



US 20250267115A1

(19) **United States**

(12) **Patent Application Publication**  
**Evans et al.**

(10) **Pub. No.: US 2025/0267115 A1**

(43) **Pub. Date: Aug. 21, 2025**

(54) **GESTURE-BASED COMMUNITY MODERATION**

**Publication Classification**

(71) Applicant: **Khoros, LLC**, Austin, TX (US)

(72) Inventors: **David Evans**, Austin, TX (US);  
**Vivekanand Venkatesh Kini**,  
Bangalore (IN); **Colin Cooper**, Austin,  
TX (US); **Vigneshwaran**  
**Dharmalingam**, Bangalore (IN);  
**Tanmay Sinha**, Bangalore (IN)

(51) **Int. Cl.**

**H04L 51/212** (2022.01)

**H04L 51/214** (2022.01)

**H04L 51/42** (2022.01)

**H04L 51/52** (2022.01)

(52) **U.S. Cl.**

CPC ..... **H04L 51/212** (2022.05); **H04L 51/214**

(2022.05); **H04L 51/42** (2022.05); **H04L 51/52**

(2022.05)

(73) Assignee: **Khoros, LLC**, Austin, TX (US)

(21) Appl. No.: **19/063,150**

(22) Filed: **Feb. 25, 2025**

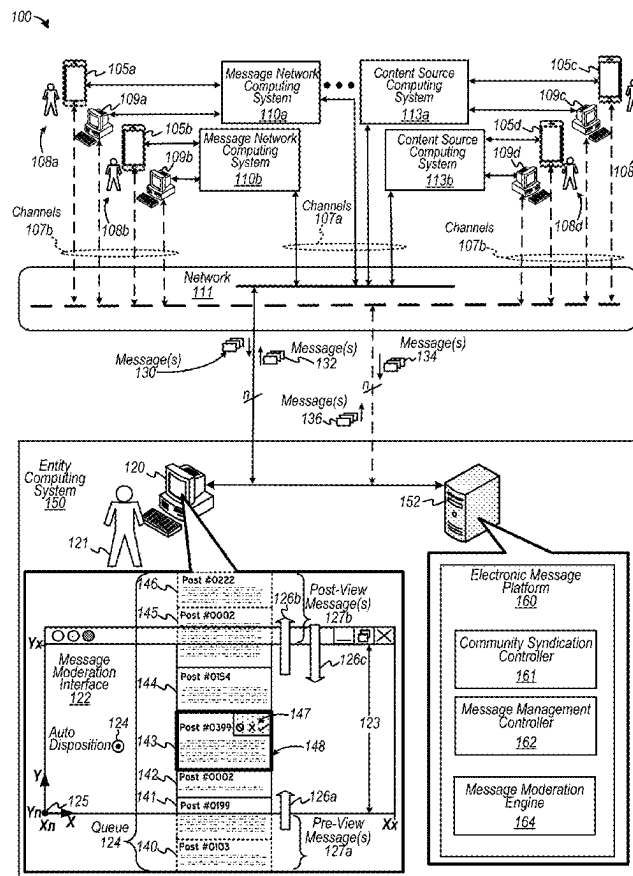
**Related U.S. Application Data**

(63) Continuation of application No. 18/233,618, filed on Aug. 14, 2023, now Pat. No. 12,238,056, which is a continuation of application No. 17/855,320, filed on Jun. 30, 2022, now Pat. No. 11,729,125, which is a continuation of application No. 17/365,222, filed on Jul. 1, 2021, now Pat. No. 11,438,289, which is a continuation of application No. 17/026,152, filed on Sep. 18, 2020, now Pat. No. 11,128,589.

(57)

**ABSTRACT**

Various techniques for gesture-based community moderation are described, including activating a moderator application at a computing platform hosting a syndication of electronic messages, receiving a gesture configured to determine whether to present an electronic message at an interface, characterizing content of an electronic message to determine one or more attributes, analyzing characterized attributes of an electronic message to identify classification values representative of message attribute data, arranging the electronic messages relative to other electronic messages to form a queue of electronic messages as a function of message attribute data, receiving one or more gestures configured to cause presentation of a first electronic message, filtering the queue of the electronic messages with which to facilitate an action automatically of a first electronic message, and performing an automatic action in accordance with the first electronic message.



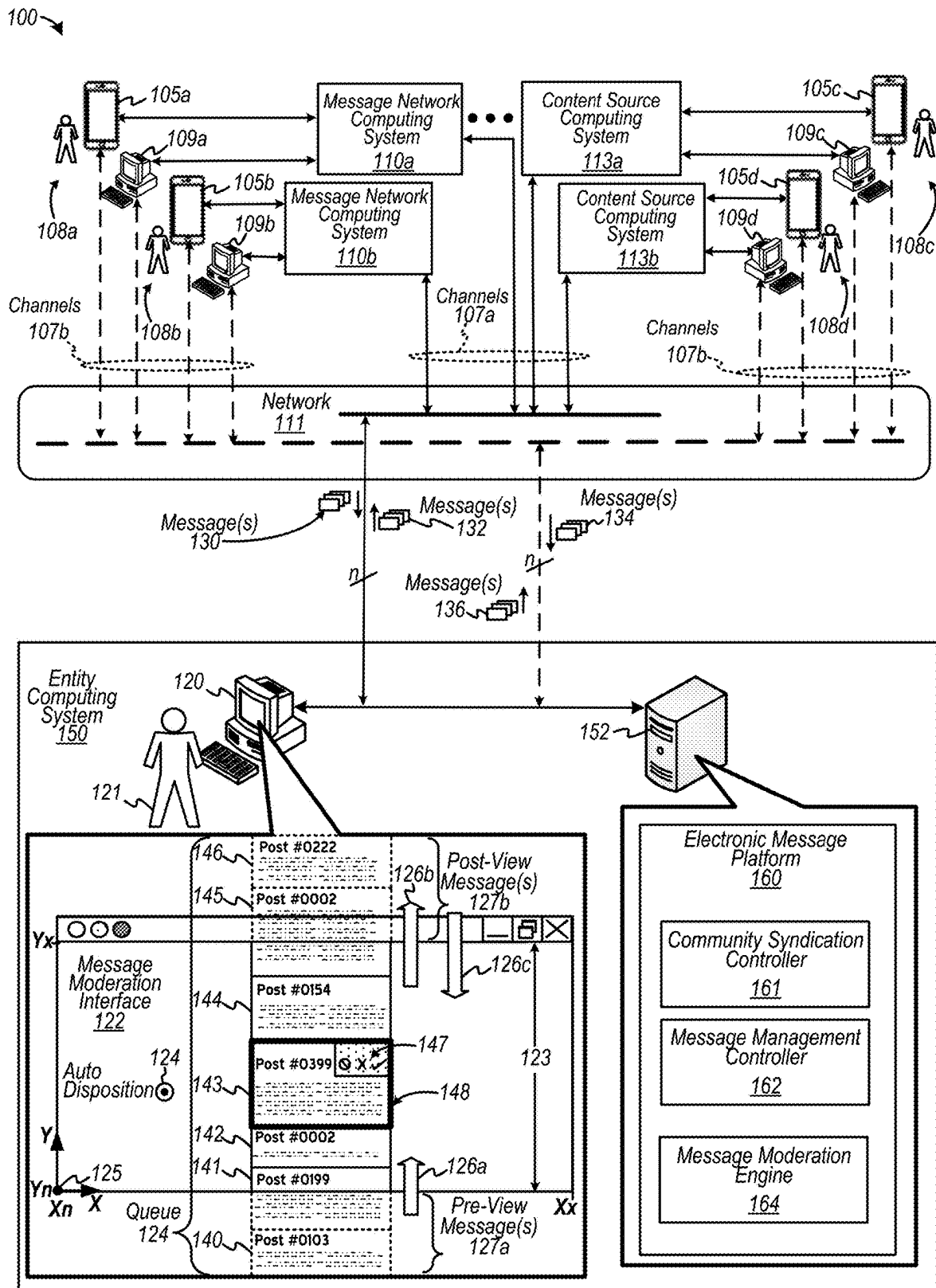


FIG. 1

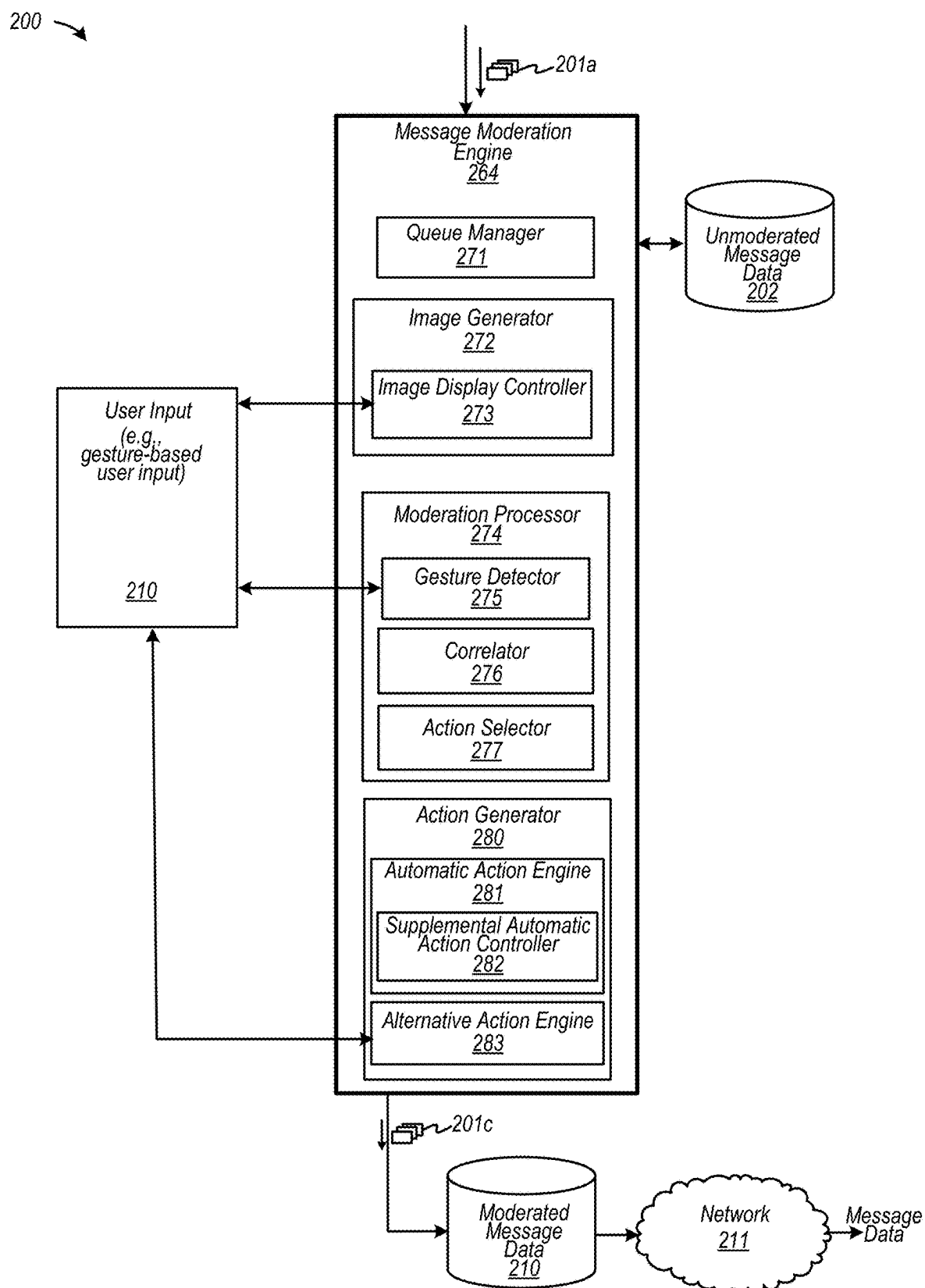
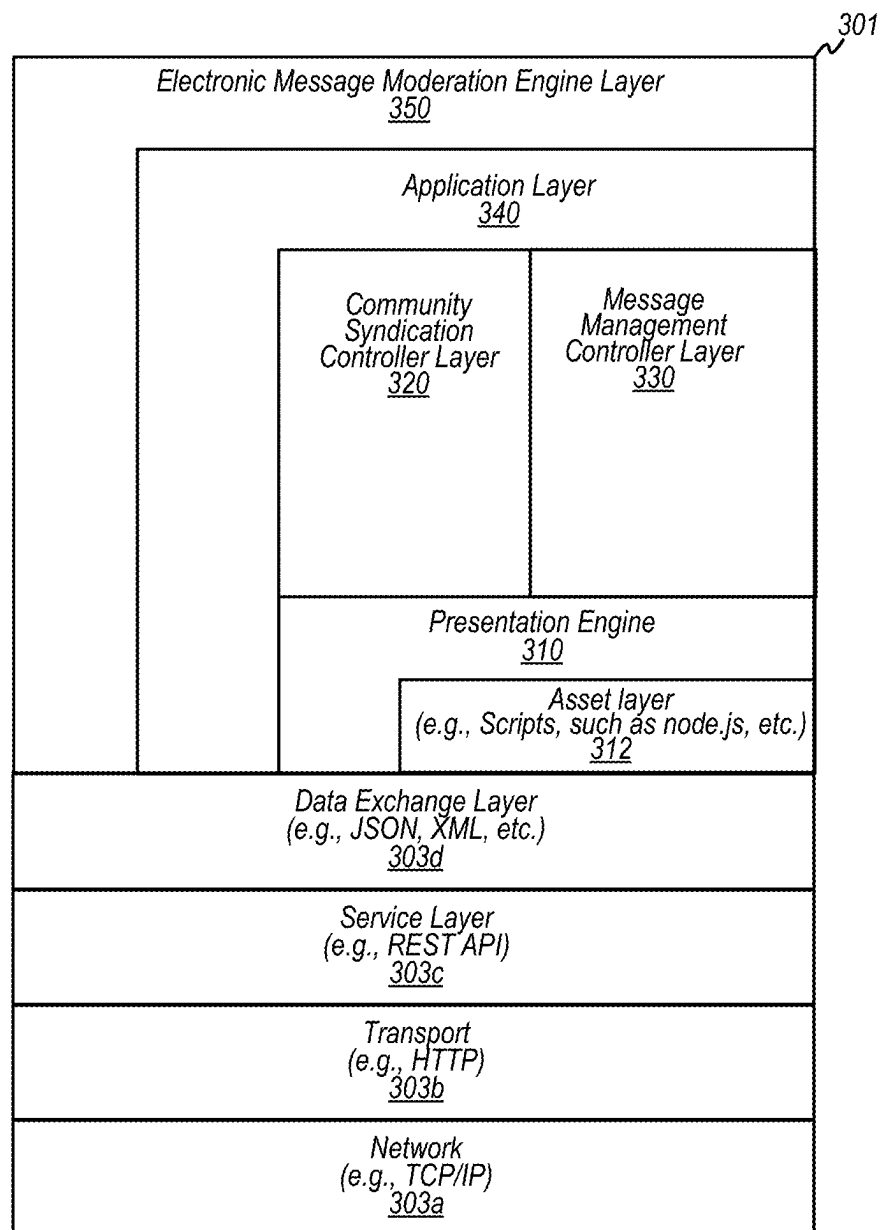


