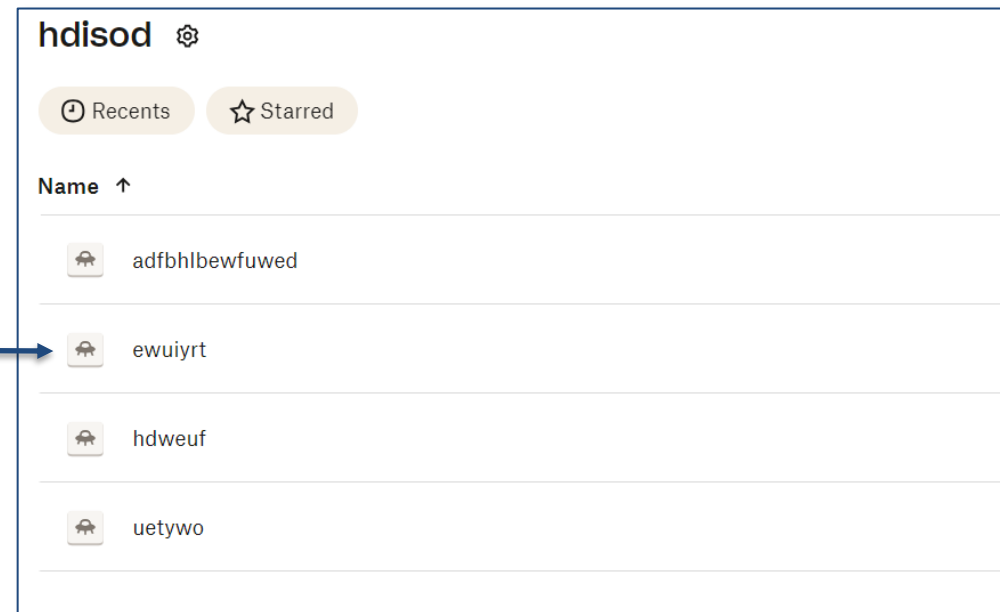
Overview of the '678 Patent – Storing searchable, encrypted file on cloud service

- File encryption process: Split file into chunks, encrypt each chunk, name them with encrypted keywords, and store in folder.

“Each container can include multiple encrypted file chunks, which may be named by means of searchable indices as mentioned in connection with the description of FIGS. 3A-B. For instance, container 1 can include four file chunks having their names (adfbhlbewfuwed, ewuiyrt, hdweuf, and uetywo).”

Ex. 1001 (678 Patent) at 12:67-13:2

“Hello” encrypted to:



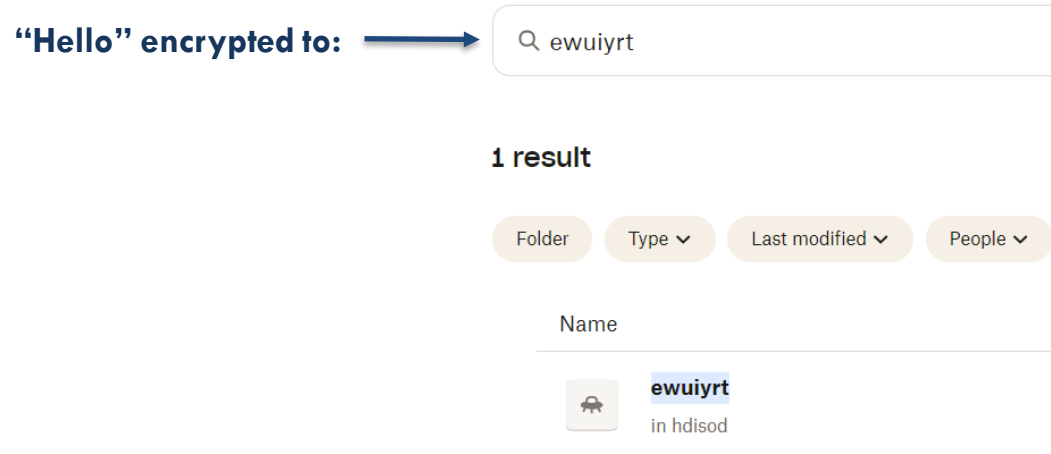
POR 5-6, 19-21; Surreply 3, 7-8

Overview of the '678 Patent – Storing searchable, encrypted file on cloud service

- File can now be searched using cloud service's built-in search
 - Use the encrypted search word as a search
 - Return the encrypted filename and/or download the encrypted file

“Access to such searchable encrypted files can allow the users to search through the content of the files within the context of the cloud service providers 206.”

Ex. 1001 (678 Patent) at 10:17-19



Overview of Prior Art

Overview of Cidon (Grounds 1-3)

- Cidon teaches encrypting a file by
 - Splitting it into segments
 - Generating a SHA-2 signature for each segment
 - Sending segments to a storage service

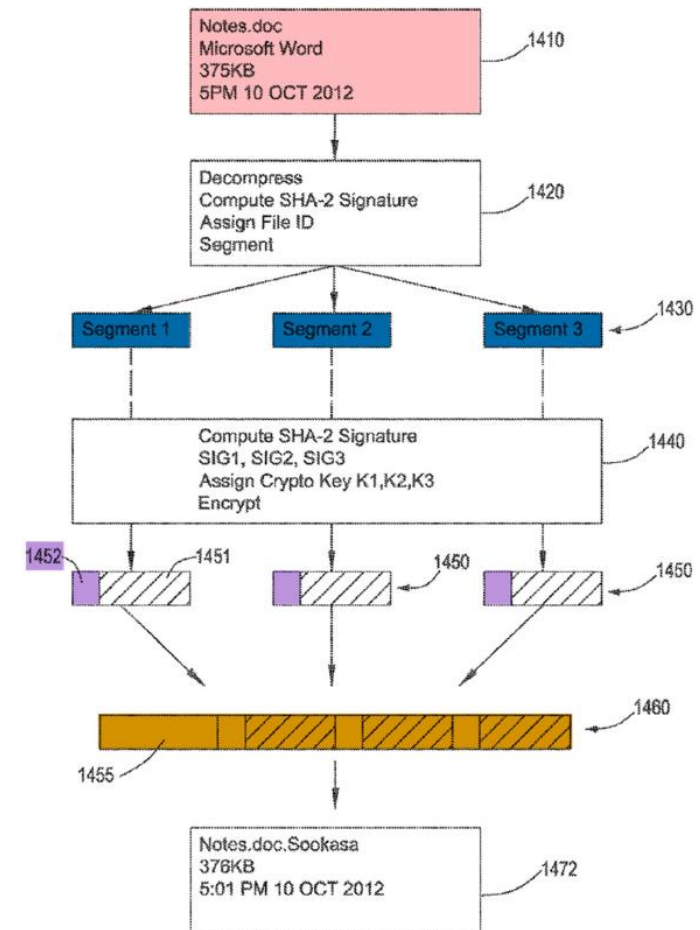


FIG. 13 (Annotated)

Ex. 1004 at Fig. 13 (annotated by Petitioner)

Overview of Cidon (Grounds 1-3)

- Cidon teaches a **filed-based** encryption policy where a file and all its descendent files will be encrypted

“[A] **specific file** was found to be under the encryption policy as defined by the file ancestry tree. This means that the original parent is encrypted and the policy dictates that all descendent files must be encrypted too.”

Ex. 1004 (Cidon) at ¶227

“Policies can be **defined on documents** which are ancestors of other documents.”

Ex. 1004 (Cidon) at ¶207

Overview of Shikfa (Grounds 1-3)

- Shikfa teaches a **client device** encrypting files and a middleware storing a **searchable encrypted index**
 - Each file is given an identifier and encrypted
 - The encrypted index maps encrypted keywords with identifiers

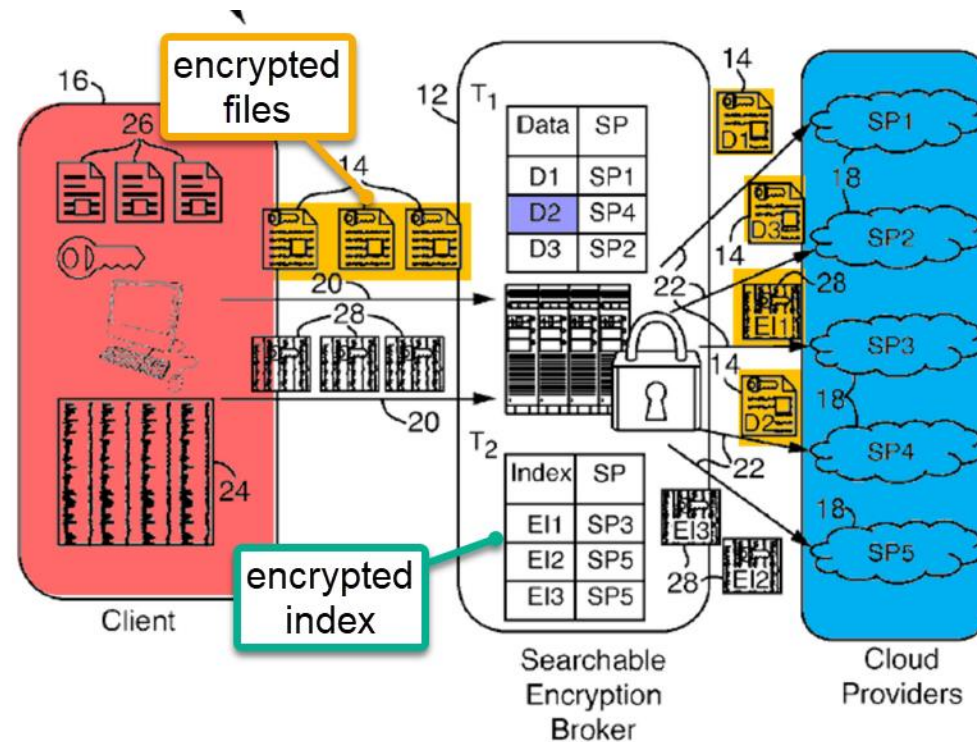


FIG. 2 (Annotated)

