other such computing device). A non-limiting embodiment of a client computing system **500** configured to implement an authoring environment is described herein with reference to FIG. **5**.

**[0031]** In FIG. **5**, the exemplary computing system **500** for implementing the principles of the disclosure includes a client computing device, such as client computing device **510**. In a basic configuration, the client computing device **510** typically includes at least one processing unit **515** for executing applications and programs stored in system memory **520**. Depending on the exact configuration and type of computing device **510**, the system memory **520** may include, but is not limited to, RAM, ROM, EEPROM, flash memory, CD-ROM, digital versatile disks (DVD) or other optical storage devices, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or other memory technology.

**[0032]** System memory **520** typically stores an operating system **522**, such as the WINDOWS® operating systems from Microsoft Corporation of Redmond, Washington, suitable for controlling the operation of the computing device **510**. System memory **520** also may include a document cache **526** in which a client copy **527** of a document can be stored. Metadata **529** of the document also can be stored within the client cache **526**.

**[0033]** The system memory **520** also may store one or more software applications, such as authoring applications **524** for creating and editing documents. One non-limiting example of an authoring application **524** suitable for authoring documents in accordance with the principles of the present disclosure is Word word processing software from Microsoft Corporation. Other non-limiting examples of authoring applications include POWERPOINT® presentation software and VISIO® drawing and diagraming software, both also from Microsoft Corporation. Other software applications can also be used.

**[0034]** Computing device **510** also may have input device (s) **530**, such as a keyboard, mouse, pen, voice input device, touch input device, etc., for entering and manipulating data. Output device(s) **535**, such as a display screen, speakers, printer, etc., also may be included. These output devices **535** are well known in the art and need not be discussed at length herein.

**[0035]** The computing device **510** also may contain communication connections **540** that allow the device **510** to communicate with other computing devices, for example, the storage device **420** of FIG. **4**, over a network in a distributed computing environment (e.g., an intranet or the Internet). By way of example, and not limitation, communication device media **540** includes wired media such as a wired network or direct-wired connection, and wireless media, such as acoustic, RF, infrared and other wireless media.

**[0036]** Turning now to FIG. **6** is a flow chart setting forth the general stages involved in a method **600** consistent with an embodiment of the disclosure for retrieving presence metadata. Method **600** may be implemented using a computing device **510** as described in above with respect to FIG. **5**. Ways to implement the stages of method **600** will be described in greater detail below. Method **600** may begin at starting block **605** and proceed to stage **610** where computing device **510** may receive metadata **529**. For example, a client using computing device **510** (e.g. a client computer) may open a document (e.g. **150**). Opening the document may cause computing device **510** to receive short-term check out metadata for the document.

**[0037]** From stage **610**, where computing device **510** received metadata **529**, method **600** may advance to stage **620** where computing device **510** may add a transition ID to the metadata **529**.

**[0038]** Once computing device **510** adds the transition ID to metadata **529** in stage **620**, method **600** may continue to subroutine **630** where computing device **510** may write the transition ID to a transition table stored in cache **526**. For example, writing the transition ID to metadata **529** may indicate more than one client is editing the document (i.e. switching from a single-client mode to a multi-client mode) as show in FIG. **7**.

**[0039]** In one example, the transition ID is a unique number, such as a Globally Unique Identifier (GUID—i.e., a pseudo-random 128-bit number). The transition table is a list of the transition IDs. The transition table can also include an expiration date for each transition ID. The expiration date can be used to clear a transition ID when a client forces creation of a transition ID and thereupon abandon editing of the document without notification. Other configurations are possible.

**[0040]** From subroutine stage **630**, where computing device **510** may write the transition ID to a transition table stored in cache **526**, method **600** may advance to stage **640** where computing device **510** may ping storage device **120** (e.g. a server) to determine if the transition ID is in cache **526**.

**[0041]** When the transition ID is not in the cache, computing device **510** may ping the server at regular intervals, and save the document to the server a plurality of times without incurring any reads/writes to a database that stores presence information.

**[0042]** When the transition ID is in the cache, computing device **510** may ping the server to collect a lock table from a database to identify a new client, and in a separate web service request, submitting the client's lock information. Computing device **510** may then delete the transition ID from the transition table.

**[0043]** Computing device **510** may also submit a client's lock information. Submitting the client's lock information may include pinging the server. For example, the computing device can submit "am I alone" pings to the server, wherein the "am I alone" pings do not carry lock information. An "am I alone" ping is simply a ping to the server or other storage device wherein computing device **510** is requesting information to determine if the document is being edit by another client or if another client had begun editing the document after the client began editing the document.

**[0044]** Every time each client downloads presence metadata, the number of clients in an editing session and the last time each client updated their presence information (e.g., transition ID, transition table, database, etc.) may be checked (among other things), at regular intervals. If the presence information has not been updated by any client in a configurable window of time, it is considered a violent exit from the session. Any client who first observes this exit can delete the client from the presence information. Further, the delete operation should be a graceful merge so that any other client who also made the same observation can re-request the omissions which results in a no-op in the server.

**[0045]** If the presence information reflects only one client (which should be the examining client, i.e., computing device **510**), the following actions may be done to minimize problems associated with the timing of the change of state of the document: 1) clear-up the presence table so it reflects the state