FIG. 2

300 →



**FIG. 3**

400 →

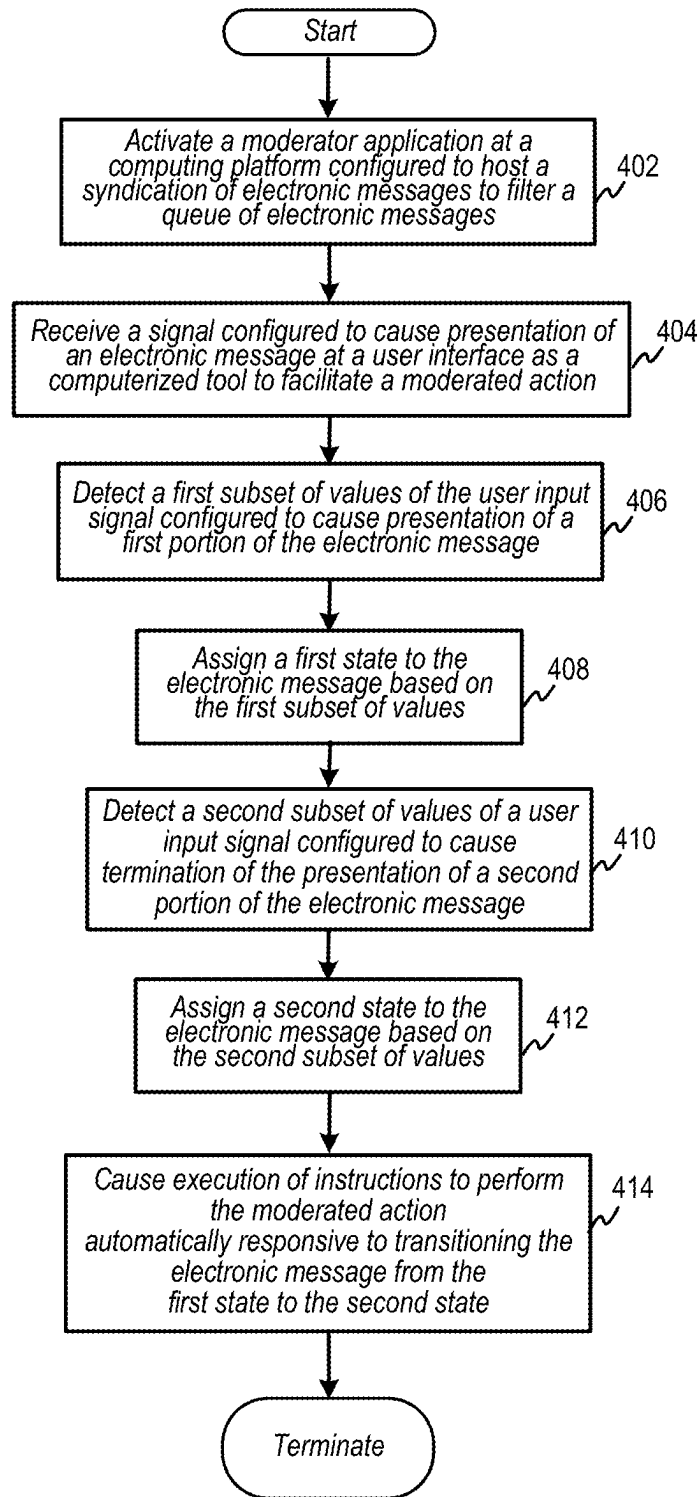


FIG. 4

500 →

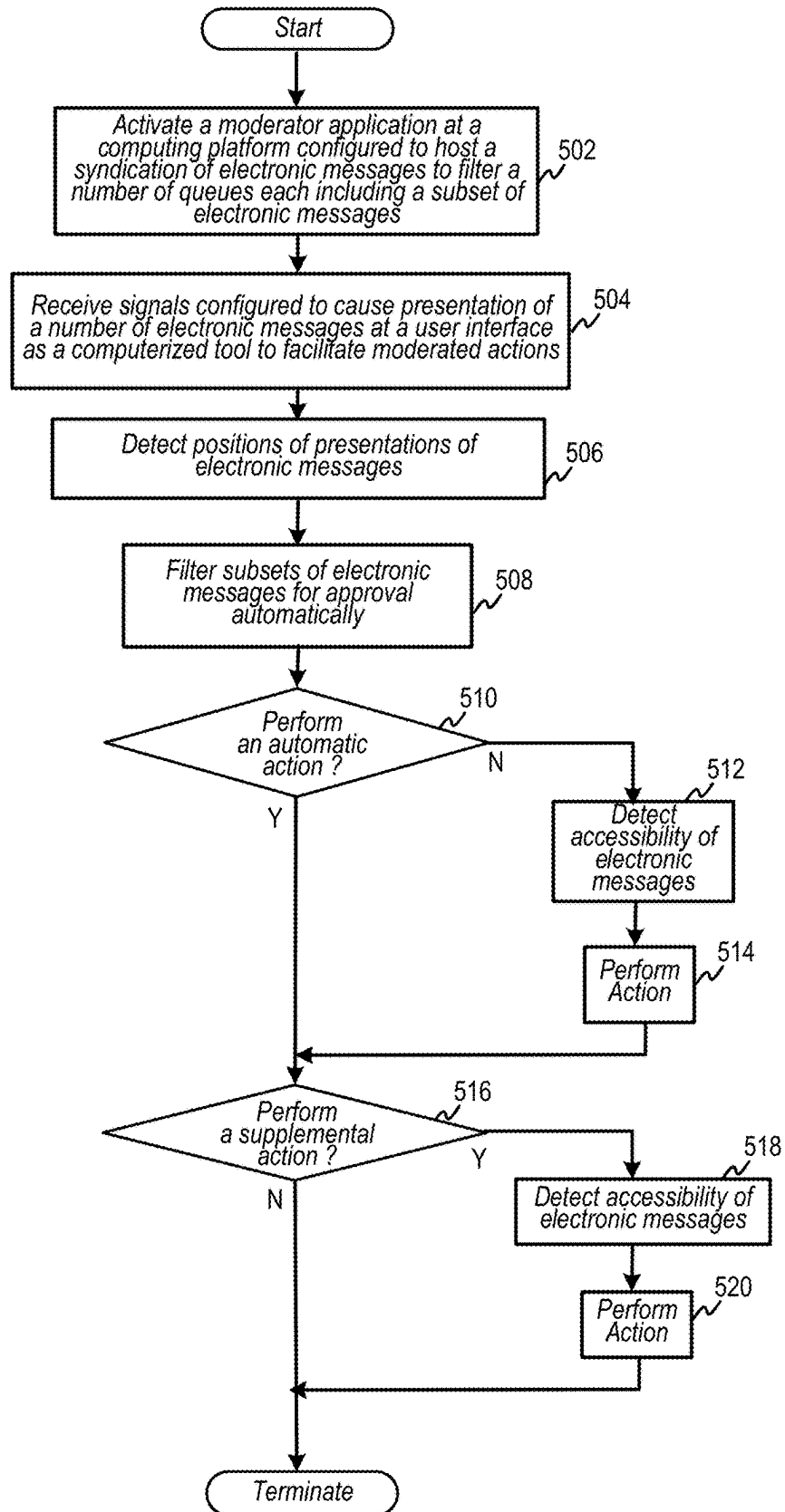


FIG. 5

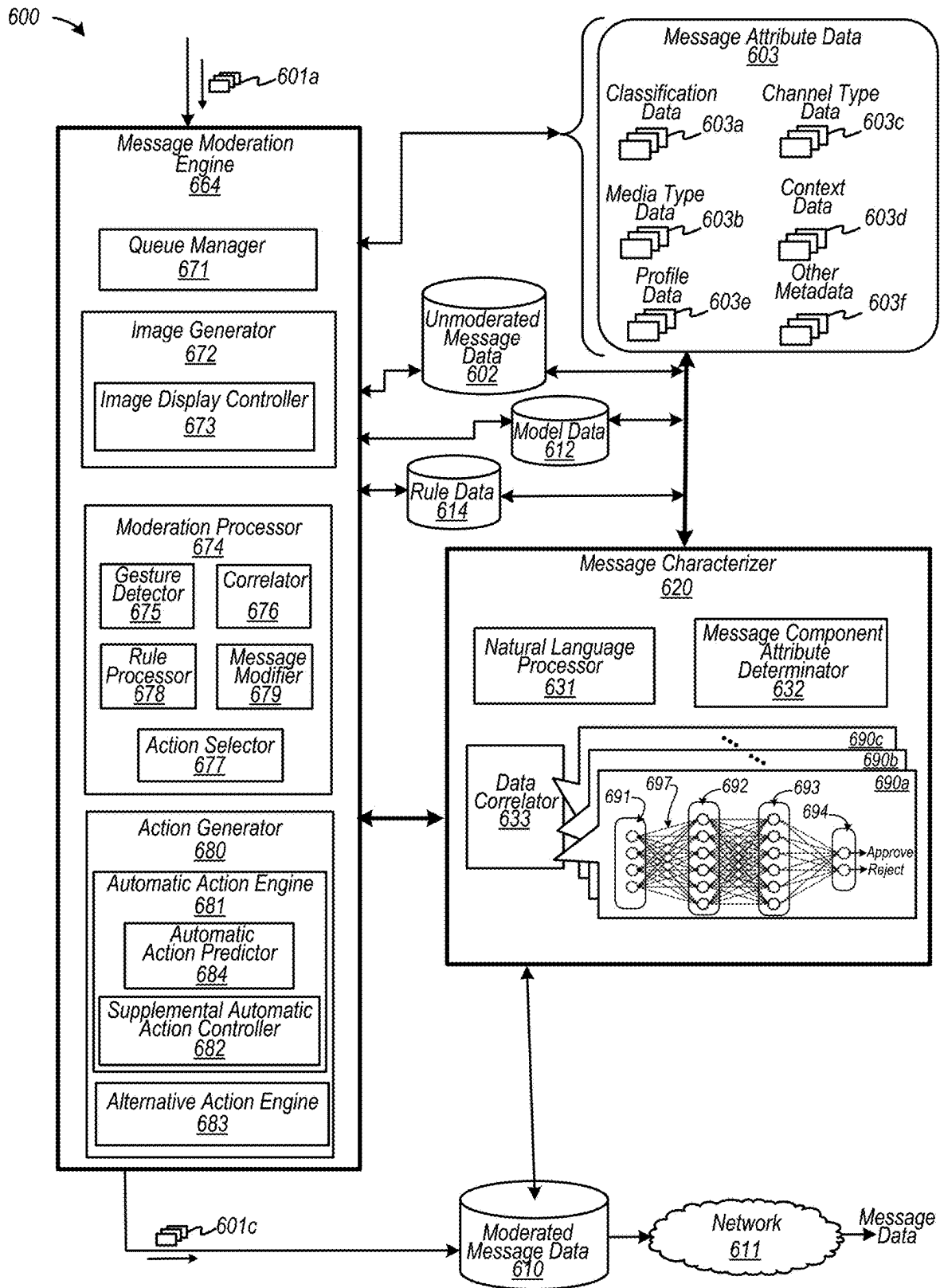


FIG. 6

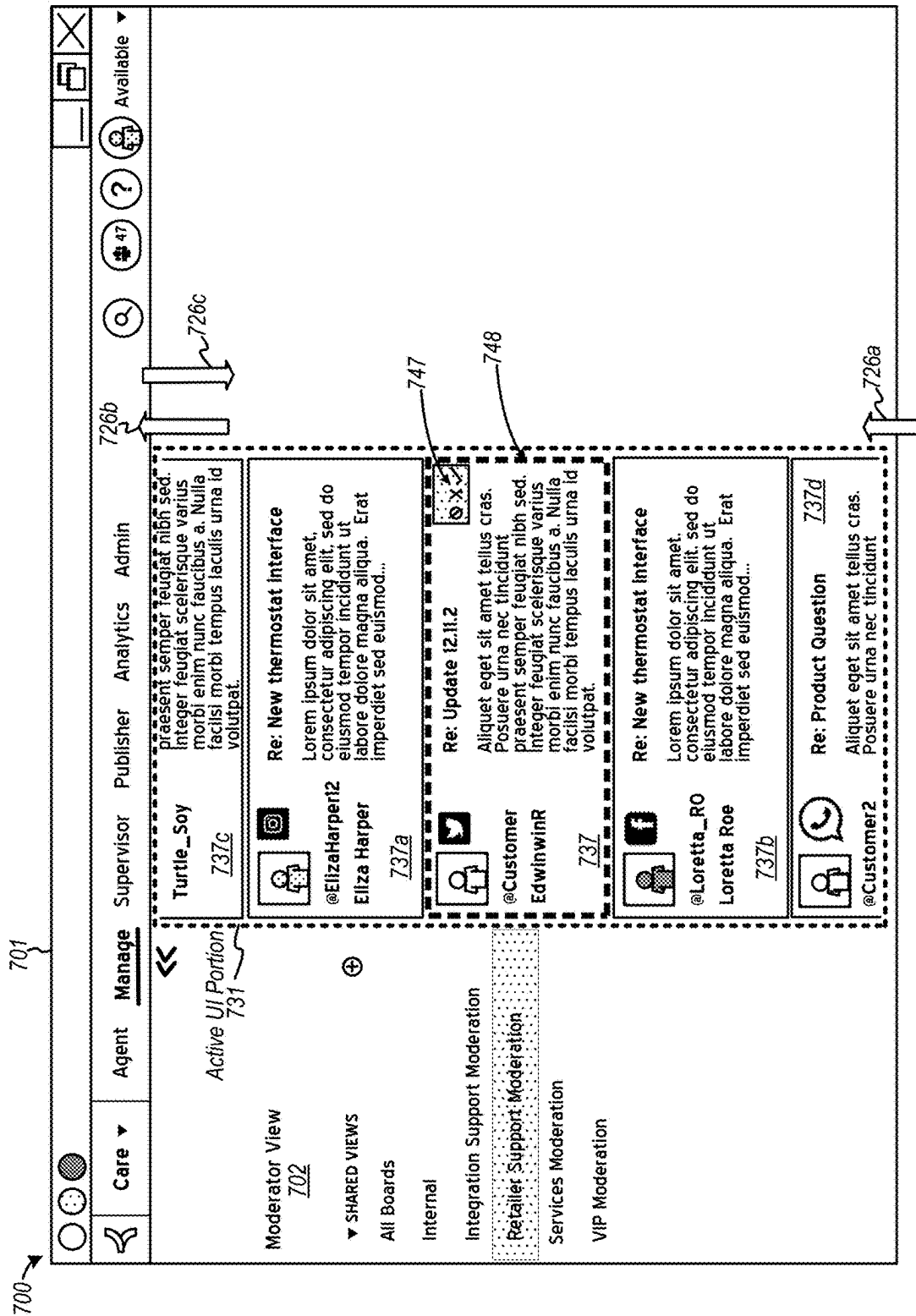
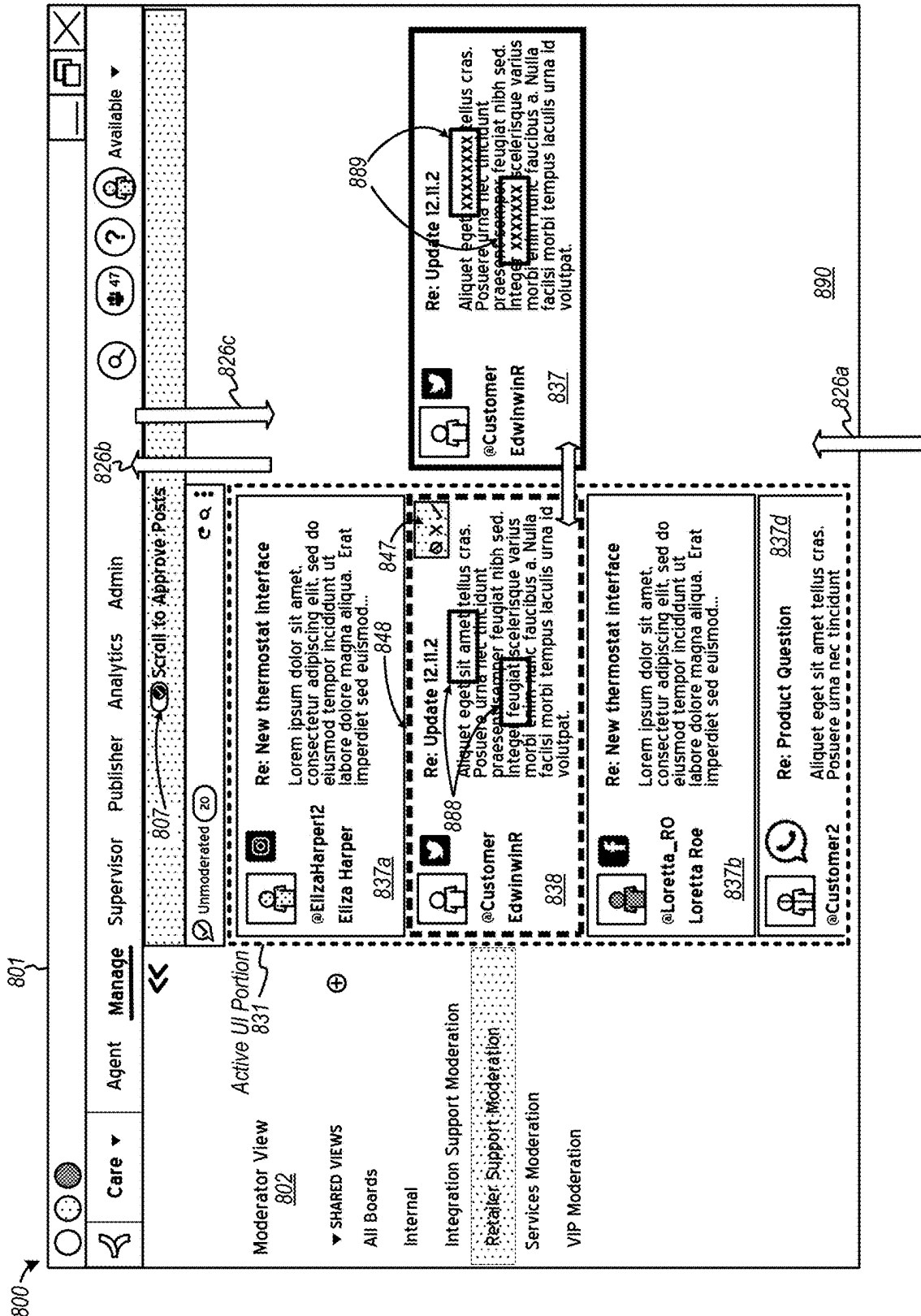
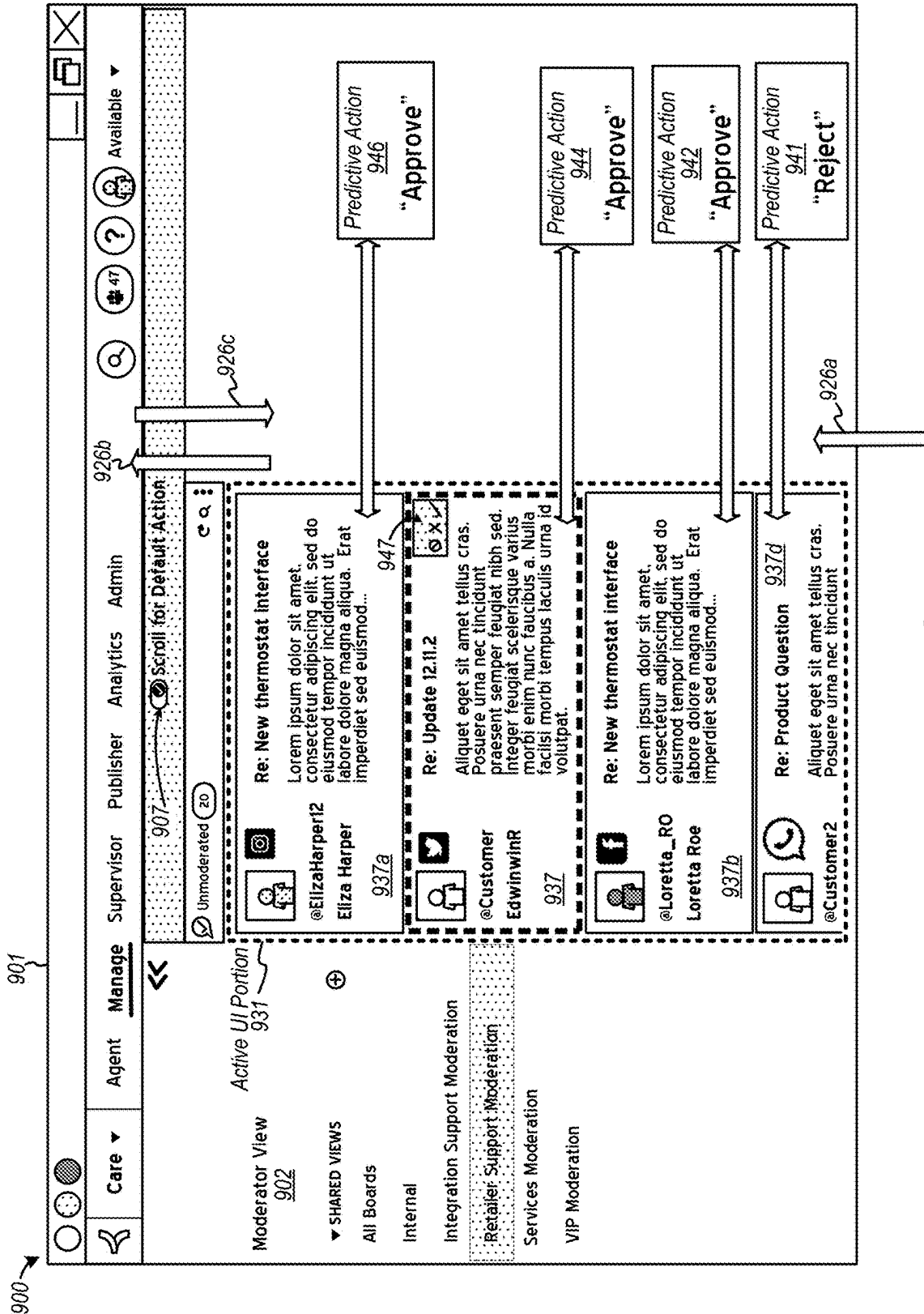