Ex. 1006 (Shikfa) at Fig. 2 (annotated by Petitioner and Patent Owner)

Pet. 20-21; POR 10-11

Overview of Petitioner's Cidon/Shikfa combination (Grounds 1-3)

- Petitioner proposes combining
 - Cidon's file segmentation and SHA-2 hashing and
 - Cidon's policies with
 - Shikfa's searchable encrypted index where the index maps encrypted keywords with chunks

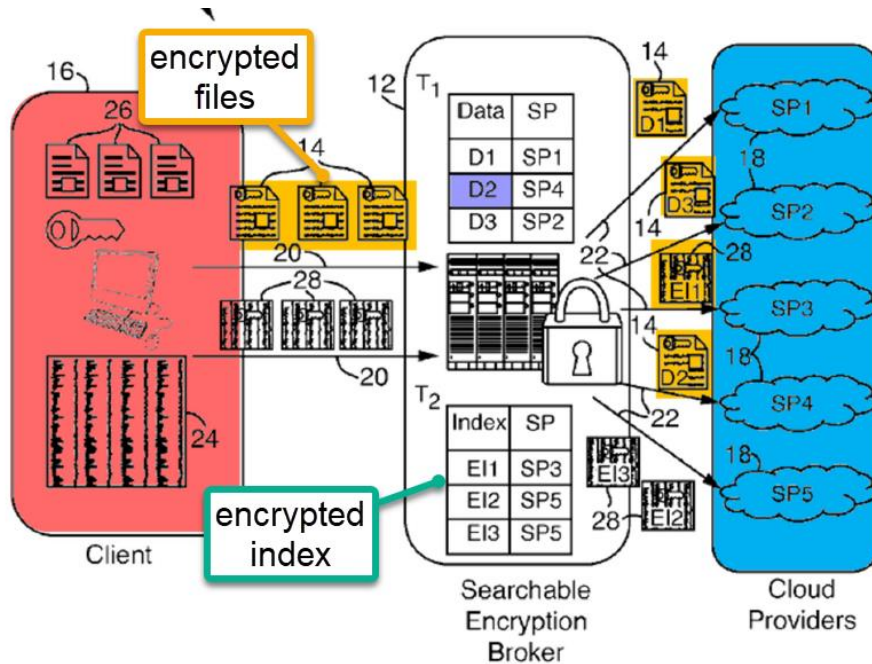


FIG. 2 (Annotated)

Ex. 1006 (Shikfa) at Fig. 2 (annotated by Petitioner and Patent Owner)

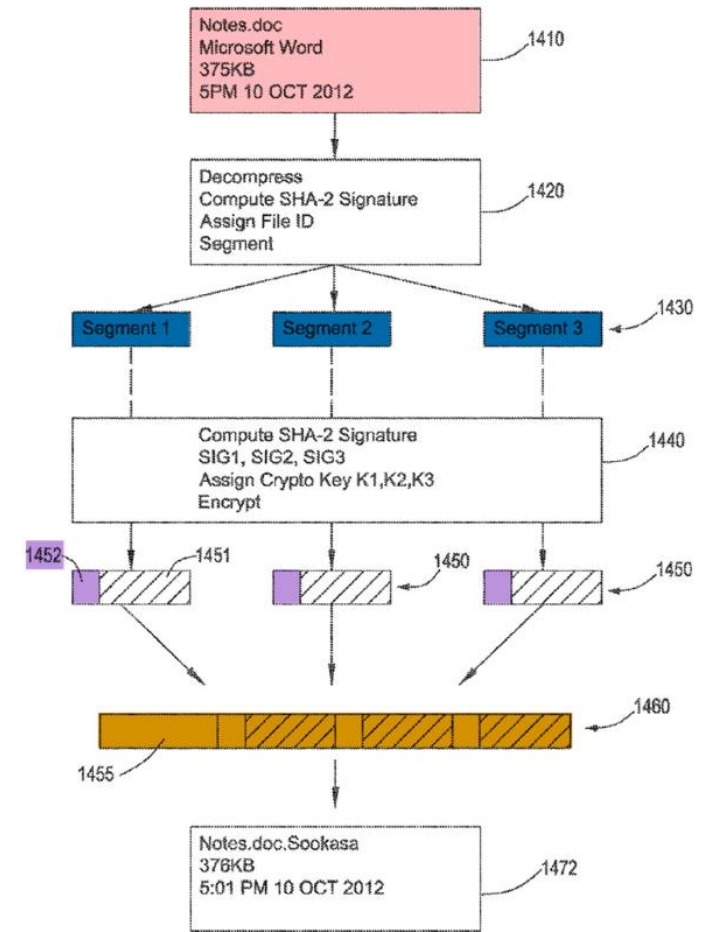


FIG. 13 (Annotated)

Ex. 1004 at Fig. 13 (annotated by Petitioner)

Overview of Auradkar (Grounds 4-5)

- Auradkar teaches encrypting documents and storing them in a server
 - A separate storage server stores an encrypted index for searching the document

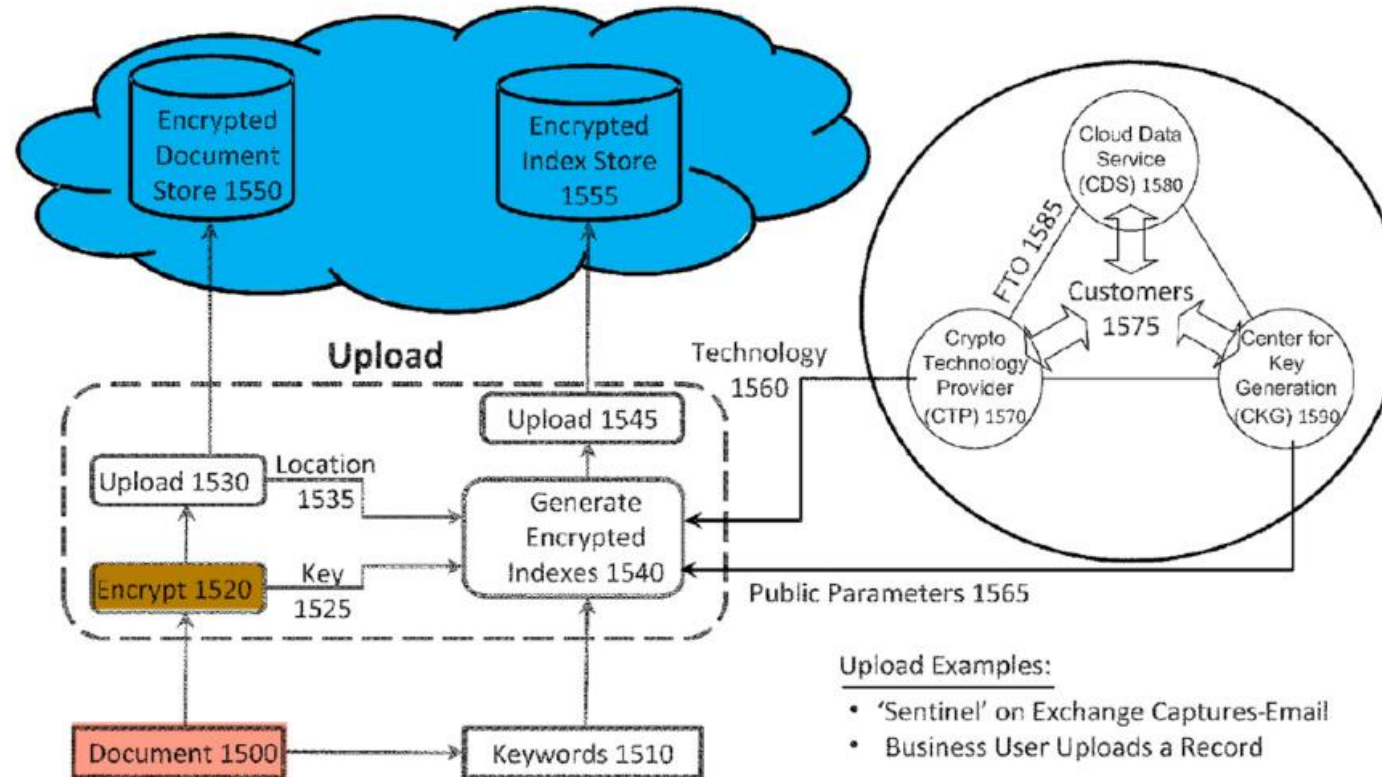


FIG. 15 (Annotated)