FIG. 7





**8.  
G.  
E.**



1000 →

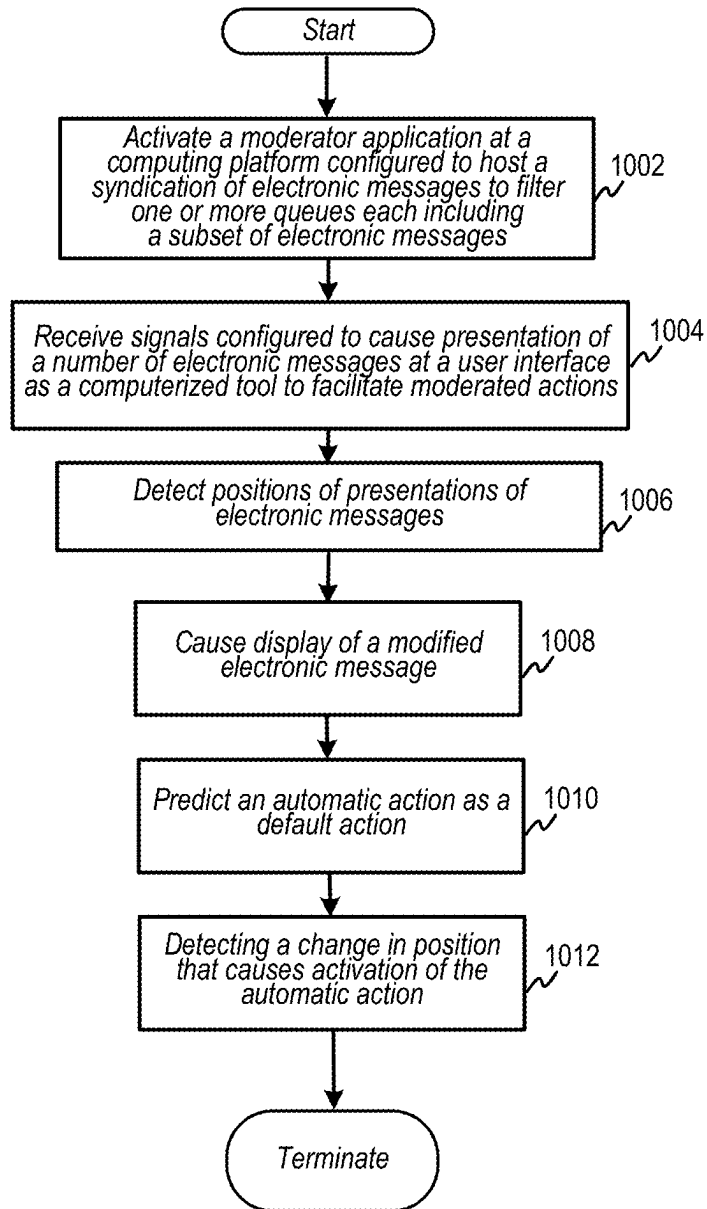


FIG. 10

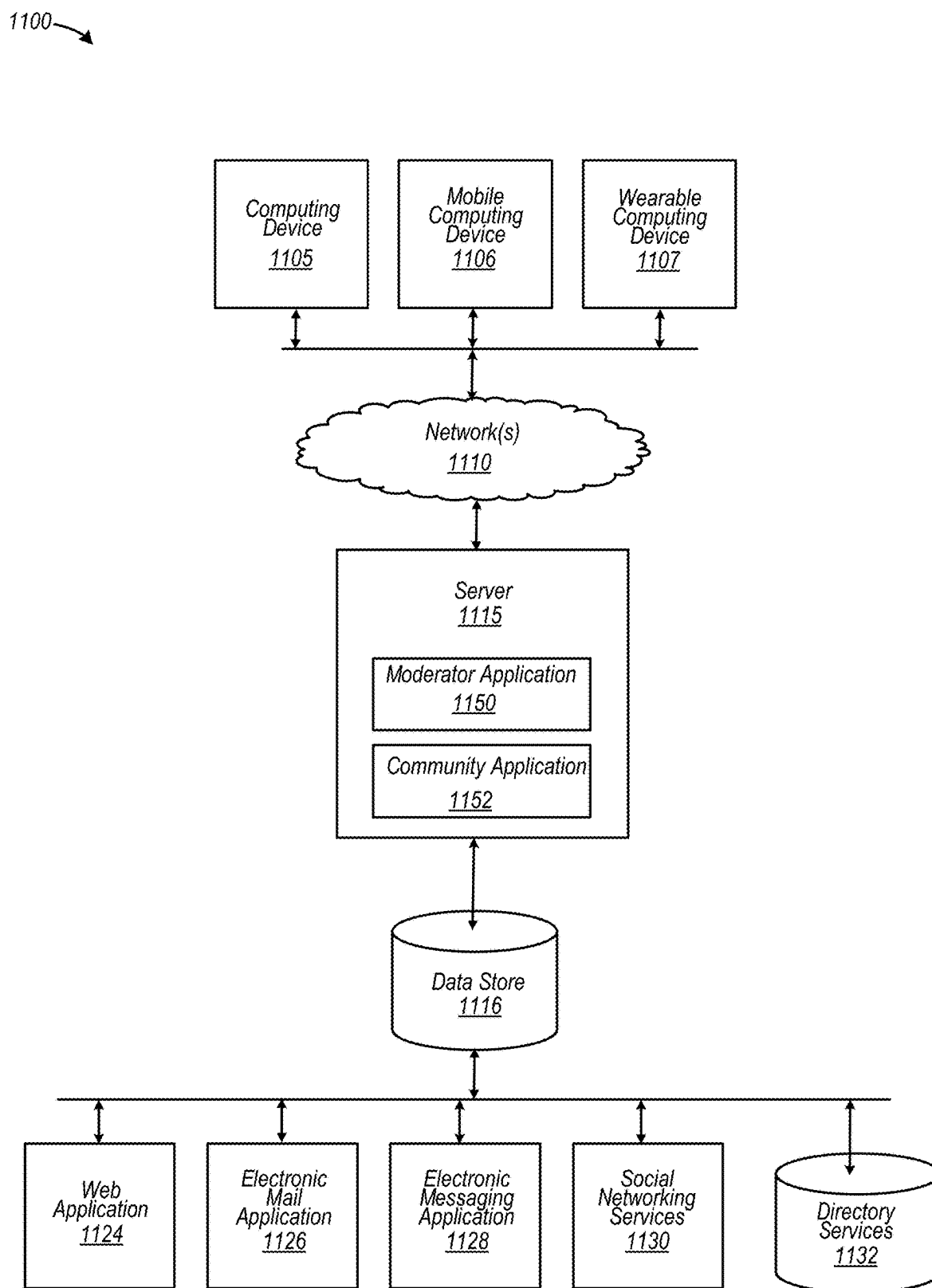


FIG. 11

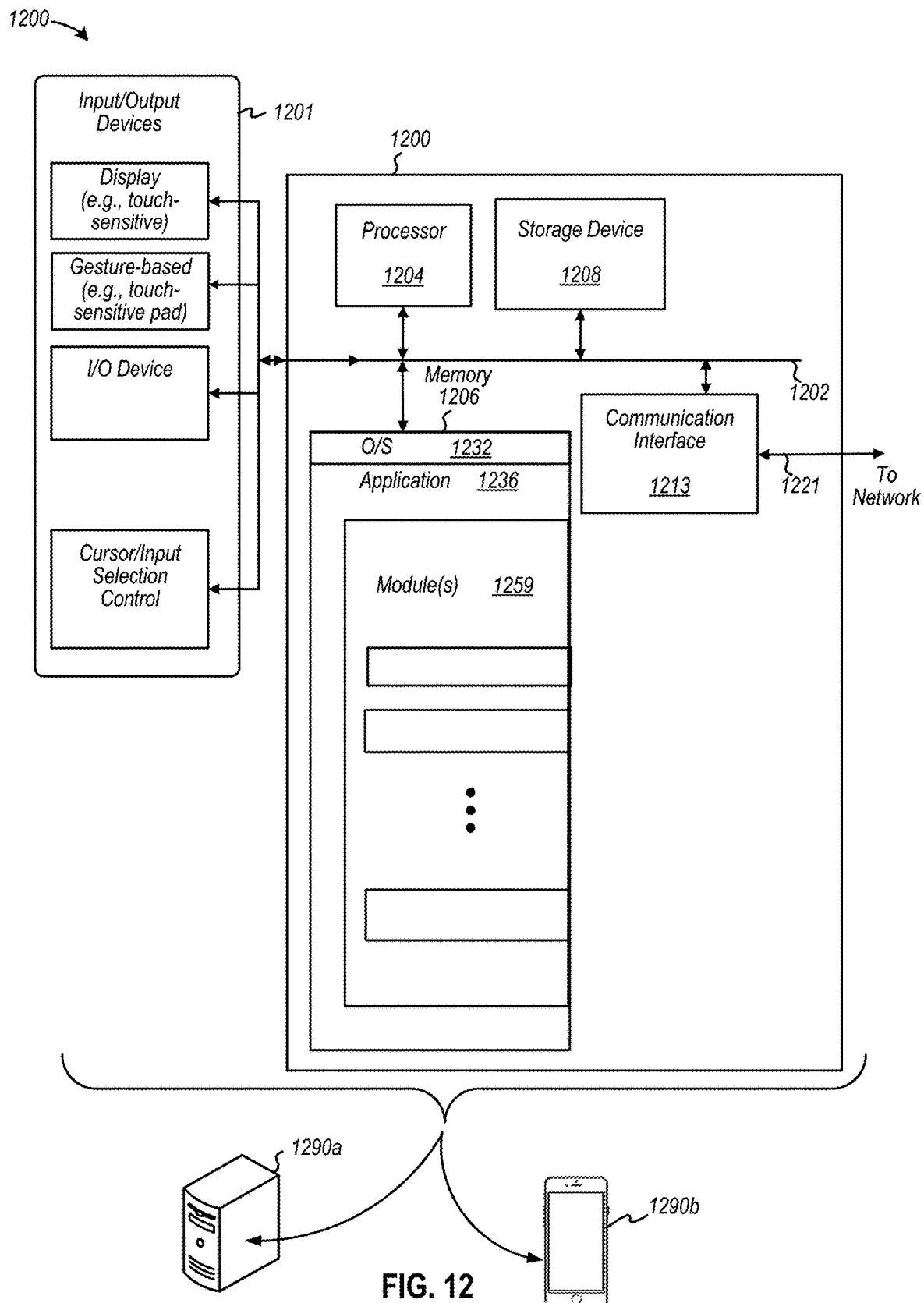


FIG. 12

## GESTURE-BASED COMMUNITY MODERATION

### CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This nonprovisional patent application is a continuation application of copending U.S. patent application Ser. No. 18/233,618, filed Aug. 14, 2023 and titled, “GESTURE-BASED COMMUNITY MODERATION,” U.S. patent application Ser. No. 18/233,618 is a continuation application of U.S. patent application Ser. No. 17/855,320, filed Jun. 30, 2022, now U.S. Pat. No. 11,729,125 and titled, “GESTURE-BASED COMMUNITY MODERATION,” U.S. patent application Ser. No. 17/855,320 is a continuation application of U.S. patent application Ser. No. 17/365,222, filed Jul. 1, 2021, now U.S. Pat. No. 11,438,289 and titled, “GESTURE-BASED COMMUNITY MODERATION,” U.S. patent application Ser. No. 17/365,222 is a continuation application of U.S. patent application Ser. No. 17/026,152, filed Sep. 18, 2020, now U.S. Pat. No. 11,128,589 and titled, “GESTURE-BASED COMMUNITY MODERATION,” all of which are herein incorporated by reference in their entirety for all purposes.

### FIELD

[0002] Various embodiments relate generally to data science and data analysis, computer software and systems, and control systems to provide a platform to facilitate implementation of an interface as a computerized tool, among other things, and, more specifically, to a computing and data platform that implements logic to facilitate moderation of electronic messages, postings, content, etc., via implementation of a moderator application configured to, for example, perform one or more actions automatically.

### BACKGROUND

[0003] Advances in computing hardware and software have fueled exponential growth in delivery of vast amounts of information due to increased improvements in computational and networking technologies. Also, advances in conventional data network technologies provide an ability to exchange increasing amounts of generated data via various electronic messaging platforms. Thus, improvements in computing hardware, software, network services, and storage have bolstered growth of Internet-based messaging applications, such as social networking platforms and applications, especially in an area of exchanging digital information concerning products and services to facilitate customer care operations, as well as online communities. For example, various organizations and corporations (e.g., retailer sellers) may exchange information through any number of electronic messaging networks, including social media networks. Such entities aim to respond quickly and efficiently to messages from customers (and potential customers) to manage brand loyalty and reputation, and to bolster customer engagement.

[0004] And since different customers prefer communicating over different communication channels (e.g., social media networked channels and communities of electronic messages) and various different data networks, traditional customer-relationship management (“CRM”) computing systems and processes are not well-suited to adapt to engage

and moderate communities of customers and associated computing devices at convenience of the customers “on-the-go.”

[0005] Social media networks and communities can be found in widespread use as there are many different types and categories often reaching as many as a billion or more users, electronic messages, or units of content, which include various types of content, such as pictures, text, video, audio, multimedia, or any combination thereof. Social media networks and communities are increasingly prolific and can range in scope, size, interest, and reach. Social media networks and communities, in some cases, can be integrated with websites, platforms, or other online properties for specific interests or purposes, which may be social, commercial, governmental, educational, academic, professional, technical, etc.

[0006] Various social media networks, websites, and communities can require communities of content to meet certain standards, terms, or conditions that are social, cultural, or legal in nature, which require compliance and, consequently, can often reduce or limit what content may be posted online. Generally, profane or hateful content, threatening content, perjurious commercial speech, and the like, may violate policies and terms of use of a community, which, if unmoderated, can result in social detriment, legal liability, breach of cultural values and mores (e.g., alongside laws, rules, and regulations), or other deleterious effects. In other cases, fraudulent product reviews may damage the online goodwill and perceived value of a company and its products.

[0007] To enforce compliance and avoid inappropriate content from being posted into a community of content and electronic messages, traditional social media networks and communities implement conventional social media networks, platforms, platform technologies, and software to monitor, moderate, and manage content and electronic message prior to posting. Conventionally, employees or other personnel (e.g., of various social media networks, platforms, platform providers, and software developers that provide products or services) may be required, as moderators, to read, review, or otherwise scrutinize content intended to be posted to one or more social media networks, or to online communities and groups.

[0008] Moderators may be required to review and act on hundreds, thousands, or perhaps hundreds of thousands of items, uploads, or other postings (e.g., electronic posts, messages, units of content, etc.) to ensure compliance with terms (e.g., terms of use), conditions, policies, laws, regulations, rules, and the like. However, conventional moderation techniques and software, including those integrated with existing social media management platforms are generally inefficient, expensive, time-consuming, and usually cumbersome to maintain given the increasing amounts of posting submissions.

[0009] Various drawbacks to traditional techniques to moderation using known computing devices and software can often lead to fatigue, resulting in lapses in focus, inadvertent posts, errors, and the like, which may create significant cost and effort to correct. Erroneously moderated content, which may result in propagation of unmoderated content, can also detrimentally affect cohesiveness, trustworthiness, reliability, and reputations of online communities. Further, suboptimal moderation techniques using computing devices and moderator software may increase risk of civil and criminal penalties for unmoderated speech or

content that is regulated by law (e.g., violations of Americans with Disabilities Act, or “ADA”). In addition, incorrect or suboptimal content posted into a community may cause customer frustration and dissatisfaction, which can typically be projected negatively unto a corporation and its brand.

[0010] Thus, what is needed is a solution for facilitating techniques that optimize computer utilization and performance associated with moderating content and/or electronic messages, without the limitations of conventional techniques.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Various embodiments or examples (“examples”) of the invention are disclosed in the following detailed description and the accompanying drawings:

[0012] FIG. 1 is a diagram depicting an electronic message platform to facilitate moderation of subsets of electronic content, according to some embodiments;

[0013] FIG. 2 depicts an example of a message moderation engine, according to various examples;

[0014] FIG. 3 illustrates an exemplary layered architecture for implementing a moderator application, according to some examples;

[0015] FIG. 4 is a flow diagram as an example of moderating an electronic message, according to some embodiments;

[0016] FIG. 5 is a flow diagram as another example of moderating a number of electronic message, according to some embodiments;

[0017] FIG. 6 depicts another example of a message moderation engine, according to various examples;

[0018] FIG. 7 depicts an example of a user interface configured to moderate electronic messages and posts, according to some examples;

[0019] FIG. 8 depicts another example of a user interface configured to moderate electronic messages and posts using proposed modified electronic messages, according to various examples;

[0020] FIG. 9 depicts yet another example of a user interface configured to moderate electronic messages and posts using predictive default actions, according to various examples;

[0021] FIG. 10 is a flow diagram as another example of moderating a number of electronic message using either proposed modified message content or predicted automatic default actions, or both, according to some embodiments;

[0022] FIG. 11 depicts an example of a system architecture to provide a computing platform to host a syndication of electronic messages and posts for moderation, according to an example; and

[0023] FIG. 12 is a diagram depicting an example of an electronic message response platform configured to moderate content and/or electronic messages, according to some examples.

#### DETAILED DESCRIPTION

[0024] Various embodiments or examples may be implemented in numerous ways, including as a system, a process, an apparatus, a user interface, or a series of program instructions on a computer readable medium such as a computer readable storage medium or a computer network where the program instructions are sent over optical, electronic, or wireless communication links. In general, opera-

tions of disclosed processes may be performed in an arbitrary order, unless otherwise provided in the claims.

[0025] A detailed description of one or more examples is provided below along with accompanying figures. The detailed description is provided in connection with such examples, but is not limited to any particular example. The scope is limited only by the claims, and numerous alternatives, modifications, and equivalents thereof. Numerous specific details are set forth in the following description in order to provide a thorough understanding. These details are provided for the purpose of example and the described techniques may be practiced according to the claims without some or all of these specific details. For clarity, technical material that is known in the technical fields related to the examples has not been described in detail to avoid unnecessarily obscuring the description or providing unnecessary details that may be already known to those of ordinary skill in the art.

[0026] As used herein, “system” may refer to or include the description of a computer, network, or distributed computing system, topology, or architecture using various computing resources that are configured to provide computing features, functions, processes, elements, components, or parts, without any particular limitation as to the type, make, manufacturer, developer, provider, configuration, programming or formatting language, service, class, resource, specification, protocol, or other computing or network attributes. As used herein, “software” or “application” may also be used interchangeably or synonymously with, or refer to, a computer program, software, program, firmware, or any other term that may be used to describe, reference, or refer to a logical set of instructions that, when executed, performs a function or set of functions within a computing system or machine, regardless of whether physical, logical, or virtual and without restriction or limitation to any particular implementation, design, configuration, instance, or state. Further, “platform” may refer to any type of computer hardware (hereafter “hardware”) or software, or any combination thereof, that may use one or more local, remote, distributed, networked, or computing cloud (hereafter “cloud”)-based computing resources (e.g., computers, clients, servers, tablets, notebooks, smart phones, cell phones, mobile computing platforms or tablets, and the like) to provide an application, operating system, or other computing environment, such as those described herein, without restriction or limitation to any particular implementation, design, configuration, instance, or state. Distributed resources such as cloud computing networks (also referred to interchangeably as “computing clouds,” “storage clouds,” “cloud networks,” or, simply, “clouds,” without restriction or limitation to any particular implementation, design, configuration, instance, or state) may be used for processing and/or storage of varying quantities, types, structures, and formats of data, without restriction or limitation to any particular implementation, design, or configuration.

[0027] As used herein, data may be stored in various types of data structures including, but not limited to databases, data repositories, data warehouses, data stores, or other data structures configured to store data in various computer programming languages and formats in accordance with various types of structured and unstructured database schemas such as SQL, MySQL, NoSQL, DynamoDB™, etc. Also applicable are computer programming languages and formats similar or equivalent to those developed by data

facility and computing providers such as Amazon® Web Services, Inc. of Seattle, Washington, FMP, Oracle®, Salesforce.com, Inc., or others, without limitation or restriction to any particular instance or implementation. DynamoDB™, Amazon Elasticsearch Service, Amazon Kinesis Data Streams (“KDS”)™, Amazon Kinesis Data Analytics, and the like, are examples of suitable technologies provide by Amazon Web Services (“AWS”).

**[0028]** Further, references to databases, data structures, or any type of data storage facility may include any embodiment as a local, remote, distributed, networked, cloud-based, or combined implementation thereof. For example, social networks and social media (e.g., “social media”) using different types of devices may generate (i.e., in the form of posts (which is to be distinguished from a POST request or call over HTTP) on social networks and social media) data in different forms, formats, layouts, data transfer protocols, and data storage schema for presentation on different types of devices that use, modify, or store data for purposes such as electronic messaging, audio or video rendering, content sharing, or like purposes. Data may be generated in various formats such as text, audio, video (including three dimensional, augmented reality (“AR”), and virtual reality (“VR”), or others, without limitation, for use on social networks, social media, and social applications (e.g., “social media”) such as Twitter® of San Francisco, California, Snapchat® as developed by Snap® of Venice, California, Messenger as developed by Facebook®, WhatsApp®, or Instagram® of Menlo Park, California, Pinterest® of San Francisco, California, LinkedIn® of Mountain View, California, as well as community-based networks, such as Khoros® Community as provided by Khoros, LLC of Austin, Texas, and others, without limitation or restriction.

**[0029]** In some examples, data may be formatted and transmitted (i.e., transferred over one or more data communication protocols) between computing resources using various types of data communication and transfer protocols such as Hypertext Transfer Protocol (“HTTP”), Transmission Control Protocol (“TCP”)/Internet Protocol (“IP”), Internet Relay Chat (“IRC”), SMS, text messaging, instant messaging (“IM”), File Transfer Protocol (“FTP”), or others, without limitation. As described herein, disclosed processes implemented as software may be programmed using Java®, JavaScript®, Scala, Python™, XML, HTML, and other data formats and programs, without limitation. Disclosed processes herein may also implement software such as Streaming SQL applications, browser applications (e.g., Firefox™) and/or web applications, among others. In some example, a browser application may implement a JavaScript framework, such as Ember.js, Meteor.js, ExtJS, AngularJS, and the like. References to various layers of an application architecture (e.g., application layer or data layer) may refer to a stacked layer application architecture such as the Open Systems Interconnect (“OSI”) model or others.