Ex. 1009 (Auradkar) at Fig. 15 (annotated by Petitioner)

Overview of Auradkar (Grounds 4-5)

- Auradkar encrypts documents and their metadata and tags and stores them in a searchable index

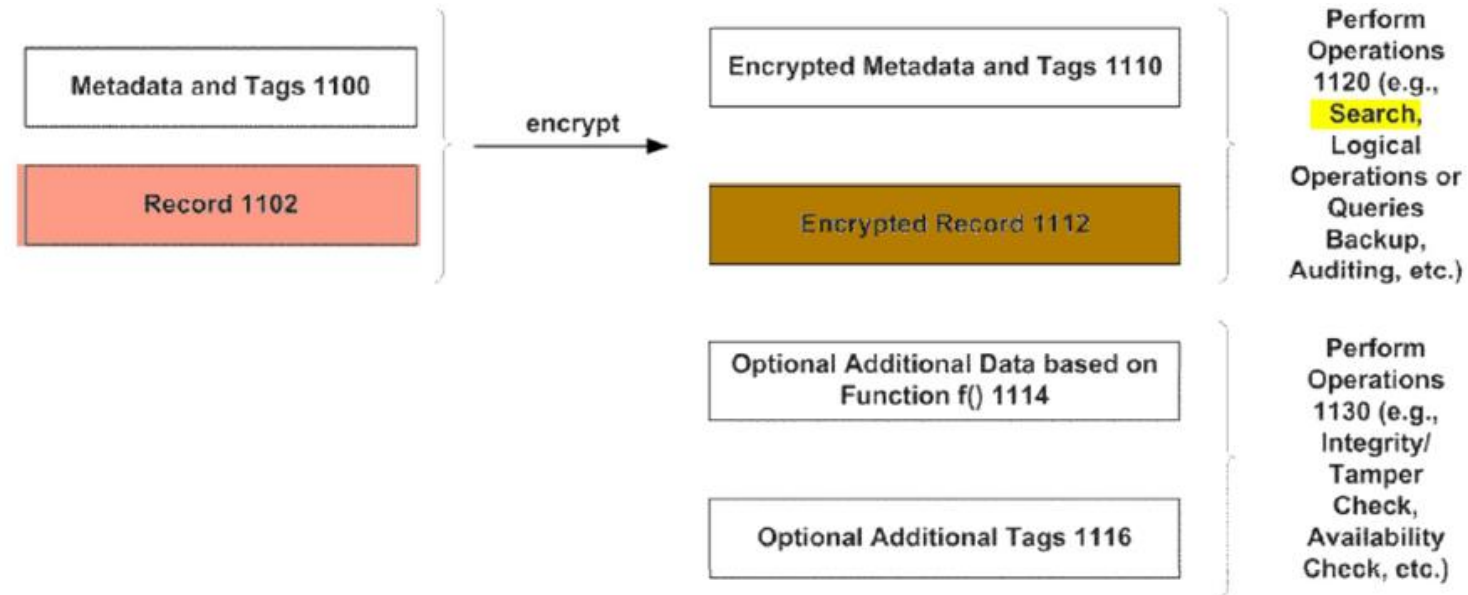


FIG. 11 (Annotated)

Ex. 1009 (Auradkar) at Fig. 11 (annotated by Petitioner)

Overview of Chiueh (Grounds 4-5)

- Chiueh describes an encrypted index where a client encrypts a document's pages and keywords
 - A server generates an encrypted index by giving each page an identifier and mapping the keywords to the page

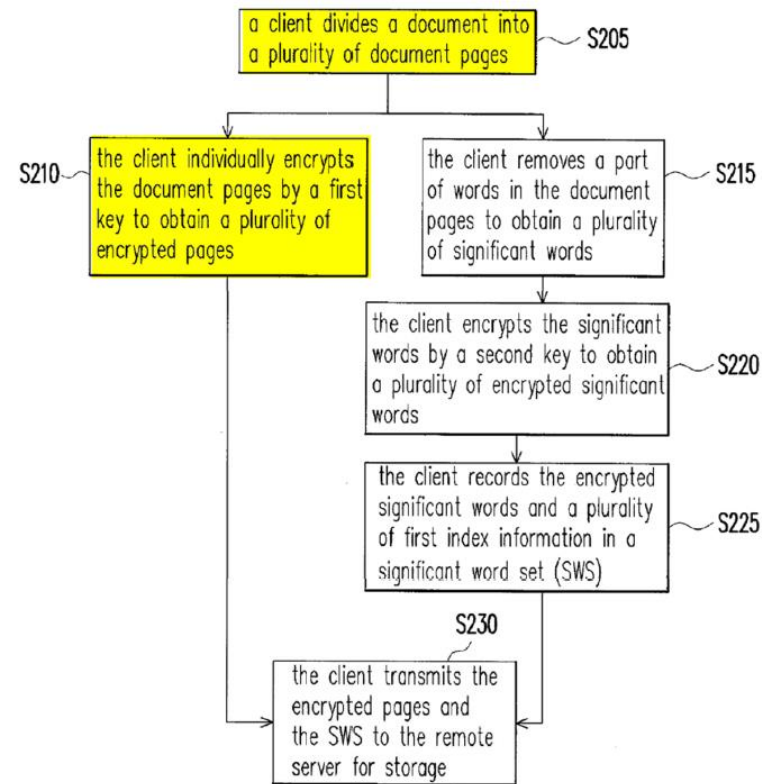


FIG. 2 (Annotated)

Ex. 1011 (Chiueh) at Fig. 2 (annotated by Petitioner)

Overview of Petitioner's Auradkar/Chiueh combination (Grounds 4-5)

- Petitioner proposes combining
 - Auradkar's policies
 - Auradkar's storage of encrypted files and encrypted indexes with
 - Chiueh's method of dividing and encrypted file and generating encrypted index

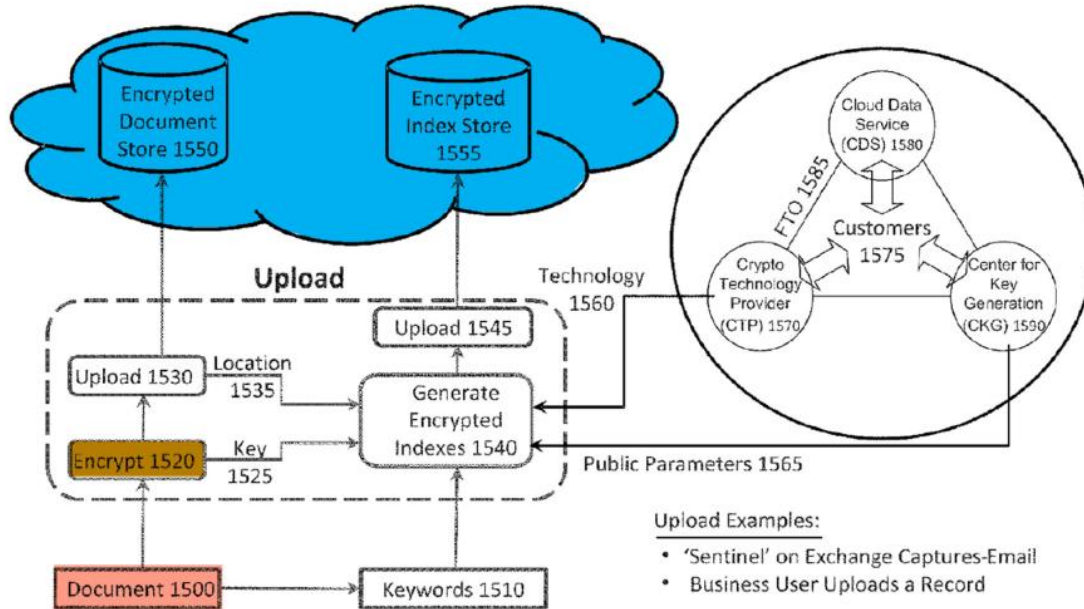


FIG. 15 (Annotated)

Ex. 1009 (Auradkar) at Fig. 15 (annotated by Petitioner)

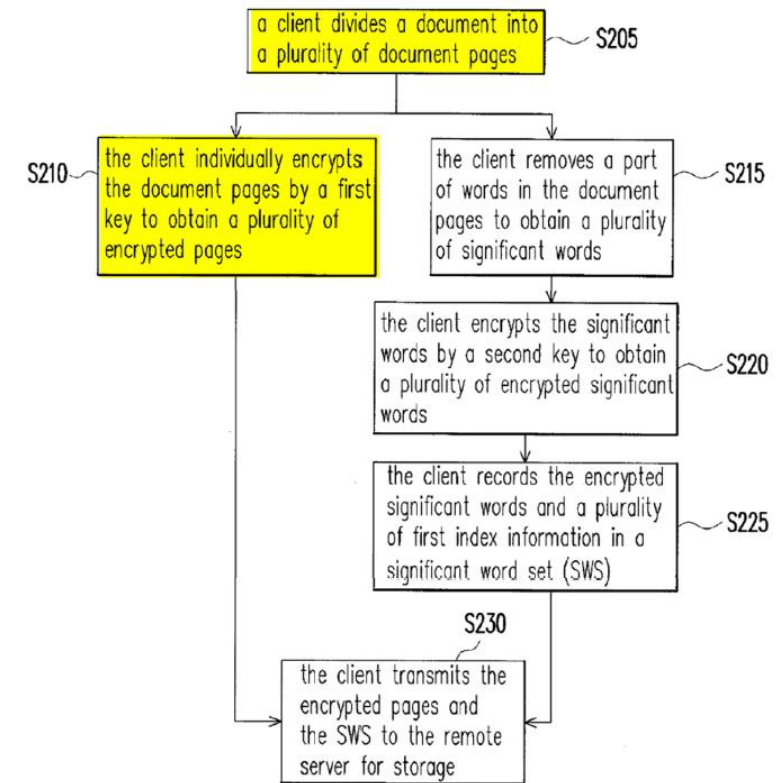


FIG. 2 (Annotated)

Ex. 1011 (Chiueh) at Fig. 2 (annotated by Petitioner)

**Cidon and Auradkar do not disclose “a policy” that performs
all recited functions
(All Grounds, All Claims)**

Summary of arguments for “a policy” that performs all recited functions

Petitioner’s Arguments

Claim Construction: “A policy” can be “one or more policies” that perform different recited functions

Cidon discloses an access policy and an encryption policy

Auradkar discloses an access policy and a policy for choosing encryption method

Patent Owner’s Arguments

Claim Construction: “a policy” is at least one policy that performs all recited functions

Cidon’s encryption policy is file-based and not assigned to the user

Cidon’s access policy is user-based and distinct from its encryption policy

Auradkar’s policy for encryption is only for documents and distinct from its user access policy

1. A method comprising:
assigning to one or more users, by a gateway device, **a policy** for managing access to and processing a file to be stored on one or more cloud platforms, wherein **the policy** defines access rights of the one or more users;
encrypting, by the gateway device, using cryptographic key information defined by **the policy**, content of the file to produce a searchable encrypted file by:
dividing the file into a plurality of chunks;
creating namespaces for one or more of the plurality of chunks; and
configuring the namespaces of the one or more chunks such that content of the file is encrypted in a manner that makes it searchable;
storing, by the gateway device, the searchable encrypted file on the one or more cloud platforms based on the **policy**; and
managing access to the searchable encrypted file by the one or more users based on **the policy**.

- Claims require at least one ***user-assigned*** policy for:
 - Managing access to a file
 - Processing a file
 - Defining access rights for a user
 - Defining cryptographic key information
 - Basing storage of files on a cloud platform

All Grounds, All Claims (“a policy”): Claim Construction

Salazar v. AT&T Mobility LLC, 64 F.4th 1311, 1318 (Fed. Cir. 2023)

- Claim language required “a microprocessor” performing various functions:

1. A communications, command, control and sensing system for communicating with a plurality of external devices comprising:

a microprocessor for generating a plurality of control signals used to operate said system, said microprocessor creating a plurality of reprogrammable communication protocols, for transmission to said external devices wherein each communication protocol includes a command code set that defines the signals that are employed to communicate with each one of said external devices;

a memory device coupled to said microprocessor configured to store a plurality of parameter sets retrieved by said microprocessor so as to recreate a desired command code set, such that the memory space required to store said parameters is smaller than the memory space required to store said command code sets;

a user interface coupled to said microprocessor for sending a plurality of signals corresponding to user selections to said microprocessor and displaying a plurality of menu selections available for the user's choice, said microprocessor generating a communication protocol in response to said user selections; and an infra-red frequency transceiver coupled to said microprocessor for transmitting to said external devices and receiving from said external devices, infra-red frequency signals in accordance with said communications protocols.

All Grounds, All Claims (“a policy”): Claim Construction

Salazar v. AT&T Mobility LLC, 64 F.4th 1311, 1318 (Fed. Cir. 2023)

- District Court held that “a microprocessor” means “one or more microprocessors” at least one which satisfies all limitations:

“The district court ... construed the term to mean ‘one or more microprocessors, at least one of which is configured to perform the generating, creating, retrieving, and generating functions.’ ...

Thus, the district court reasoned, “at least one microprocessor must satisfy all the functional (and relational) limitations recited for `said microprocessor.’”

Salazar v. AT&T Mobility LLC, 64 F.4th 1311, 1314 (Fed. Cir. 2023)

All Grounds, All Claims (“a policy”): Claim Construction

Salazar v. AT&T Mobility LLC, 64 F.4th 1311, 1318 (Fed. Cir. 2023)

- Salazar raised the same arguments as Petitioner:

“Put another way, in Mr. Salazar's view, a correct claim construction would encompass one microprocessor capable of performing one claimed function and another microprocessor capable of performing a different claimed function, even if no one microprocessor could perform all of the recited functions.”

Salazar v. AT&T Mobility LLC, 64 F.4th 1311, 1314 (Fed. Cir. 2023)

All Grounds, All Claims (“a policy”): Claim Construction

Salazar v. AT&T Mobility LLC, 64 F.4th 1311, 1318 (Fed. Cir. 2023)

- Federal Circuit rejected the argument and upheld the District Court’s construction:



“We agree with the district court that while the claim term ‘a microprocessor’ does not require there be only one microprocessor, the subsequent limitations referring back to ‘said microprocessor’ require that at least one microprocessor be capable of performing each of the claimed functions. This approach is *entirely consistent with our precedent*.

“[I]t does not suffice to have multiple microprocessors able to perform just one of the recited functions; the claim language requires at least one microprocessor capable of performing each of the recited functions.”

Salazar v. AT&T Mobility LLC, 64 F.4th 1311, 1318 (Fed. Cir. 2023); POR 38-39

All Grounds, All Claims (“a policy”): Claim Construction

TiVo, Inc. v. EchoStar Communications Corp., 516 F. 3d 1290 (Fed. Cir. 2008)

- Relevant claim language:

wherein said Output Section **assembles said video and audio components**
into **an MPEG stream**

TiVo, Inc. v. EchoStar Communications Corp., 516 F. 3d 1290, 1295 (Fed. Cir. 2008)

All Grounds, All Claims (“a policy”): Claim Construction

TiVo, Inc. v. EchoStar Communications Corp., 516 F. 3d 1290 (Fed. Cir. 2008)

- Appellant argued that “an MPEG stream” is “one or more MPEG streams,” including separate video and audio streams:

“TiVo argues that the ‘assembles’ limitation also covers the assembly of each component, audio and video, into its own **separate stream**. ...

TiVo argues that the 50X DVRs satisfy the ‘assembles’ limitation because the reference to ‘an MPEG stream’ must be interpreted to mean ‘**one or more MPEG streams**.’”

- Appellee argued audio and video must be one stream:

“EchoStar argues that the ‘assembles’ limitation covers only the assembly of audio and video components into a **single, interleaved MPEG stream**.’”

TiVo, Inc. v. EchoStar Communications Corp., 516 F. 3d 1290, 1303 (Fed. Cir. 2008)

All Grounds, All Claims (“a policy”): Claim Construction

TiVo, Inc. v. EchoStar Communications Corp., 516 F. 3d 1290 (Fed. Cir. 2008)

- Federal Circuit agreed with Appellee that it is one stream:



“[C]ontext *clearly indicates* that two separate components are assembled into a single stream, not that the video components are assembled into one stream and the audio components into a second stream.

“Those steps clearly describe the separation of a single stream into two components and then reassembly of the components into a single stream for conversion into television signals.”

TiVo, Inc. v. EchoStar Communications Corp., 516 F. 3d 1290, 1304 (Fed. Cir. 2008)

All Grounds, All Claims (“a policy”): Cidon/Shikfa do not disclose “a policy”

- Petitioner relies on different policies from Cidon:

Recited Function	Cidon’s Policy
Policy for managing access to a file	“enforce access policy for secure file access” (Cidon ¶102) (user-based policy)
Policy for ... processing a file	“encryption policy as defined by the file ancestry tree” (Cidon ¶224) (file-based policy)
The policy defines access rights of the ... users	“enforce access policy for secure file access” (Cidon ¶102) (user-based policy)
Encrypting ... using cryptographic key information defined by the policy	“encryption parameters for encrypting a file associated with a specific user” (Cidon ¶324) (user-based policy)
Storing ... the searchable encrypted encrypted file ... based on the policy	“[p]lacement [of files] in specific directories or shares” (Cidon ¶208) (file-based policy)

All Grounds, All Claims (“a policy”): Cidon/Shikfa do not disclose “a policy”

- Petitioner relies on two separate and distinct sets of policies: a **user-based** policy and a **file-based** policy
- Cidon’s **user-based** access policy:

“The module will check the credentials of the requesting application user and the file to determine if access is to be granted.”