**[0030]** In some examples, systems, software, platforms, and computing clouds, or any combination thereof, may be implemented to facilitate online “communities” of subsets of units of content, postings, electronic messages, and the like. In some cases, units of content, electronic postings, electronic messages, and the like may originate at social networks, social media, and social applications, or any other source of content.

**[0031]** FIG. 1 is a diagram depicting an electronic message platform to facilitate moderation of subsets of elec-

tronic content, according to some embodiments. Diagram **100** depicts an example of an entity computing system **150** including an electronic message platform **160** that may be configured to, among other things, facilitate moderation of electronic messages, postings, content, etc., via implementation of a moderator application and/or computing device configured to, for example, perform one or more actions automatically. In various examples, one or more inbound electronic messages may be disposed (e.g., as unmoderated electronic messages) in a queue so that electronic message platform **160** may facilitate moderation or filtering of the inbound electronic messages for posting into an online community, such as messages **130** and **134**. In some examples, moderation of electronic messages may be based on a user input that may be configured to perform multiple actions regarding each electronic message in a queue. In some examples, a user input to perform multiple actions may be implemented using a common user input, which may include a common gesture. Hence, a gesture, as a user input into a gesture-based interface, may facilitate moderation of electronic messages, at least in some examples. According to various examples, an electronic message may refer to an electronic post, content (e.g., or portion thereof, such as a unit of content), and the like, any of which may originate in various different formats and may be adapted for integration into an online community of, for example, categorized or classified content as presented in one or more webpages or any other electronic media. An electronic message or post may also refer to, for example, data representing an article, a comment, a reply submitted to an online community, or the like.

**[0032]** Diagram **100** depicts an entity computing system **150** including a user interface **120** and a computing device **152** (e.g., one or more servers, including one or more processors and/or memory devices), both of which may be configured to moderate electronic messages and implement any number of actions to facilitate the moderation of such messages based on logic disposed in electronic message platform **160**. As shown, computing device **152** may be configured, in at least one example, to provide one or more software functionalities and/or hardware structures to implement a community syndication controller **161**, a message management controller **162**, and a message moderation engine **164**.

**[0033]** Community syndication controller **161** may be configured to host an online community to facilitate an electronic exchange of information and data among a group of users with related interests, goals, questions, problems, suggestions, experiences, etc., regarding one or more products, one or more services, or any other one or more topics or subject matter-related issues, or the like. Further, community syndication controller **161** may be configured to interact electronically with message moderation engine **164**, which may be configured to moderate or filter (e.g., for approval) exchanges or postings of electronic messages in a moderated online community regardless of data formats (e.g., as a blog, a website, an email, a text message, or the like). Message management controller **162** may include logic configured to manage electronic interactions and messages among an online community as well as any other sources of data (e.g., online chat sessions, electronic messages directed to an entity rather than a community, or the like). In at least one example, community syndication controller **161** may be implemented with at least some func-



tionality provided by an application configured to operate in accordance with Lithium Community technologies (formally of Lithium Technologies, LLC), Khoros® Communities of Khoros, LLC of Austin Texas, and “Atlas” Communities of Khoros, LLC of Austin Texas, among other online community configurations. Further, message management controller 162 may be implemented using at least some functionality provided by an application configured to operate in accordance with “Modern Chat”-related technologies and “Khoros Care”-related technologies, both of Khoros, LLC of Austin Texas, among other technologies.

[0034] In some examples, a subset of an electronic community (e.g., online community) may include any number of electronic messages or posts that may relate to each other by subject matter or any other classification. As an example, an online community may be subdivided based on whether content relates to a “forum” (e.g., content directed to resolving a problem), an “idea” (e.g., content directed to proposed suggestions related to any item, such as a product), a “frequently-asked question” (e.g., content directed to searchable solutions that are determined to be effective), an “expert” classification (e.g., directed to users or electronic accounts associated with expert-based content), a “knowledge base” of searchable solutions to user inquiries, and any other classification or categorization.

[0035] Electronic messages may originate at any computing devices 109a to 109d, which are respectively associated with users 108a to 108d. In the example shown, user 108a may be associated with one or more computing devices, such as mobile computing device 105a or any type of computing device 109a, user 108b may be associated with one or more computing devices, such as mobile computing device 105b or any type of computing device 109b, user 108c may be associated with one or more computing devices, such as mobile computing device 105c or any type of computing device 109c, and user 108d may be associated with one or more computing devices, such as mobile computing device 105d or any type of computing device 109d. Note that any number of mobile and other types of computing devices may be configured to transmit and/or receive messages and are not limited to those shown. Any of mobile computing devices 105a to 105d and any of computing devices 109a to 109d may be configured to generate electronic messages to, for example, initiate moderation of those messages for inclusion in one or more data arrangements (e.g., in data storage) that constitute or implement an online community of messages.

[0036] Any one or more of message network computing systems 110a and 110b (including one or more applications) may be configured to receive and transmit electronic messages, regardless of a context, to convey an inquiry, experience, observation, request for assistance (e.g., in relation to a product or service), or any other information with or among any number of users for any reason. Such messages and content may be directed to resolving a problem via an inquiry, to providing experienced advice or suggestions (e.g., as an expert), to provide observations as an idea to, for example, improve a product or a service, to request for assistance, or to exchange any information among users 108a to 108d, whereby electronic message platform 160 and/or community syndication controller 161 may be configured to host and moderate, for example, peer-to-peer exchanges of messages using message moderation engine 164. Similarly, or equivalently, one or more of message

network computing systems 110a and 110b may be configured to communicate electronic message content in any form in any digital media or channel 107a. Also, one or more computing systems 113a and 113b may be configured to communicate electronic message content in any form in any digital media or channel 107a. Also, electronic message platform 160, community syndication controller 161, and/or message moderation engine 164 may be configured to moderate electronic message content originating at computing systems 113a and 113b as well as message network computing systems 110a and 110b.

[0037] Note that in some examples, channels 107a may be publicly-accessible channels, whereas channels 107b may constitute secure, private, and/or and proprietary communication channels. As such, mobile computing devices 105a to 105d and computing devices 109a to 109d may be configured to submit electronic messages for posting in an online community via a secure data channel 107b.

[0038] In various examples, message network computing systems 110a and 110b may include any number of computing systems configured to propagate electronic messaging, including, but not limited to, computing systems including third party servers, such as third parties like Facebook™, Twitter™, LinkedIn™, Instagram™, Snapchat™, as well as other private or public social networks to provide social-media related informational data exchange services. Hence, message network computing systems 110a and 110b may include any social network computing system. Computing systems 113a and 113b (including one or more applications, such as text messaging applications) may be configured to provide any type of digital content, such as email, text messaging (e.g., via SMS messaging, Multimedia Messaging Service (“MMS”), WhatsApp™, WeChat™, Apple® Business Chat™, Instagram™ Direct Messenger, etc.), web pages (e.g., Facebook® websites and posts, Instagram® websites and posts, Twitter® websites and posts, etc.), audio (e.g., Spotify®-based content, etc.), video (e.g., YouTube™-based content, etc.), and any other content.

[0039] According to some examples, message network computing systems 110a and 110b may include applications or executable instructions configured to facilitate interactions (e.g., social interactions) amongst one or more persons, one or more subpopulations (e.g., private groups or public groups), or the public at-large. Examples of message network computing systems 110a and 110b may include the above-mentioned electronic accounts for Facebook™, Twitter™, LinkedIn™, Instagram™, and Snapchat™, as well as YouTube™, Pinterest™, Tumblr™, WhatsApp™ messaging, or any other platform, including Khoros® community, that may be configured to promote sharing of content, such as videos, audio, or images, as well as sharing ideas, thoughts, etc. in a socially-based environment, such as an online community moderated by implementing structures and functions of a message moderation engine 164 and/or any other component of electronic message platform 160 or entity computer system 150. According to some examples, content source computing systems 113a and 113b may include applications or executable instructions configured to promote an activity, such as a sports television network, a profession sports team, a news or media organization, a product producing or selling organization, and the like, or to promote sales or acquisition of goods or services. Content source computing systems 113a and 113b may implement websites, email, chat bots (e.g., “bots”), or any other digital

communication channels, and may further implement electronic accounts to convey information via message network computing systems **110a** and **110b**.

**[0040]** In some examples, structures and/or functions of message network computing systems **110a** and **110b** and content source computing systems **113a** and **113b** may be implemented to operate similarly or equivalently as each other. Any electronic message may include a “tweet” (e.g., a message via a Twitter™ computing system), a “post” (e.g., a message via a Facebook™ computing system), or any other type of social network-based messages, along with any related functionalities, such as forwarding a message (e.g., “retweeting” via Twitter™), sharing a message, associating an endorsement of another message (e.g., “liking” a message, such as a tweet™, or sharing a Facebook™ post, etc.), and any other interaction that may convey or otherwise may generate a “response” or electronic posts to an online community from one or more electronic accounts at relatively increased rates of transmissions or propagation to address concerns or statements that may otherwise affect a reputation of a brand. As such, message moderation engine **164** may be configured to moderate electronic posts to ensure compliance with policies, terms of use, legal regulations, and any other rule. According to various examples, an electronic message received via a network **111** can include any type of digital messaging that can be transmitted over any digital network (e.g., the Internet, etc.).

**[0041]** Entity computing system **150** is shown to include a computing device **120** and display configured to generate a user interface, such as a message moderation interface **122**. Entity computing system **150** may also include a server computing device **152**, which may include hardware and software, or a combination thereof, configured to implement an electronic message platform **160** configured to moderate a queue of electronic messages based on a user input that may be configured to perform multiple actions. In some examples, a user input to perform multiple actions may be a common user input, which may include a common gesture, such as a movement or interaction of one or more portions of a user **121** (e.g., a motion of a finger in one direction, and, optionally a reverse direction). Hence, a gesture, as a user input into a gesture-based interface, may facilitate moderation of electronic messages **130** and **134**, at least in some examples. In various examples, user **121** may be an agent acting in a role of a “moderator,” or as a user in any other function or role (e.g., a supervisory moderator, a quality control moderator, etc.).

**[0042]** Message moderation engine **164** may be configured to include logic to cause generation of a message moderation interface **122** at a computing device **120** to facilitate moderation or filtering of one or more electronic messages. Message moderation interface **122** may be configured to receive user input **124** to activate or enable automatic disposition (e.g., “auto disposition”) or automatic application of an action, such as automatically approving an unmoderated electronic message as one or more portions of an electronic message translates (e.g., visually moves or scrolls) to a specific position relative to a portion of interface **122**. For example, automatic approval of an unmoderated electronic message may be implemented when any or all portions of a message transitions into a post-view set of messages **127b** (e.g., responsive to a gesture). According to some examples, message moderation engine **164** may

include logic that constitutes a computer program or set of executable instructions for a moderator application.

**[0043]** In some examples, message moderation engine **164** may be configured to access unmoderated electronic messages **130** and **134** for disposition in a queue **124** of electronic messages for moderation. As shown, message moderation interface **122** may include a viewable area bounded by a distance **123** over which one or more images of an electronic post may be accessed, reviewed, and acted upon, according to various examples. Distance **123** may be a function of a direction of, for example a gesture or user input. A viewable area of interface **122** may correlate to a coordinate system or a grid of pixels relative to, for example, a reference **125**. Note, too, that viewable area of interface **122** may represent a display field or matrix of pixels. As an example, a position of a generated image that may be presented or displayed within message moderation interface **122** may be referenced relative to a reference point **125**. Reference point **125** may serve as an origin (e.g., having a coordinate of 0,0) for a Y-axis and an X-axis. Thus, values of Y may extend from Y<sub>n</sub> (e.g., a minimum Y value) adjacent a bottom edge of message moderation interface **122** to Y<sub>x</sub> (e.g., a maximum Y value) adjacent a top edge of message moderation interface **122**. Further, values of X may extend from X<sub>n</sub> (e.g., a minimum X value) adjacent a left edge to X<sub>x</sub> (e.g., a maximum X value) adjacent a right edge of message moderation interface **122**. Note that descriptions, as shown, of a “bottom” edge (i.e., at which Y is Y<sub>n</sub>, such as a “0th” pixel value), a “top” edge (i.e., at which Y is Y<sub>x</sub>, such as a “1920th” pixel value), a “right” edge, and a “left” edge are not intended to be restrictive, but rather, descriptive of an example of delineations between, for example, non-viewable content and viewable content of a queue **124** of electronic messages **140** to **146**, as applied to a moderator application.

**[0044]** According to some examples, a moderator application may be configured to establish queue **124** of electronic messages **140** to **146**, and filter queue **124** of one or more electronic messages (e.g., unmoderated electronic messages) to identify whether to apply an one or more actions (e.g., at least one of which may be performed automatically) in association one of electronic messages **140** to **146** (e.g., unmoderated electronic messages). In at least one implementation, a moderator application (e.g., message moderator engine **164**) may be configured to identify data representing each of electronic messages **140** to **146** to generate images for presentation in message moderation interface **122**. In the example shown, queue **124** of electronic messages may represent data stored in memory that constitutes queue **124**, whereby image data representing electronic message **140** and a portion of electronic message **141** may, at a point of time (e.g., prior to a user input), be associated with a non-viewable region, such as one of pre-view messages **127a** (e.g., prior to presentation in message moderation interface **122**). Also, image data representing electronic message **146** and a portion of electronic message **145** may also be a non-viewable region, such as one of post-view messages **127b** (e.g., subsequent to presentation in message moderation interface **122**). Note, however, electronic messages **142**, **143**, and **144** are viewable as presented or displayed in message moderation interface **122**.

**[0045]** A user input signal may originate at computing device **120** that is configured to implement a user interface as a computerized tool (or a distributed portion thereof),

whereby a user input signal may be received into a moderator application (e.g., message moderation engine 164) to facilitate a moderated action. In some examples, a moderated action may be configured to cause assignment of an approved state automatically to an electronic message, thereby “automatically” approving the electronic message. For example, an approved state may indicate that unmoderated content is transformed into moderated content, thus approved content. According to some implementations, a user input signal may be configured to cause presentation and termination of one or more portions of an electronic message, whereby the user input signal may originate from a user input (e.g., a common user input, such as a single or unitary input with which to perform an automatic action, or multiple automatic actions).

[0046] In at least one example, a user input signal may originate from a gesture-based user input, such as a touch-sensitive screen or display, a touch-sensitive pad (e.g., a touch pad), or any other gesture-based user input as well as any other type of user input (e.g., other than gesture-based user inputs). In diagram 100, user inputs may be represented as implementing, for example, scrolling inputs 126a and 126b in one direction (e.g., scrolling from a bottom edge to a top edge), whereas scrolling input 126c may represent implementation of a common user input in, for example, a reverse direction (e.g., scrolling from a top edge toward a bottom edge). According to some examples, scrolling inputs 126a to 126b may be effectuated by detecting movement of a finger in a specific direction in association with a touch-sensitive user input (or any other gesture-based user input, including contactless gesture interfaces).