Ex. 1004 (Cidon) at ¶108

- Cidon’s **file-based** encryption policy:

“Policies can be defined on documents which are ancestors of other documents.”

Ex. 1004 (Cidon) at ¶207

“[E]ncryption [i]s one of the policies you can define on a document.”

Ex. 2004 (Lee Tr.) at 57:6-9

“[File] B is a child of [file] A regardless who owns B or who owns A, because 192, 193, 194, did not say anything about that – did not require they belong to the same user...”

Ex. 2004 (Lee Tr.) at 48:14-19 Pet. 25; POR 32-33, 36

All Grounds, All Claims (“a policy”): Cidon/Shikfa do not disclose “a policy”

- Petitioner relies on two separate and distinct sets of policies: a ***user-based*** policy and a ***filed-based*** policy
- As explained by Patent Owner’s expert, they are not the same policy in multiple respects:

“Unlike an encryption policy which is defined by a file’s ancestry, access policies are defined by an administrator; an encryption policy is specific to a file, whereas an access policy is assigned for a user or users; and an encryption policy dictates whether a file must be encrypted, whereas an access policy determines if a user can be granted access to a file.”

Ex. 2003 (Black Decl.) at ¶109.

All Grounds, All Claims (“a policy”): Cidon/Shikfa do not disclose “a policy”

- Petitioner: Cidon discloses a “comprehensive policy framework” (Reply 18)
- There is **no** disclosure of a “comprehensive policy framework” in Cidon and Petitioner cites to none
- Petitioner does not argue it would be obvious
- Dr. Black explained why the policies could not be combined, which is undisputed:

“Assigning an encryption policy to a file is different from assigning an encryption policy to a user because, while a user-assigned policy would allow subjecting a file to different encryption requirements depending on the users, a file-assigned policy would require that certain files be encrypted regardless of who the users are. ...

It is also my opinion that a user-assigned policy would not be obvious in view of teachings that describe file-assigned policies. Again, **the concepts are fundamentally different.**”

Ex. 2003 (Black Decl.) at ¶¶101-02

All Grounds, All Claims (“a policy”): Cidon/Shikfa do not disclose “a policy”

- Petitioner also argues that Cidon’s **file-based** encryption policy is “assign[ed] to one or more users,” as claimed
- But Cidon’s “encryption policy” is **file-based** and cannot be assigned to a user

“[A] **specific file** was found to be under the encryption policy as defined by the file ancestry tree. This means that the original parent is encrypted and the policy dictates that all descendent files must be encrypted too.”

Ex. 1004 (Cidon) at ¶227

“Policies can be **defined on documents** which are ancestors of other documents.”

Ex. 1004 (Cidon) at ¶207

“[E]ncryption [i]s one of the policies you can **define on a document**.”

Ex. 2004 (Lee Tr.) at 57:6-9

All Grounds, All Claims (“a policy”): Auradkar/Chiueh do not disclose “a policy”

- Petitioner relies on different policies from Auradkar:

Recited Function	Auradkar’s Policy
Policy for managing access to a file	“[E]nterprise 620 define[d] its enterprise access policy 630” (Auradkar ¶117) (user-based)
Policy for ... processing a file	“enterprise access policy 630 ... affect how data 512 [is] manipulated” (Auradkar ¶117) (user-based)
The policy defines access rights of the ... users	“enterprise access policy 630 ... affect how data 512 [is] accessed” (Auradkar ¶117) (user-based)
Encrypting ... using cryptographic key information defined by the policy	“a policy ... that requires that (sensitive) medical data ... us[es] AES 256 encryption” (Auradkar ¶96) (file-based)
Storing ... the searchable encrypted encrypted file ... based on the policy	“a policy on ... where [data] could be stored” (Auradkar ¶96) (file-based)

All Grounds, All Claims (“a policy”): Auradkar/Chiueh do not disclose “a policy”

- Petitioner relies on two separate and distinct sets of policies: a ***user-based*** policy and a ***document-based*** policy
- Auradkar’s ***user-based*** access policy:

“[W]hen users 508 of data 512 attempt[ed] to access such data 512, various crypto access controls 514 guided by, but separated from, enterprise access policy 530 protect[ed] the data 512 from unwarranted access by users 508.

Ex. 1009 (Auradkar) at ¶117

- Auradkar’s ***document-based*** encryption policy:

“[A] document may exist that has Public, Secret and Top Secret paragraphs. A paragraph’s classification can also be used to determine encryption mechanism, key and access policy. ... [A] policy could be created that requires that (sensitive) medical data must be backed up once a day using AES 256 encryption to a secure server in a trusted data center.”

Ex. 1009 (Auradkar) at ¶¶95-96

All Grounds, All Claims (“a policy”): Auradkar/Chiueh do not disclose “a policy”

- Petitioner on Reply: Auradkar’s encryption policy is “not limited to classifications of [document] paragraphs” (Reply 31)
- Auradkar only discloses encryption for specific paragraphs of a document:

“[A] document may exist that has Public, Secret and Top Secret paragraphs. A paragraph’s classification can also be used to determine encryption mechanism, key and access policy. ... [A] policy could be created that requires that (sensitive) medical data must be backed up once a day using AES 256 encryption to a secure server in a trusted data center.”

Ex. 1009 (Auradkar) at ¶¶95-96

**Prior art does not disclose “namespace”
(All Grounds, All Claims)**

Summary of arguments for creating and configuring “namespaces”

Petitioner’s Arguments

Claim Construction: “namespace” is “a set of names available for naming things”

Cidon/Shikfa creates a namespace when it creates an identifier for a file chunk

Cidon/Shikfa configures a namespace when it maps keywords to the identifier

Auradkar/Chiueh creates a namespace when it creates a tag or identifier for a file chunk

Auradkar/Chiueh configures a namespace when it maps keywords to the tag or identifier

Patent Owner’s Arguments

Claim Construction: “namespace” is a “set of names available within a *particular context*”

An identifier is not a namespace

An identifier does not define a context

Mapping keywords to an identifier does not configure the namespace

All Grounds, All Claims (“namespace”): Claim Construction

- Parties partially agree that “namespace” means “a set of names available for naming things”
- Parties dispute whether the “set of names” must be for “a particular context”

“namespace”	
Petitioner’s Definition	Patent Owner’s Definition
“a set of names available for naming things”	“a set of names available <i>within a particular context</i> ”

- Petitioner ignores the specification

“Examples of storage containers include, but are not limited to, directories, directory structures, web directories, folders, smart folders, files and namespaces of same.”

Ex. 1001 (678 Patent) at 6:44-47

“[O]ne or more containers (e.g., directories, folders, files and their associated namespaces) of cloud service providers 206 can store metadata and/or content associated with searchable encrypted files.”

Ex. 1001 (678 Patent) at 10:12-15

All Grounds, All Claims (“namespace”): Claim Construction

- Petitioner’s expert repeatedly testified that the plain meaning of “namespace” provides a particular context or scope

“A namespace ensures that every object **within the context of the namespace** can be identified uniquely by its name(s).”

Ex. 1002 (Lee Decl.) at ¶42

“[T]he namespace **does have a scope** of the objects that it’s identifying.”

Ex. 2004 (Lee Tr.) at 35:10-12

“[W]hen you talk about namespace, *first of all* you talk about **what’s a scope of this current namespace**.”

Ex. 2004 (Lee Tr.) at 37:1-5

“The namespace is [the directory] ‘documents’. That means **the scope** is everything within the directory ‘documents’.”

Ex. 2004 (Lee Tr.) at 37:4-5

All Grounds, All Claims (“namespace”): Claim Construction

- Petitioner ignores the specification and relies exclusively on a dictionary definition
 - But the dictionary definition expressly provides the same name can be used in multiple contexts:

namespace

1. the set of names available for naming things such as files, variables in a program, or computers in a network. If two parts of a program have different namespaces, the same name can be used in both places for different purposes without conflict. *See also* LOCAL VARIABLE.

Ex 1014

All Grounds, All Claims (“namespace”): Claim Construction

- Petitioner argues “within a particular context” is “vague and encapsulates practically anything” (Reply 2)
- “context” has a clear, accepted meaning, reflected in the patent’s specification:

“Hence, naming the folder with the encrypted filename and storing the file chunks in that folder creates a **namespace scope** for the file chunks.”

Ex. 2003 (Black Supp. Decl.) at ¶71

“[O]ne or more **containers** (e.g., **directories, folders, files and their associated namespaces**) of cloud service providers 206 can store metadata and/or content associated with searchable encrypted files.”

Ex. 1001 (678 Patent) at 10:12-15

“The namespace is [the directory] ‘documents’. That means **the scope** is everything within the directory ‘documents’.”

Ex. 2004 (Lee Tr.) at 37:4-5

All Grounds, All Claims (“namespace”): Claim Construction

- In the specification, a container (or folder) is created (“creating a namespace”)
- Then, file chunks are named and stored in the folder (“configuring a namespace”)

“In the present example, the top-most portion of each container may represent a folder within which the other portions are included in the form of files. The name of the folder may be the encrypted version of the original file name and the names of the files may include one or more searchable indices.”

Ex. 1001 (678 Patent) at 13:2-7

“Each container can include multiple encrypted file chunks, which may be named by means of searchable indices as mentioned in connection with the description of FIGS. 3A-B. For instance, container 1 can include four file chunks having their names (adfbhlbewfuwed, ewuiyrt, hdweuf, and uetywo).”

Ex. 1001 (678 Patent) at 12:67-13:2

All Grounds, All Claims (“namespace”): Claim Construction

- Petitioner argues that “creating namespaces” and “configuring namespaces” are “**not** part of encrypting” step (Reply 2)
 - Claim language is clear that it is:

1. A method comprising:
assigning to one or more users, by a gateway device, a policy for managing access to and processing a file to be stored on one or more cloud platforms, wherein the policy defines access rights of the one or more users;
encrypting, by the gateway device, using cryptographic key information defined by the policy, content of the file to produce a searchable encrypted file by:
dividing the file into a plurality of chunks;
creating namespaces for one or more of the plurality of chunks; and
configuring the namespaces of the one or more chunks such that content of the file is encrypted in a manner that makes it searchable;
storing, by the gateway device, the searchable encrypted file on the one or more cloud platforms based on the policy; and
managing access to the searchable encrypted file by the one or more users based on the policy.

**Cidon in view of Shikfa does not “create” or “configure” a
“namespace”
(All Grounds, All Claims)**

All Grounds, All Claims (“namespace”): Petitioner’s Cidon/Shikfa mapping

- Cidon/Shikfa: For each file, Petitioner argues that—
 - “creating a namespace” is either creating an identifier for each file segment or creating multiple identifiers
 - “configuring a namespace” is mapping encrypted keywords to identifiers in an index

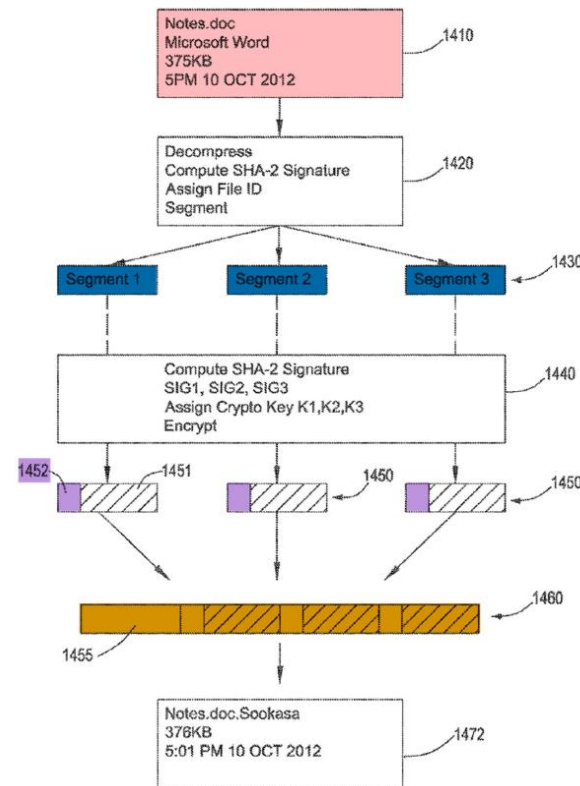


FIG. 13 (Annotated)

All Grounds, All Claims (“creating namespaces”): Cidon/Shikfa

- Petitioner argues that creating a single “identifier” is “creating a namespace”, and does not require defining a range of names

“Cidon disclosed this limitation because it described calculating (and creating) a “unique” file segment identifier (or file segment signature) for each of file segments (chunks) based on the content of a file segment using, e.g., SHA2 signature computation.”

Pet. 36-37

“Namespace creation does not require defining a range of names as PO argues.”

Reply 6

All Grounds, All Claims (“creating namespaces”): Cidon/Shikfa

- A single identifier is not a “namespace”
- Both parties **agree** that namespace means “**set of names** available for naming things”
- Petitioner’s expert agrees:

“[A] namespace normally tells you the naming convention and the **range of ... values.**”

Ex. 2005 (Lee Tr.) at 32:7-9

All Grounds, All Claims (“creating namespaces”): Cidon/Shikfa

- Generating a single identifier does not create names “***available for naming things***”
- According to Petitioner’s own expert, namespace is created ***first*** and then identifiers are generated from it:

A namespace is the “naming convention” and defines “a range of ... values” from which an “**identifier [is] chosen**”

Ex. 2005 (Lee Tr.) at 32:7-21

“[A] flat identifier might be **a number chosen** uniformly at random from $[0, 2^{128} - 1]$).”

Ex. 1023 at 344

All Grounds, All Claims (“creating namespaces”): Cidon/Shikfa

- Generating identifiers does not generate a “**context**”
- Cidon uses the SHA-2 signature as a **name** for a file chunk, not a namespace

“A namespace defines the set of allowable names within a particular context. Standing alone, an identifier is simply a single item within the set of allowed names.”

Ex. 2003 (Black Supp. Decl.) at ¶80

All Grounds, All Claims (“creating namespaces”): Cidon/Shikfa

- Petitioner argues on Reply that assignment of multiple identifiers to file segments creates “a unique set of names within the context of encrypted file 1460.” Reply 8.
- But the encrypted file 1460 is created **after** the file segments are given identifiers, thus the assignment of identifiers does not **create context**.

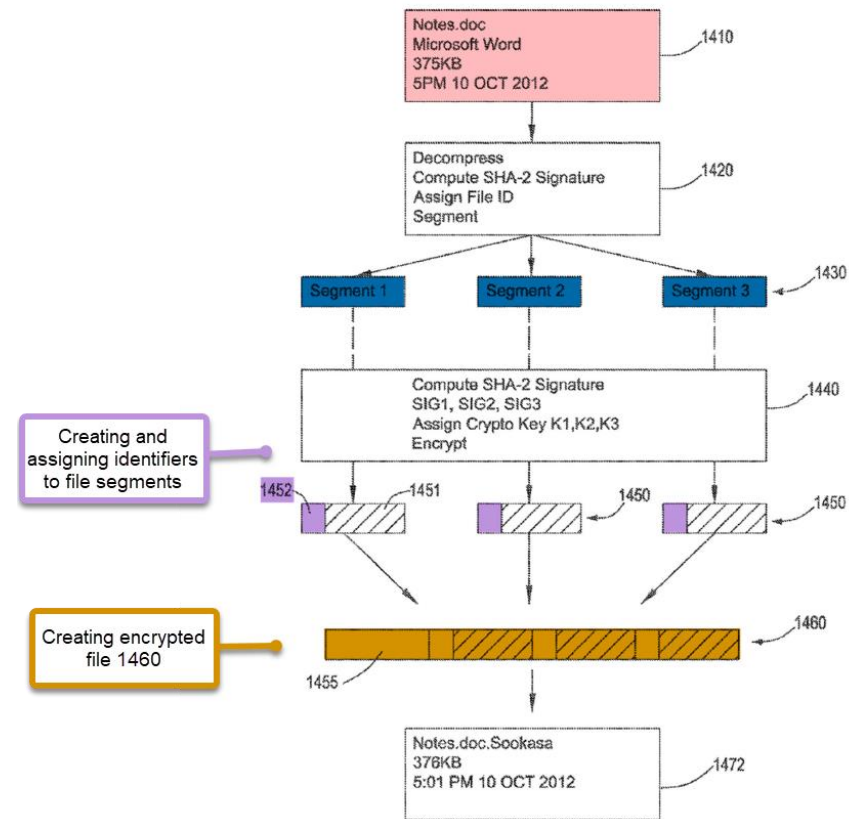


FIG. 13 (Annotated)

All Grounds, All Claims (“configuring namespaces”): Cidon/Shikfa

- Petitioner argues that mapping keywords to identifiers in a separate index is “configuring namespaces”
- Using the identifiers in an index does not “configure” the namespace

“[I]n the proposed Cidon-Shikfa combination, the SHA2 identifiers are used (not configured) within Shikfa’s indexes as pointers to file chunks. No configuration of the alleged namespace is performed.”

Ex. 2003 (Black Supp. Decl.) at ¶84

All Grounds, All Claims (“configuring namespaces”): Cidon/Shikfa

- Patent requires “configuring the namespaces ... such that ... file is encrypted in a manner that makes it searchable”
- Petitioner’s expert agreed that Cidon’s mapping of keywords to identifiers is “orthogonal” and “independent” to how file is encrypted

1. A method comprising:
assigning to one or more users, by a gateway device, a policy for managing access to and processing a file to be stored on one or more cloud platforms, wherein the policy defines access rights of the one or more users;
encrypting, by the gateway device, using cryptographic key information defined by the policy, content of the file to produce a searchable encrypted file by:
dividing the file into a plurality of chunks;
creating namespaces for one or more of the plurality of chunks; and
configuring the namespaces of the one or more chunks such that content of the file is encrypted in a manner that makes it searchable;
storing, by the gateway device, the searchable encrypted file on the one or more cloud platforms based on the policy; and
managing access to the searchable encrypted file by the one or more users based on the policy.