[0047] Further to the example set forth in diagram 100, consider that electronic message 143 (i.e., Post #0399) may originate in pre-view message region 127a of queue 124. Responsive to a user input 126a (or one or more user inputs 126a of a common user input), electronic message 143 may transition (e.g., from a non-accessible, but viewable state) to an accessible state. In some examples, user input 126a may cause an image of electronic message 143 to transition from a first region (or position) to a second region (or position). Upon transitioning into an accessible state, electronic message 143 may be identified using a given color, shape, size, or any other visual or perceptible indicator 148.

[0048] As such, different types of indicators may be used and are not limited solely to those that are visual. In an accessible state, electronic message 143 may be associated with a subset of one or more user other inputs 147, which may be activated to perform one or more alternative actions. For example, a moderator 121 may identify an issue with electronic message 143 and activate a user input to, for example, reject, modify (e.g., edit), or forward electronic message 143, whereby a forwarded message may be transmitted to a specialized or supervisory moderator. Or, other user inputs 147 may be configured to implement any other action. Upon implementation of an alternative action, notifications of rejections, modifications, and the like may be transmitted via messages 132 and 136 to originating computing devices 105 and 109 that may have submitted an electronic message or post.

[0049] An alternative action may be activated responsive, for example, to non-compliant or violative content. For example, moderator 121 may cause a gesture indicative of a disapproved or rejected electronic message or post as violating one or more policies or, perhaps, a specific policy or

term or use (e.g., content contains hate-related speech, includes pornographic images, includes prurient content, or the like). The gesture (e.g., a touch-sensitive user input, or scrolling using a mouse or scroll wheel) may indicate that an electronic message or post is to be moved within a given region of an interface 122. In response, the electronic message or post may be deleted and message 132 or 136 is sent back to the submitting user 108 indicating deletion/refusal and, in some examples, a specific policy, rule, or regulation that indicates why the electronic message or post has been refused for posting during a moderation process.

[0050] Note, however, if moderator 121 or message moderation interface 122 of computing device 120 do not cause activation of an alternative action in an accessible state, then another user input 126b may cause electronic message 143 to exit the accessible state. Responsive to a user input 126b (or one or more user inputs 126b of a common user input), electronic message 143 may transition (e.g., from an accessible and viewable state) to a non-accessible state, or to an approved state automatically (e.g., using a common user input or gesture). In some examples, user input 126b may cause an image of electronic message 143 to transition from a second region (or position) to a third region (or position), whereby the transition relative to the third region may cause an automatic action to be applied to electronic message 143 (e.g., an action to automatically approve electronic message 143, or to automatically apply any other default or predicted action).

[0051] In some examples, a moderator application (e.g., message moderation engine 104) may be configured to detect one or more user inputs 126c, which may be a common user input (e.g., same as user inputs 126a and 126b), but optionally in a different or reverse direction. For example, a common user input may be a “moving” or “scrolling” gesture relative to, for example, a touch-sensitive user input. As such, user inputs 126a, 126b, and 126c may be implemented using the same gesture or user input regardless of direction (e.g., user input 126c may be in a reverse direction relative to user inputs 126a and 126b). User input 126c (and associated signal) may be configured to cause presentation of any portion of an electronic message, such as electronic message 143, in message moderation interface 122 subsequent to scrolling that message into a post-view message region 127b. As such, user input 126c may be configured to cause re-display of at least a portion of electronic message 143, which had been previously approved automatically. In some examples, user input 126c may cause electronic message 143 to enter a review state in which moderator 121 may review a previously-approved electronic message. Subsequently, after a review of re-displayed electronic message 143, user inputs 126b may be implemented to again approve automatically that message for publication into a community of electronic messages and posts.

[0052] In view of the foregoing, structures and/or functionalities depicted in FIG. 1 as well as other figures herein, illustrate one or more applications, algorithms, systems and platforms to leverage or otherwise implement common user inputs, such as common gestures, to cause one or more actions automatically to be applied to an electronic message or post under moderation and review, according to some embodiments.

[0053] According to one or more examples, a moderation application and its functionalities (as well as any other

functionalities described herein) are configured to reduce user behavior required to moderate content for posting to one or more social media networks, using one or more software platforms on computing devices over one or more distributed data networks (e.g., cloud-based data networks) to review, approve, reject, or perform other actions automatically on individual electronic messages submitted for posting without performing active actions (e.g., mouse or button clicks, or employing multiple different inputs in a touch-sensitive interface). Rather, a moderation application may be implemented to employ a gesture-based user interface to correlate a user input (e.g., a common user input) to select and approve actions to be performed automatically based on values of the user input. As such, an electronic message may be automatically approved without requiring different manual user inputs, thereby obviating a need to select an “approve” user input interface element.

**[0054]** Implementation of a moderator application, as described herein, may reduce or negate fatigue that otherwise may afflict moderators **121**, which, in turn, may reduce or negate lapses of focus, inadvertent posts, errors, and the like. This may enhance user efficiency and accelerate a moderation process. Therefore, a moderation application may effectuate a computerized tool that may reduce cost and resources that otherwise may be necessitated to, for example, review and correct posts. Further, a moderator application may facilitate preservation of the reputation, trustworthiness, and reliability of an online community.

**[0055]** Note that message moderation interface **122** may implement, for example, functionalities provided by Khoros® Manage View user interface and a Khoros® Community software platform. Any of described elements or components set forth in FIG. 1, and any other figure herein, may be implemented as software, applications, executable code, application programming interfaces (“APIs”), processors, hardware, firmware, circuitry, or any combination thereof.

**[0056]** FIG. 2 depicts an example of a message moderation engine, according to various examples. Diagram **200** depicts a message moderation engine **264** configured to, at least in some examples, provide functionalities of a moderator application, whereby moderated electronic messages **201a** and posts may be received and stored in a repository or memory, such as an unmoderated message data repository **202**. Further, message moderation engine **264** may be configured to generate a moderated electronic message **201c** or post for storage in a moderated message data repository **210**, which may store a data arrangement that constitutes an online community. As such, moderated electronic message **201c** may be viewable or accessible via any network **211** as message data to any authorized member of an online community.

**[0057]** Diagram **200** depicts a message moderation engine **264** including a queue manager **271**, an image generator **272**, a moderation processor **274**, and an action generator **280**. Queue manager **271** may include logic configured to manage storage of unmoderated electronic messages **201a** in unmoderated message data repository **202**. Further, queue manager **271** may also include logic configured to fetch data representing unmoderated messages from repository **202** for image processing to display a formatted electronic message as part of a queue of electronic messages under moderation, as displayed in a moderator user interface. In accordance with some embodiments, queue manager **271** may be con-

figured to arrange electronic messages for moderation as a function of a message attribute, such as a level of priority, or any other message attribute.

**[0058]** Image generator **272** may include logic configured to receive data representing unmoderated messages from queue manager **271**, whereby the logic may be further configured to generate a queue of electronic messages including any type of content, such as text, audio, video, graphics, etc. In some examples, image generator **272** may generate data representing a queue of electronic messages including viewable and non-viewable imagery in preparation for display in a user interface. As shown, image generator **272** may include an image display controller **273** that may be configured to receive data representing values of user input data from a user input **210** associated with a user interface. Responsive to user input data, image display controller **273** may be configured to cause presentation of electronic messages, and portions thereof, in a viewable area of a user interface so as to facilitate moderation.

**[0059]** In some examples, user input **210** may be configured to generate gesture-based user input data (e.g., that can be embodied in an associated user input data signal). As such, user input **210** may be activated based on any number of gestures, such as scrolling, pinching, and tapping, as well as tipping, shaking, or tilting a device, among other gestures. In some examples, a gesture-based user input may be activated responsive to eye movement (e.g., eye-gaze tracking), or to motion of any part of a user (e.g., with or without direct contact).

**[0060]** Moderation processor **274** may include a gesture detector **275**, a correlator **276**, and an action selector **277**, according to at least some examples. In various examples, moderation processor **274** may be configured to determine and/or assign a state (e.g., an accessible state, an approved state, etc.) for an electronic message as a function of data values of a user input signal received from user input **210**. Gesture detector **275** may be configured to receive user input signal data from user input **210** to determine a region or position (e.g., relative to a reference) for any electronic message or post presented in a viewable area of a user interface. Correlator **276** may include logic configured to detect a region or position of an electronic message (or any portion thereof), and to correlate the region or position to one or more automatically-invoked actions (e.g., based on the user input, such as a common user input). Action selector **277** may be configured to receive data representing a correlation between a detected set of user input data values and actions (e.g., automatic actions) that are configured to trigger or otherwise activate as a function of the set or range of user input data values. Thus, action selector **277** may be configured to activate an action automatically (e.g., responsive to a common user input data signal) or to activate an alternative action based on, for example, receiving other user inputs (e.g., to reject, to edit, etc.) based on accessibility to an electronic message under moderation.

**[0061]** Consider an example in which gesture detector **275** may be configured to detect a first subset of values of a user input signal, whereby the first subset of values of a user input signal may be configured to cause presentation of an electronic message (or a portion thereof) in a user interface for a moderator. In some examples, correlator **276** may be configured to correlate a first subset of values to a range of values responsive to detected values of a signal. In some examples, correlator **276** may detect a range of pixels, each

pixel being uniquely identifiable relative to a reference (e.g., a bottom edge of a viewable area of a user interface, such as  $Y=Y_n$  of FIG. 1). Based on a detected range of pixels, or a displacement of a number of pixels (or any other unit of an image) relative to a reference, correlator 275 may be configured to identify that an action correlates with a detected range of pixels. In some implementations, one or more ranges of pixels may correlate to one or more corresponding actions, one or more of which may be activated automatically.

[0062] In one example, correlator 276 may be configured to correlate user input data that may position an electronic message in a user interface at a region associated with a first range of pixel values, the first range of pixel values being associated with an accessible state. In response, action selector 277 may be configured to cause that electronic message to receive other user inputs to invoke alternate actions (e.g., to reject, edit, or forward a post), whereby action selector 277 may be activated to select a corresponding action to be performed. In another example, correlator 276 may be configured to correlate a second subset of user input data that may position an electronic message in a user interface in another region that may be associated with a second range of pixel values. The second range of pixels values may be associated with an approved state. In response, action selector 277 may be configured to cause approval of the electronic message, thereby enabling an approved electronic message to be published into an online community. In yet another example, correlator 276 may be configured to correlate a third subset of user input data that may re-position an electronic message in a visible area of a user interface so that an electronic message, such as a previously-approved message, may be reviewed and/or accessed. In some cases, the third subset of user input data may be similar or equivalent to the first subset or the second subset of data, but may be associated with a user input data signal indicative of scrolling in a reverse direction.

[0063] Action generator 280 may include logic configured to implement any number of actions responsive to any user input. In some examples, action generator 280 may be configured to implement one or more actions automatically as a function of the user input signal. Action generator 280 may include an automatic action engine 281, which, in turn, may include a supplemental automatic action controller 282. In addition, action generator 280 may include an alternative action engine 283. Automatic action engine 281 may be configured to receive a signal from action selector 277 to implement or execute an action automatically as a function a region or position at which an electronic message (e.g., in a queue) may be detected in a viewable area of a user interface. For example, automatic action engine 281 may activate any of a multiple number of actions automatically based on detected subsets of user input signal values, whereby differently detected subsets of user input signal values may automatically invoke actions based on an accessible state or an approved state of an electronic message under moderation. Automatic action engine 281 may be configured to activate supplemental automatic action controller 282 to automatically implement a supplemental action, such as causing review of a previously-reviewed electronic message that may be scrolled down into a viewable area of a user interface for further review. Alternative action engine 283 may be activated in response to one or more user inputs associated with an electronic message in an

accessible state, whereby the one or more user inputs may cause an accessible electronic message to be rejected, to be edited, to be forwarded, or the like.

[0064] FIG. 3 illustrates an exemplary layered architecture for implementing a moderator application, according to some examples. Diagram 300 depicts application stack (“stack”) 301, which is neither a comprehensive nor a fully inclusive layered architecture for moderating electronic posts and messages of an online community or social media network, including performing automatic actions using, for example, gesture-based user inputs or the like. One or more elements depicted in diagram 300 of FIG. 3 may include structures and/or functions as similarly-named or similarly-numbered elements depicted in other drawings, or as otherwise described herein, in accordance with one or more examples, such as described relative to FIG. 1 or any other figure or description herein.

[0065] Application stack 301 may include an electronic message moderation engine layer 350 upon application layer 340, which, in turn, may be disposed upon any number of lower layers (e.g., layers 303a to 303d). Electronic message moderation engine layer 350 may be configured to provide functionality and/or structure to implement a moderator application, as described herein. Electronic message moderation engine layer 350 and application layer 340 may be disposed on data exchange layer 303d, which may implemented using any programming language, such as HTML, JSON, XML, etc., or any other format to effect generation and communication of requests and responses among computing devices and computational resources constituting an enterprise and an enterprise resource planning application and/or platform. Data exchange layer 303d may be disposed on a service layer 303c, which may provide a transfer protocol or architecture for exchanging data among networked applications. For example, service layer 303c may provide for a RESTful-compliant architecture and attendant web services to facilitate GET, PUT, POST, DELETE, and other methods or operations. In other examples, service layer 303c may provide, as an example, SOAP web services based on remote procedure calls (“RPCs”), or any other like services or protocols (e.g., APIs). Service layer 303c may be disposed on a transport layer 303b, which may include protocols to provide host-to-host communications for applications via an HTTP or HTTPS protocol, in at least this example. Transport layer 303b may be disposed on a network layer 303a, which, in at least this example, may include TCP/IP protocols and the like. Note that in accordance with some examples, layers 303a to 303d facilitate implementation of a risk management data channel as set forth herein.

[0066] As shown, electronic message moderation engine layer 350 may include (or may be layered upon) an application layer 340 that includes logic constituting a community syndication controller layer 320, a message management controller layer 330, a presentation engine 310, and an asset layer 312. According to some examples, community syndication controller 320 may include logic to implement an online community, such as the Lithium Community (formally of Lithium Technologies, LLC), Khoros Communities of Khoros, LLC of Austin Texas, or “Atlas” Communities of Khoros, LLC of Austin Texas, among other online community configurations. Further, message management controller layer 330 may include logic to implement at least some functionality provided by an application configured to

operate in accordance with “Modern Chat”-related technologies and “Khoros Care”-related technologies, both of Khoros, LLC of Austin Texas, among other technologies. Presentation engine layer 342 may include logic configured to facilitate presentation of electronic messages, as well as associated functionalities (e.g., to detect position or ranges of pixels associated with a displayed electronic message under moderation). In some examples, an asset layer 312 may be configured to implement node.js, which may be a cross-platform, JavaScript runtime environment. In some cases, node.js may execute JavaScript code independent of a browser, or any other protocol, any other programming language, or any other set of executable instructions. Node.js is maintained by the Linux Foundation of San Francisco, CA, USA.

**[0067]** Any of the described layers of FIG. 3 or any other processes described herein in relation to other figures may be implemented as software, hardware, firmware, circuitry, or a combination thereof. If implemented as software, the described techniques may be implemented using various types of programming, development, scripting, or formatting languages, frameworks, syntax, applications, protocols, objects, or techniques, including, but not limited to, Python™, ASP, ASP.net, .Net framework, Ruby, Ruby on Rails, C, Objective C, C++, C#, Adobe® Integrated Runtime™ (Adobe® AIR™), ActionScript™, Flex™, Lingo™, Java™, JSON, Javascript™, Ajax, Perl, COBOL, Fortran, ADA, XML, MXML, HTML, DHTML, XHTML, HTTP, XMPP, PHP, and others, including SQL™, SPARQL™, Turtle™, etc., as well as any proprietary application and software provided or developed by Khoros, LLC, or the like. The above described techniques may be varied and are not limited to the embodiments, examples or descriptions provided.

**[0068]** FIG. 4 is a flow diagram as an example of moderating an electronic message, according to some embodiments. Flow 400 may be an example of facilitating moderation of electronic messages, postings, content, etc., to determine whether to include electronic messages in an electronic community (or any subset thereof). In some examples, a subset of an electronic community (e.g., online community) may include any number of electronic messages or posts that may relate to each other by subject matter or any other classification. As an example, an online community may be subdivided based on whether content relates to a “forum” (e.g., content directed to resolving a problem), an “idea” (e.g., content directed to proposed suggestions related to any item, such as a product), a “frequently-asked question” (e.g., content directed to searchable solutions that are determined to be effective), an “expert” classification (e.g., directed to users or electronic accounts associated with expert-based content), and any other classification or categorization.

**[0069]** At 402, a moderator application (or a subset of executable instructions) may be configured to perform one or more actions automatically, such as approving an electronic message as a post in the electronic community, according to some examples. In some implementations, a moderator application may be implemented in association with a computing platform configured to host a syndication of subsets of electronic messages (e.g., an electronic community). A moderator application may be configured to filter a queue of one or more electronic messages (e.g., unmoderated electronic messages) to, for example, identify

whether to apply an one or more actions (e.g., at least one of which may be performed automatically) in association electronic message (e.g., an unmoderated electronic message)

**[0070]** At 404, a user input signal may be received, whereby the user input signal may be configured to cause presentation of an electronic message at a user interface as a computerized tool to facilitate a moderated action of the moderator application. In some examples, a moderated action may be configured to cause assignment of an approved state automatically to an electronic message, thereby “automatically” approving the electronic message. For example, an approved state may indicate that unmoderated content is transformed into moderated content. According to some implementations, a user input signal may be configured to cause presentation and termination of one or more portions of an electronic message, whereby the user input signal may originate from a user input (e.g., a common user input, such as a single or unitary input with which to perform an automatic action, or multiple automatic actions). In at least example, a user input signal may originate from a scroll bar interface element (e.g., a vertical scroll bar) or a mouse-based user input, such as a mouse wheel configured to implement scrolling functionalities. Note, however, any other user input may be implemented. In some examples, touch sensitive-based user inputs may be implemented (e.g., touch-sensitive screens or touch-sensitive pads (e.g., touch pads)), as well as one or more voice-based or audio-based inputs (e.g., speech-to-text/input computer program). Another example of a user input may include eye-gaze detection and tracking devices and/or software, or any other technology configured to convey a user input to a moderator application.

**[0071]** At 406, a first subset of values of a user input signal may be detected, whereby the first subset of values of a user input signal may be configured to cause presentation of an electronic message (or a portion thereof) in a user interface. In some examples, a first subset of values may correlate to, or may be associated with, a range of values responsive to detected values of a signal.

**[0072]** At 408, a first state may be assigned to an electronic message based on a first subset of values. In at least one example, a first state associated with an electronic message may be indicative of an accessible state. An electronic message may be detectable in an accessible state in association with a user interface, whereby an electronic message may be accessible to receive another user input (e.g., a second user input). An example of another user input may be associated with activating a user interface element (e.g., a button) or causing a touch-sensitive input in a specific direction. In one example, another user input may not be activated or detected in association with an electronic message in an accessible state, whereby the electronic message may transition to a second state (e.g., automatically) responsive to a user input signal (e.g., a common user input signal). In another example, another user input may be activated or detected in association with an electronic message in an accessible state. In response, a moderator application may be configured to provide and detect the other user input (e.g., a second user input), which may be configured to transition an electronic message from a first state to a rejected state, an editable state, or a forwarded state, among others.

**[0073]** Responsive to detecting a user input to transition an electronic message to a rejected state, a moderator application can classify the message as “rejected,” prevent the message from being transitioned into an “approved” state, generate a notification to transmit to an author of the message (i.e., notifying the author of rejection) with optional reasons to facilitate correction, and to implement any other function. Upon detecting a user input to transition an electronic message to an editable state, a moderator application can classify the message as “editable,” enable a user (e.g., a moderator, an agent, or other users) to modify the message to, for example, redact a profane word. The moderator application can be further configured to generate a notification to transmit to an author of the message (i.e., notifying the author of a modification) and to implement any other function. Thereafter, the message may advance to an approved state, in at least some cases. Another user input may be configured to cause an electronic message to transition to a forwarded state, whereby a moderator application can be configured to classify the message as “forwarded,” enable the message to be transmitted to another user (e.g., an expert moderator, a supervisory agent, or other users). Thereafter, the message may advance to an approved state, in at least some cases.

**[0074]** At **410**, a second subset of values of a user input signal can be detected, whereby the second subset of values of a user input signal may be configured to cause termination of the presentation of at least a second portion of the electronic message in a user interface. In at least one example, detection of a first subset and a second subset of values of a user input signal may include computing displacement of a number of pixels (or any other unit of an image) relative to a reference associated with the user interface, and detecting a displacement value a function of the number of pixels to determine a transition from at least one of the first state and the second state of the electronic message to another state. For example, a user interface element may be configured to cause an electronic message (or post) to modify its presentation by scrolling up in a first direction. In at least one other example, detection of a first subset and a second subset of values of a user input signal indicative of a position of an electronic message (or an image thereof), as presented or displayed on a user interface.

**[0075]** Further to **410**, a first subset of values of a user input signal may be detected by, for example, detecting that a first subset of values may be indicative that a first portion of an electronic message is viewable in a user interface (e.g., a first portion of an electronic message may enter a user interface at a first edge of a user interface). A second subset values of a user input signal may be indicative that a second portion of an electronic message is not viewable in a user interface (e.g., a second portion of an electronic message may exit a user interface at a second edge of a user interface). In some implementations, a first edge and a second edge may be a bottom edge and a top edge, respectively, of a user interface (e.g., a viewable portion thereof).

**[0076]** At **412**, a second state may be assigned to an electronic message based on a second subset of values of a user input signal. A second subset of values may include a range of values indicative of activating a transition to a second state (e.g., indicative of moderator intent). In some examples, a second state is an “approved” state

**[0077]** At **414**, execution of instructions may cause performance of a moderated action automatically responsive to

transitioning an electronic message from a first state to a second state (e.g., from an accessible state to an approved state). In some examples, a third subset of values of a user input signal may be configured to cause presentation of any portion of an electronic message in a user interface (e.g., re-display of at least a portion of an electronic message). In some examples, a third subset of values may be detected subsequent to detecting a second subset values of a user input signal (e.g., subsequent to an approved state). Further, another state to an electronic message based on a third subset of values. In at least one example, the third subset of values may indicate a review of an approved electronic message. According to some examples, a third subset of values may be equivalent to (or overlap) one or more other subsets of values. A third subset of values may be equivalent to at least portions of either a second subset of values (e.g., to revoke approval) or a first subset of values (e.g., to re-access an electronic message in an accessible state). A third subset of values may be detected at a subsequent period or unit of time. In at least one case, detecting a third subset of values may include receiving a user input signal specifying data representing a reverse displacement of a number of pixels. As such, a user input signal may originate from a common user interface element, and may be in a second or reverse direction, such as scrolling down (e.g., a reversed direction) to pull down or re-display an approved electronic message through a top of a user interface.

**[0078]** FIG. 5 is a flow diagram as another example of moderating a number of electronic message, according to some embodiments. Flow **500** may begin at **502**, at which a moderator application may be activated to filter a number of queues each including a subset of electronic messages. A moderator application may be implemented in association with a computing platform, which may be configured to host a syndication of electronic messages (e.g., an online community). At **504**, signals configured to cause presentation of a number of electronic messages at a user interface may be received. In some examples, a moderator application implemented in electronic communication with a user interface may configure the user interface as a computerized tool to facilitate moderated actions.

**[0079]** At **506**, positions at which to present electronic messages may be detected, whereby one or more positions may be indicative of one or more states that may activate one or more actions. At **508**, subsets of electronic messages may be filtered to facilitate approval automatically of approved electronic messages.

**[0080]** At **510**, a determination may be made as to whether to perform an automatic action, such as action that specifies that an electronic message is approved, for example, to post within an online community. If no, flow **500** may transition to **512**, at which accessibility to electronic messages may be detected to enable performance of an action at **514**. Examples of such actions include rejecting, modifying, or forwarding an electronic message. Otherwise, flow **500** may transition to **516**, whereby performance of an automatic action may be indicative of approving an electronic message or post automatically (e.g., using a common user input).

**[0081]** At **516**, a determination may be made as to whether to perform a supplemental action, such as action that specifies that an approved electronic message may be reviewed by, for example, re-displaying the electronic message in a user interface. If no, flow **500** may transition to termination, at which the approved electronic message or post may be

published into an online community. If yes, flow **500** may transition to **518**, whereby accessibility of electronic messages may be detected. Accessible message may be configured to receive other user inputs to perform other actions at **520** (e.g., other action including, but not limited to, rejecting, modifying, or forwarding an electronic message).

[0082] FIG. 6 depicts another example of a message moderation engine, according to various examples. Diagram **600** depicts a message moderation engine **664** configured to, at least in some examples, provide functionalities of a moderator application, whereby moderated electronic messages **601a** and posts may be received and stored in a repository or memory, such as an unmoderated message data repository **602**. Further, message moderation engine **664** may be configured to generate a moderated electronic message **601c** or post for storage in a moderated message data repository **610**, which may store a data arrangement that constitutes an online community. As such, moderated electronic message **601c** may be viewable or accessible via any network **611** as message data to any authorized member of an online community.

[0083] Diagram **600** depicts a message moderation engine **664** including a queue manager **671**, an image generator **672**, a moderation processor **674**, and an action generator **680**. As shown, image generator **672** may include an image display controller **673**, moderation processor **674** may include a gesture detector **675**, a correlator **676**, and an action selector **677**, and action generator **680** may include an automatic action engine **681**, which, in turn, may include a supplemental automatic action controller **682**. In addition, action generator **680** may include an alternative action engine **683**. Message moderation engine **664** may include structures and/or functionalities set forth in FIGS. 1 and 2, or any other figure, and may include additional structures and/or functionalities described in FIGS. 1 and 2 and elsewhere herein. In one or more implementations, elements depicted in diagram **600** of FIG. 6 may include structures and/or functions as similarly-named or similarly-numbered elements depicted in other drawings.

[0084] According to various examples, message moderation engine **664** may include other or additional functionality. For example, moderation processor **674** may include logic configured to implement a rule processor **678** and a message modifier **679**, and automatic action engine **681** may include logic configured to implement an automatic action predictor **684**.

[0085] In at least one embodiment, moderation processor **674** may be configured to provide automatically a modified electronic message or post in a user interface with which a moderator may review and approve automatically, based on a user input signal data and correlation thereto as described herein. In this case, moderation processor **674** may be configured to provide a “modification assist” functionality, whereby logic may be configured to propose modifications to an unmoderated message that a moderator may approve automatically (e.g., without activating another user input). For example, moderation processor **674** may be configured to detect a noncompliant message attribute, such as a typographical error or a profane word, and may be configured to propose a modified electronic message for approval. Rule processor **678** may be configured to access rule data in a rule data repository **614**, whereby rule data may include any number of rules with which to analyze an unmoderated

electronic message (or post) to determine whether to approve or modify the message (e.g., automatically).

[0086] Rule data stored in repository **640** may include data arrangements of various threshold values to determine whether an unmoderated message may comply with community terms (e.g., terms of use), conditions, policies, laws, regulations, rules, and the like. For example, rule data **614** may include rule data to reject, block, or redact variants of words, word stems, tokens (in any language), and the like, that may be profane, libelous, perjurious, hateful, and the like. Message modifier **679** may include logic configured to detect violations or breaches of terms or policies, and to modify an electronic message (e.g., by replacing words, images, etc., by redacting inappropriate content, etc., and the like). In one example, rule data stored in repository **640** may define threshold or conditions to determine whether an electronic message conforms with, for instance, the Americans with Disabilities Act (“ADA”) or the like.

[0087] In at least one embodiment, an automatic action predictor **684** may include logic configured to provide “action assistance” to a moderator, whereby automatic action predictor **684** may predict whether to select or modify a default action to implement automatically a predicted disposition of an electronic message. As an example, automatic action predictor **684** may be configured to access model data **612** to compute or calculate, based on various attributes (e.g., message attribute data) of an electronic message, a predicted disposition of an electronic message, such as a prediction to approve, a prediction to reject, a prediction to edit, a prediction to forward, or any other predictive actions.

[0088] In some examples, message characterizer **620** may be configured to characterize one or more messages **601a** to determine or predict various characterized message attributes with which to assist in modifying a message or assist in providing a predicted action during moderation of one or more electronic messages. In some examples, message characterizer **620** may be configured to characterize, for example, a “newly-received” message **601a** for comparison against a data model in model data repository **612** to form a set of characterized data. Thus, message characterizer **620** may be configured to identify attributes and corresponding attributes that may be matched, as a data pattern, against patterns of data including correlated datasets stored in, for example, model data **612**. Matching patterns may facilitate the correlation of message characteristics to assist in providing an optimal response during a moderation process. In various examples, one or more rules implemented in executable instructions may be configured to generate an optimized electronic message for review by a moderator.

[0089] Message characterizer **620** may be configured to characterize content of message **601a** to identify or determine one or more attributes such as, for example, a status of an author or customer, a number of times an author or customer has had an electronic message rejected or modified, an associated URL, a referrer computing device, application, website, or link, one or more site visits, a number of days since a customer last interacted digitally with a website or application, an amount of time on a web page or web site, meta and cookie-related data, a location (including GPS coordinates, city, country, etc.), an operating system, a type of browser application, a device type (e.g., a hardware identification of a computing device), a MAC ID, an IP address, and other message attribute that may be character-



ized. One or more message characteristics may facilitate characterization or classification of unmoderated messages to, for example, optimize moderation processes at computing devices based on one or more detected or derived message characteristics. In some examples, message characterizer may derive a characteristic indicative of a priority value, or any other factor that may affect moderation of electronic messages.