20 A. So, again, if you read what they say
21 about index, right, index talks about -- here the
22 keywords -- here the -- here the IDs or pointers
23 to the encrypted documents that has the keywords.
24 So that -- you know, I mean, really that **index**
25 itself when you -- you know, is basically it's
1 **orthogonal to the encryption algorithm they use to**
2 **encrypt the documents**, because this already says,
3 hey, this encrypted document has those keywords,
4 right?
5 And then, like I said, the index table
6 later on encrypted using the searchable encryption
7 algorithm.

Ex. 2004 (Lee Tr.) at 136:20-137:7

POR 19-21, 25-27; Surreply 5-7, 12-13

All Grounds, All Claims (“configuring namespaces”): Cidon/Shikfa

- Patent requires “configuring the namespaces ... such that ... file is encrypted in a manner that makes it searchable”
- Dr. Black explains the technical distinction between Shikfa and the ‘678 Patent

“[W]hereas the prior art describes the use of external encrypted indexes, the ‘678 Patent describes a novel process where an encrypted index is embedded into the encrypted file by using the encrypted keywords as filenames for the file’s chunks. As a result, queries can now be performed using a cloud storage service’s built-in searching mechanism. ... **The encrypted file itself becomes searchable**. Indeed, the ‘678 Patent explains that the users can ‘download[] the searchable encrypted files ..., which would then allow the user 202b to search metadata and/or content associated with the searchable encrypted files even in offline mode when cloud service providers 206 are inaccessible via a network connection.’ Ex. 1001, 10:23-28. **Such portability is not possible using Shikfa’s teachings**, because the encrypted files must be accompanied by the index in order to be searched—the encrypted files themselves are not searchable.

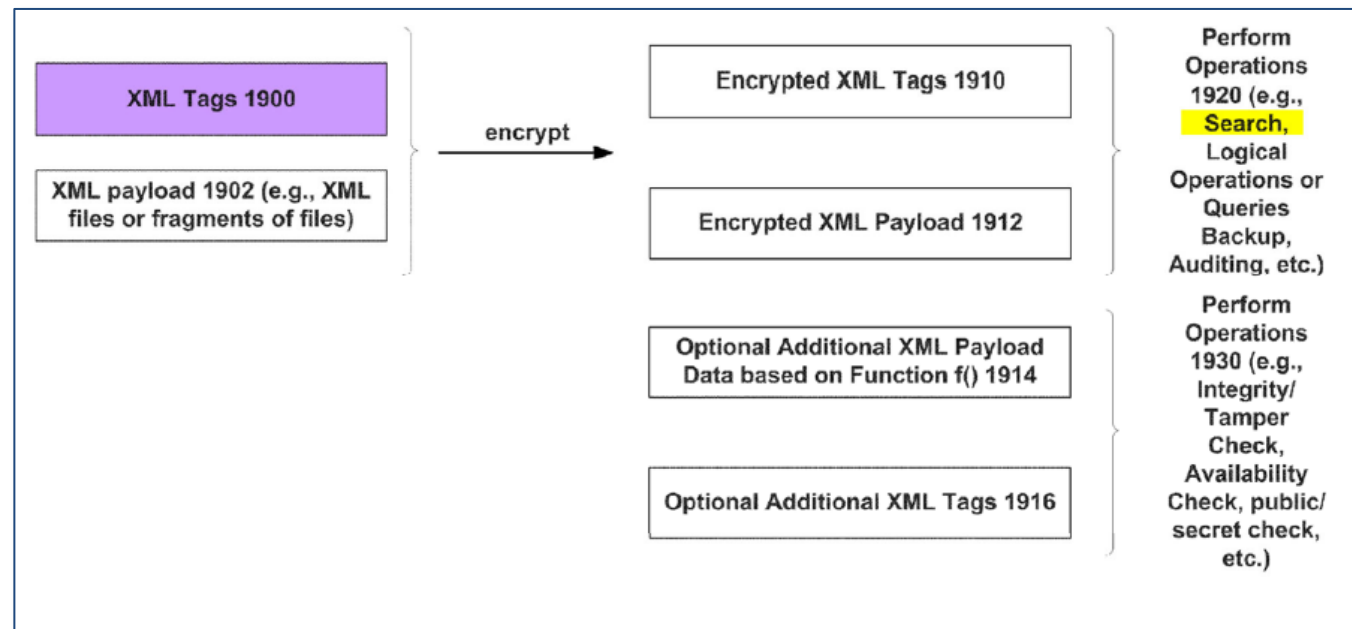
Ex. 2003 (Black Supp. Decl.) at ¶¶87

POR 19-21, 23-27; Surreply 5-7, 10-13

**Auradkar in view of Chiueh does not “create” or “configure”
a “namespace”
(All Grounds, All Claims)**

All Grounds, All Claims (“namespace”): Petitioner’s Auradkar/Chiueh mapping

- Auradkar/Chiueh: For each file, Petitioner argues that—
 - “creating a namespace” is “creat[ing] tags for payload”, e.g. “XML tags” or “document ID”
 - “configuring a namespace” is “allow[ing] search” e.g. by “creating an index”



All Grounds, All Claims (“creating namespace”): Auradkar/Chiueh

- Petitioner argues that creating an “XML tag” or “document ID” is “creating a namespace”

“Auradkar disclosed creating tags for payloads (e.g., payload 1200, XML fragment 1902 (chunks)), such as tag 1200 that included the filename of the payload (FName: fox.tx) and XML tags 1900, and encrypting them. ... [T]hese tags constituted namespaces for the payloads.”

Pet. 95-96

“Chiueh’s system also created and assigned a document identification or ID (namespace) to each of the encrypted pages.”

Pet. 96

All Grounds, All Claims (“creating namespace”): Auradkar/Chiueh

- Generating an XML tag or identifier does not create “a set of names ***available for naming***”
- Petitioner conflates creating a namespace (the “set”) with generating a name from within that namespace (a single element from within the “set”)

“The plain meaning of ‘namespace’ should be ‘the **set of names available** for naming things.’”

Reply 1

“[A] namespace normally tells you the naming convention and the **range of ... values**” from which “an identifier [is] chosen.”

Ex. 2005 (Lee Tr.) at 32:7-21

“[A] flat **identifier** might be a number **chosen** uniformly at random **from $[0, 2^{128} - 1]$** .”

Ex. 1023 at 344

All Grounds, All Claims (“creating namespaces”): Auradkar/Chiueh

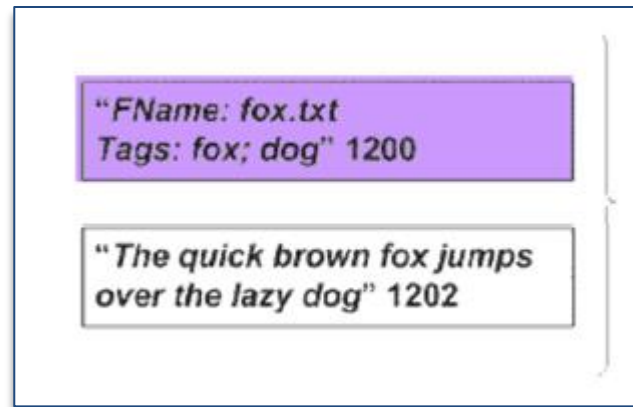
- Generating an XML tag or identifier does not generate any context
- At best, an XML tag or identifier is a name within the “set of names,” but does not itself provide any context

“A namespace defines the set of allowable names within a particular context. Standing alone, an identifier is simply a single item within the set of allowed names.”

Ex. 2003 (Black Supp. Decl.) at ¶80

All Grounds, All Claims (“creating namespaces”): Auradkar/Chiueh

- Petitioner on Reply: identifiers assigned to file segments “bec[o]me a set of unique names within the context of the encrypted file(s).” Reply 23-24.
- But the set of names, *i.e.* XML tags or document IDs, are contextless
 - XML tags are just keywords, e.g. “fox; dog”
 - Document IDs identify any document, so lack context/scope for the set of names



All Grounds, All Claims (“configuring namespaces”): Auradkar/Chiueh

- Petitioner argues that mapping keywords to identifiers in a separate index is “configuring namespaces”
- Using the identifiers in an index does not “configure” the namespace

“[I]n the proposed Auradkar-Chiueh combination, the document identifiers (filenames) are used (not configured) within Chiueh’s indexes as index information in order to locate a document page. No configuration of the alleged namespace is performed, because the alleged namespace.”

Ex. 2003 (Black Supp. Decl.) at ¶133

All Grounds, All Claims (“configuring namespaces”): Auradkar/Chiueh

- Patent requires “configuring the namespaces ... such that ... file is encrypted in a manner that makes it searchable”
- Petitioner’s expert agreed that mapping of keywords to identifiers is “orthogonal” and “independent” to how file is encrypted

1. A method comprising:
assigning to one or more users, by a gateway device, a policy for managing access to and processing a file to be stored on one or more cloud platforms, wherein the policy defines access rights of the one or more users;
encrypting, by the gateway device, using cryptographic key information defined by the policy, content of the file to produce a searchable encrypted file by:
dividing the file into a plurality of chunks;
creating namespaces for one or more of the plurality of chunks; and
configuring the namespaces of the one or more chunks such that content of the file is encrypted in a manner that makes it searchable;
storing, by the gateway device, the searchable encrypted file on the one or more cloud platforms based on the policy; and
managing access to the searchable encrypted file by the one or more users based on the policy.

20 A. So, again, if you read what they say
21 about index, right, index talks about -- here the
22 keywords -- here the -- here the IDs or pointers
23 to the encrypted documents that has the keywords.
24 So that -- you know, I mean, really that **index**
25 itself when you -- you know, is basically it's
1 **orthogonal to the encryption algorithm they use to**
2 **encrypt the documents**, because this already says,
3 hey, this encrypted document has those keywords,
4 right?
5 And then, like I said, the index table
6 later on encrypted using the searchable encryption
7 algorithm.

Ex. 2004 (Lee Tr.) at 136:20-137:7

POR 19-21, 47-48; Surreply 5-7, 21-22

**No motivation to combine Cidon and Shikfa
(Grounds 1-3, All Claims)**

Summary of arguments for lack of motivation to combine Cidon/Hermann with Shikfa

Petitioner's Arguments

Shikfa's encryption method enhances Cidon/Hermann's server by adding encrypted search

Patent Owner's Arguments

Shikfa expressly states that encryption must occur on client device, cannot occur on a server due to security

Shikfa teaches away from Cidon/Hermann

No motivation to combine Cidon/Hermann with Shikfa

- Petitioner proposes using Shikfa's encryption teachings with Cidon/Hermann's server
- Shikfa is **explicit** that encryption must occur on the client, not server, to maintain confidentiality

“The broker system 10 also ensures that the broker server 12 cannot access and/or discover the content of the client's data because the broker server 12 does not know the encryption used on the encrypted documents 14. Thus, the broker system 10 provides searchable encryption while maintaining the confidentiality of data external to the client 16.

Ex. 1006 (Shikfa) at ¶48

“Just basically, if you look at figures of Shifka, it's pretty clear that the client performs the encryption, and then once it gets the data back, it performs the decryption.”

Ex. 2004 (Lee Tr.) at 148:22-25

No motivation to combine Cidon/Hermann with Shikfa

- Shikfa teaches away from the combination



“[A reference] teaches away when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken in the claim.”

Meriesonne v. Google, Inc., 849 F.3d 1379, 1382 (Fed. Cir. 2017); POR 30



The Board should not “ignore[] the express disclosure ... that teaches away from the claimed invention.”

Chermours Co. FC, LLC v. Daikin Industries, 4 F.4th 1370, 1376 (Fed. Cir. 2021); POR 53

**No motivation to combine Auradkar and Chiueh
(Grounds 4-5, All Claims)**

Summary of arguments for lack of motivation to combine Auradkar with Chiueh

Petitioner's Arguments

Chiueh's encryption method enhances Auradkar server by adding encrypted search

Patent Owner's Arguments

Chiueh expressly states that encryption must occur on client device, cannot occur on a server due to security

Chiueh teaches away from Auradkar

Grounds 1-3, All Claims: No motivation to combine Auradkar with Chiueh

- Petitioner proposes using Chiueh's encryption processes with Auradkar's server
- Chiueh is **explicit** that encryption must occur on the client, not unreliable server

“For example, the conventional remote server should have a decryption key to convert an encrypted document into a plaintext, and then search the plaintext for the keyword. However, the remote server is probably not reliable. In case that the remote server has the decryption capability, the client could not prevent an unfaithful query processing of the remote server. Namely, the document stored in the remote server could be peeked/leaked.

Ex. 1011 (Chiueh) at ¶5

Grounds 1-3, All Claims: No motivation to combine Auradkar with Chiueh

- Chiueh teaches away from the combination



“[A reference] teaches away when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken in the claim.”

Meriesonne v. Google, Inc., 849 F.3d 1379, 1382 (Fed. Cir. 2017); POR 30



The Board should not “ignore[] the express disclosure ... that teaches away from the claimed invention.”

Chermours Co. FC, LLC v. Daikin Industries, 4 F.4th 1370, 1376 (Fed. Cir. 2021); POR 53

**Cidon and Auradkar do not process a file “to be
stored/uploaded”
(All Grounds, All Claims)**

Summary of arguments for “processing a file to be stored/uploaded”

Petitioner’s Arguments

Petition: Alleges “encryption” is “processing” but does not allege encryption on file “*to be* stored”

Reply: “Processing a file” could be anything

Patent Owner’s Arguments

Cidon only discloses that the file is encrypted *after* it is stored/uploaded

All Grounds, All Claims (“processing a file to be stored/uploaded”): Cidon

- Petitioner mapped Cidon’s “encryption” of a file to the “processing a file” limitation.
- But Cidon is clear that the encryption occurs **after** the file is stored/uploaded to the cloud service.
- Petitioner’s expert agreed.

“The management server is the entity that encrypts files on the user Dropbox accounts.”

Ex. 1004 (Cidon) at ¶223

15 So I think -- I think if you read 222
16 and sort of understand what the patent is saying
17 is that, hey, look, the user can access the files
18 through the -- let's say the application, let's
19 say from Dropbox as if nothing were going to
20 bother them, nothing is in the way, but then the
21 management server would monitor the file that
22 should be encrypted. And then after a while the
23 file has not been changed, meaning that the user
24 is not actively changing the content and because
25 the file is in that -- in that special folder or
1 secured folder, then go ahead and encrypt that and
2 then remove the -- erase the unencrypted file. I
3 mean, that's what it's talking about, I think.

Ex. 2004 (Lee Tr.) at 67:15-68:3

All Grounds, All Claims (“processing a file to be stored/uploaded”): Auradkar

- Petitioner mapped Auradkar’s access policy to the “processing a file” limitation.
- But Auradkar is clear that Auradkar’s policy is for files **after** they are stored/uploaded to the cloud service.

“Enterprise access policy 530 can affect how data 512 stored in a data container 510 of a cloud container 500 can be accessed, manipulated, retrieved, searched, etc. Accordingly, when users 508 of data 512 attempt to access such data 512, various crypto access controls 514 guided by, but separated from, enterprise access policy 530 protect the data 512 from unwarranted access by users 508.”

Ex. 1009 (Auradkar) at ¶117

Cidon does not disclose storing a file “across two cloud platforms”

(Ground 1, Claims 11 and 25)

Ground 1, Claims 11 and 25 (“across two cloud platforms”)

Summary of arguments for “across two or more cloud platforms”

Petitioner’s Arguments

“Backup” and “archiving” stores a file “across two or more cloud platforms”

Patent Owner’s Arguments

“Across” means the file chunks are stored on different platforms, not a copy.

Ground 1, Claims 11 and 25 (“across two cloud platforms”)

- Plain meaning of storing “across” multiple platforms is consistent with specification’s disclosure of storing different file chunks on different platforms, i.e. striping

“Clients may also store such content across two or more stores 114 such that the content is divided into multiple chunks and each chunk is stored on a different cloud store 114.”

Ex. 1001 (678 Patent) at 7:43-45

Ground 1, Claims 11 and 25 (“across two cloud platforms”)

- Both experts agree that archiving and backup create full-copies of the file, it does not stripe the file

“So ‘backup’ normally means that I let’s say **put a copy** of the -- of your file [in] some kind of backup.”

Ex. 2004 (Lee Tr.) at 107:24-108:1

“[I]t is my opinion that archiving and backup of a file does not constitute splitting up a file into chunks and storing chunks on different platforms. Instead, archiving and backup would be understood to mean **merely storing a copy** of the file to a different store.”

Ex. 2003 (Black Supp. Decl.) at ¶118

Ground 1, Claims 11 and 25 (“across two cloud platforms”)

- Petitioner’s citation to the ‘678 Patent’s “redundancy” does not imply a “copy”

“A POSITA would understand that storing chunks across multiple cloud services could also involve providing redundancy. For example, RAID 5 was a known method of storing data across multiple disks that also provided for redundancy.”

Surreply 18

**Inoue does not disclose “storing content of the file as
filenames”**

(Ground 5, Claim 6)

Ground 5, Claim 6 (“storing content of the file as filenames”)

Summary of arguments for Inoue’s lack of disclosure of “storing content of the file as filenames”

Petitioner’s Arguments

Inoue teaches generating a keyword from a file and using the keyword as a filename

Patent Owner’s Arguments

Inoue does not disclose that the filename is “content.” The keyword is an arbitrary word vocalized by the user that is used to name a photograph.

Ground 5, Claim 6 (“storing content of the file as filenames”)

- Inoue discloses naming files using ***voice-input, arbitrary*** keywords, not content of the file

“In the voice recognition processing, one or more **keywords** (words) can be extracted **from the input voice data** using a word spotting voice recognition technology.”

Ex. 1012 (Inoue) at ¶101

“[T]he voice information setting section 707 ... sets one of the **keywords as the title (a filename)** of the image data.”

Ex. 1012 (Inoue) at ¶104