**[0090]** Further, message characterizer **620** may be configured to detect and parse the various components of an electronic message, and further may be configured to perform analytics to analyze characteristics or attributes of one or more message components. As shown, message characterizer **620** may include a natural language processor **631** and a message component attribute determinator **632**. Natural language processor **631** may be configured to ingest data to parse portions of an electronic message (e.g., using word stemming, etc.) for identifying components, such as a word or a phrase. Also, natural language processor **631** may be configured to derive or characterize a message as being directed to a particular topic or subject matter based on, for example, sentiment analysis techniques, content-based classification techniques, and the like. In some examples, natural language processor **631** may be configured to apply word embedding techniques in which components of an electronic message may be represented as a vector, which may be a data arrangement for implement machine learning, deep learning, and other artificial intelligence-related algorithmic functions.

**[0091]** Message component attribute determinator **632** may be configured to identify characteristics or attributes, such as message attribute data **603**, for a word, phrase, topic, etc. In various examples, message attribute data **603** may be appended, linked, tagged, or otherwise associated with a component to enrich data in, for example, model data repository **612**. A classification value may be a characteristic or an attribute of a message component, and thus may be used as a “tag.” Examples of message attribute data **603** are depicted as classification data **603a** (e.g., an attribute specifying whether a component or message may be classified as, for example, being directed to particular subject matter, or being direct to non-compliant messaging), media type data **603b** (e.g., an attribute specifying whether a component may be classified as being associated with a Tweet™, an email, a post, a webpage, a text message, etc.), channel type data **603c** (e.g., an attribute specifying whether a component may be associated with a type of social networking system, such as Twitter™), and the like. Message attribute data **603** may also include context metadata **603d**, which may include attributes that specify environmental data or contextual data, such as a context in which an electronic message is received for submission into a particular community. For instance, context metadata **603d** may include data representing a time of day, a year, a season, a subject matter-related context, a product-related context, an idea-related context, a solution-related context, a service-related context, a payment-related context, etc.

**[0092]** Also, message component attribute determinator **632** may be configured to generate a tag including metadata **603d** may refer to a context in which a word is used in a transmission of a number of electronic messages (e.g., a tag indicating a marketing campaign, a tag directed to a particular community or sub-community, or the like). Also, a tag including metadata **603d** may refer to an industry or

activity (e.g., a tag indicating an electronic message component relating to autonomous vehicle technology, or basketball), etc. Furthermore, message attribute data **603** may also include profile data **603e**, which may include attributes that describe, for example, demographic data regarding an author or a customer of a received electronic message, or the like. Other metadata **603f** may be associated with, or tagged to, a word or other message component. As such, other metadata **603f** may include a tag representing a language in which the word is used (e.g., a tag indicating English, German, Mandarin, etc.). In some cases, other metadata **603d** may include data representing values of computed threshold values or classification values (e.g., a tag may indicate a value of an amount of likelihood of generating a response, etc.). Message attribute data **603**, and the corresponding tags, may be stored in a data repository.

**[0093]** Data correlator **633** may be configured to statistically analyze components and attributes of electronic messages and posts bound for submission to a community to identify predictive relationships between, for example, an attribute and a value predicting a likelihood that an electronic message may invoke a specific predictive action, which may be moderated by an agent, a moderator, or the like. According to some embodiments, data correlator **633** may be configured to classify and/or quantify various “attributes” and/or “received electronic messages” (and exchanges thereof) by, for example, applying machine learning or deep learning techniques, or the like.

**[0094]** In one example, data correlator **633** may be configured to segregate, separate, or distinguish a number of data points (e.g., vector data) representing similar (or statistically similar) attributes or received electronic messages, thereby forming one or more sets of clustered data. Clusters of data (e.g., predictively grouped data) may be grouped or clustered about a particular attribute of the data, such as a source of data (e.g., a channel of data), a type of customer (e.g., a loyal customer), a degree of urgency for an issue (e.g., a customer is, a type of language, a degree of similarity with synonyms or other words, etc., or any other attribute, characteristic, parameter or the like. In at least one example, a cluster of data may define a subset of electronic messages having one or more similarities (e.g., a statistically same topic) that may be configured to characterize a class of messages for purposes of selecting and applying predictively one or more rules or more actions to unmoderated message **601a**.

**[0095]** While any number of techniques may be implemented, data correlator **633** may apply “k-means clustering,” or any other clustering data identification techniques to form clustered sets of data that may be analyzed to determine or learn optimal classifications of data and associated predictive responses thereto. In some examples, data correlator **633** may be configured to detect patterns or classifications among datasets **633** and other data through the use of Bayesian networks, clustering analysis, as well as other known machine learning techniques or deep-learning techniques (e.g., including any known artificial intelligence techniques, or any of k-NN algorithms, linear support vector machine (“SVM”) algorithm, regression and variants thereof (e.g., linear regression, non-linear regression, etc.), Bayesian inferences and the like, including classification algorithms, such as Naïve Bayes classifiers, or any other statistical or empirical technique).

[0096] In the example shown, data correlator 633 may be configured to implement any number of statistical analytic programs, machine-learning applications, deep-learning applications, and the like. Data correlator 633 is shown to have access to any number of predictive models, such as predictive model 690a, 690b, and 690c, among others. In this implementation, predictive data model 690a may be configured to implement one of any type of neuronal networks to predict an action or disposition of an electronic message under moderation, so as to minimize a number of different user inputs in use (e.g., to enhance moderator efficiency and reduce fatigue). In this case, a neural network model 690a includes a set of inputs 691 and any number of “hidden” or intermediate computational nodes 692 and 693, whereby one or more weights 697 may be implemented and adjusted (e.g., in response to training). Also shown, is a set of predicted outputs 694, such as “approve” or “reject,” among any other type of output.

[0097] In view of the foregoing, rule processor 678 and message modifier 679 may be configured to operate to predictively or proactively suggest modifications to an electronic message, thereby enabling a moderator to forgo accessing a non-compliant electronic message to reject, or modify the electronic message. Further, automatic action predictor 684 may be configured to analyze characterized message data of electronic message 601a to provide a proposed default course of action or disposition of an electronic message. Therefore, an electronic message under moderation may be associated with a connotation of its predicted disposition, such as whether a message ought to be approved, reviewed, edited, rejected, forwarded, and the like. As such, a moderator may forego actively selecting an alternative action manually as automatic action predictor 684 may be configured to propose an alternative action that may be implemented automatically (e.g., automatically rejecting or editing a message).

[0098] Any of described elements or components set forth in FIG. 6, and any other figure herein, may be implemented as software, applications, executable code, application programming interfaces (“APIs”), processors, hardware, firmware, circuitry, or any combination thereof.

[0099] In some examples, computing devices using computer programs or software applications may be used to implement gesture-based community moderation, using computer programming and formatting languages such as Java®, JavaScript®, Python®, HTML, HTML5, XML, and data handling techniques and schemas. Moderation may be performed for various purposes ranging from reviewing/publishing content to moderating user posted content to a content, news, or video aggregation site such as YouTube® or a social media website or network such as Twitter®, Facebook®, Instagram®, Snapchat®, or others.

[0100] FIG. 7 depicts an example of a user interface configured to moderate electronic messages and posts, according to various examples. Diagram 700 depicts a user interface 701 configured to moderate electronic messages (or posts) in accordance with various methods described herein. In this example, a user interface includes a portion 702 describing a specific moderator view indicating an associated queue of unmoderated electronic messages in an active user interface (“UI”) portion 731. User interface 701 depicts a presentation of a number of electronic messages under moderation, such as messages 737d, 737b, 737, 737a, and 737c. User inputs, such as gesture-based user inputs

726a, 726b, and 726c may be configured to scroll messages 737d, 737b, 737, 737a, and 737c up and down, whereby one or more automatic actions may be applied to an electronic message. For example, electronic message 737 may be detected by a moderator application, as an example, as being displaced into a region or position that automatically assigns or places electronic message 737 into an accessible state. In this state, a visual indicator 748 indicates that electronic message 737 may be accessible in response to, for example, alternative actions activated by one or more user inputs 747. If no alternative actions are activated via user input 747, then one or more user input 726b may cause electronic message 737 to scroll off a visible area of user interface 701, thereby automatically approving that message.

[0101] FIG. 8 depicts another example of a user interface configured to moderate electronic messages and posts using proposed modified electronic messages, according to various examples. Diagram 800 depicts a user interface 801 configured to moderate electronic messages (or posts) in accordance with various methods described herein. In this example, a user interface includes a portion 802 describing a specific moderator view indicating an associated queue of unmoderated electronic messages in an active user interface (“UI”) portion 831. Note that user interface 801 includes a user input 807 to cause a moderator application to approve scrolled posts, or to disable that functionality.

[0102] User interface 801 depicts a presentation of a number of electronic messages under moderation, such as messages 837d, 837b, 838, 837a, and 837c in a queue, whereby message 838 is a modified version of original electronic message 837, which is presented or displayed at its side. In this example, a moderator application or other logic may be configured to detect inappropriate or noncompliant text 889, and propose alternative words or text 888. In this state, a visual indicator 848 indicates that electronic message 838 may be accessible in response to, for example, alternative actions activated by one or more user inputs 847, such as whether to undo or reject proposed alternate modifications 888. User inputs, such as gesture-based user inputs 826a, 826b, and 826c may be configured to scroll messages 837d, 837b, 838, 837a, and 837c up and down, whereby one or more automatic actions may be applied to an electronic message. If no alternative actions are activated via user input 847, then one or more user input 826b may cause electronic message 838 to scroll off a visible area of user interface 801, thereby automatically approving that a modified version of electronic message 837.

[0103] FIG. 9 depicts yet another example of a user interface configured to moderate electronic messages and posts using predictive default actions, according to various examples. Diagram 900 depicts a user interface 901 configured to moderate electronic messages (or posts) in accordance with various methods described herein. In this example, a user interface includes a portion 902 describing a specific moderator view indicating an associated queue of unmoderated electronic messages in an active user interface (“UI”) portion 931. Note that user interface 901 includes a user input 907 to cause a moderator application to automatically apply a predicted default action to posts scroll up beyond visibility of user interface 901, or to disable that functionality.

[0104] User interface 901 depicts a presentation of a number of electronic messages under moderation, such as messages 937d, 937b, 937, and 937a in a queue, whereby a

moderator application predictively calculates predictive default actions each of the messages. For example, messages **937d**, **937b**, **937**, and **937a** are shown to be associated with predicted default actions (“Reject”) **941**, (“Approve”) **942**, (“Approve”) **944**, and (“Approve”) **946**, respectively. User inputs, such as gesture-based user inputs **926a**, **926b**, and **926c** may be configured to scroll messages **937d**, **937b**, **937**, and **937a** up and down, whereby one or more automatic actions and dispositions may be applied to an electronic message. If no alternative actions are activated via user input **947**, as an example, then one or more user input **926b** may cause electronic messages **937d**, **937b**, **937**, and **937a** to scroll off a visible area of user interface **901**, whereby predicted default actions may be applied. Thus, electronic message **937d**, if no user inputs **947** are detected, may be rejected automatically as a predicted default disposition, and based on a common user input or gesture.

[0105] FIG. 10 is a flow diagram as another example of moderating a number of electronic message using either proposed modified message content or predicted automatic default actions, or both, according to some embodiments. Flow **1000** may begin at **1002**, at which a moderator application may be activated to filter a number of queues each including a subset of electronic messages. A moderator application may be implemented in association with a computing platform, which may be configured to host a syndication of electronic messages (e.g., an online community). At **1004**, signals configured to cause presentation of a number of electronic messages at a user interface may be received. In some examples, a moderator application implemented in electronic communication with a user interface may configure the user interface as a computerized tool to facilitate moderated actions. At **1006**, positions at which to present electronic messages may be detected, whereby one or more positions may be indicative of one or more states that may activate one or more actions.

[0106] At **1008**, a moderator application may be configured to cause display of a modified electronic message based on, for example, detection of non-compliant message components (e.g., profane text), or the like. At **1010**, a predicted automatic action may be predicted so as to implement a default action automatically, whereby the predicted automatic action may be any default action, such as approval, rejection, or the like. At **1012**, a detected change in position of an electronic message in a viewable area of a user interface may invoke or cause activation of an automatic action, such as a predicted automatic default action.

[0107] FIG. 11 depicts an example of a system architecture to provide a computing platform to host a syndication of electronic messages and posts for moderation, according to an example. Content, messages, and/or posts may be stored in (or exchanged with) various communication channels or storage devices as unmoderated or moderated content. For example, various units of content may be stored using one or more of a web application **1124**, an email application service **1126**, an electronic messaging application **1128** (e.g., a texting or messenger application), social networking services **1130** and a directory services repository **1132** (e.g., an AWS® directory service provided by Amazon Web Services, Inc., or any other directory service). A server **1115** may implement a moderator application **1150** for use by moderator-users and a community application **1152** for use by client applications and devices. As an example, server **1115** may be a web server providing the applications **1150**

and **1152** via networks **1110**. As an example, a client computing device may be implemented and/or embodied in a computer device **1105**, a mobile computing device **1106** (e.g., a smart phone), a wearable computing device **1107**, or other computing device. Any of these client computing devices **1105** to **1107** may be configured to transmit content (e.g., as electronic posts) from the store **1116**, and may be configured to receive content (e.g., other electronic posts constituting an online community).

[0108] FIG. 12 illustrates examples of various computing platforms configured to provide various functionalities to components of an electronic message platform **1200** to moderate electronic messages. Computing platform **1200** may be used to implement computer programs, applications, methods, processes, algorithms, or other software, as well as any hardware implementation thereof, to perform the above-described techniques.

[0109] In some cases, computing platform **1200** or any portion (e.g., any structural or functional portion) can be disposed in any device, such as a computing device **1290a**, mobile computing device **1290b**, and/or a processing circuit in association with initiating any of the functionalities described herein, via user interfaces and user interface elements, according to various examples.

[0110] Computing platform **1200** includes a bus **1202** or other communication mechanism for communicating information, which interconnects subsystems and devices, such as processor **1204**, system memory **1206** (e.g., RAM, etc.), storage device **1208** (e.g., ROM, etc.), an in-memory cache (which may be implemented in RAM **1206** or other portions of computing platform **1200**), a communication interface **1213** (e.g., an Ethernet or wireless controller, a Bluetooth controller, NFC logic, etc.) to facilitate communications via a port on communication link **1221** to communicate, for example, with a computing device, including mobile computing and/or communication devices with processors, including database devices (e.g., storage devices configured to store atomized datasets, including, but not limited to triplestores, etc.). Processor **1204** can be implemented as one or more graphics processing units (“GPUs”), as one or more central processing units (“CPUs”), such as those manufactured by Intel® Corporation, or as one or more virtual processors, as well as any combination of CPUs and virtual processors. Computing platform **1200** exchanges data representing inputs and outputs via input-and-output devices **1201**, including, but not limited to, keyboards, mice, audio inputs (e.g., speech-to-text driven devices), user interfaces, displays, monitors, cursors, touch-sensitive displays, touch-sensitive input and outputs (e.g., touch pads), LCD or LED displays, and other I/O-related devices.

[0111] Note that in some examples, input-and-output devices **1201** may be implemented as, or otherwise substituted with, a user interface in a computing device associated with, for example, a user account identifier in accordance with the various examples described herein.

[0112] According to some examples, computing platform **1200** performs specific operations by processor **1204** executing one or more sequences of one or more instructions stored in system memory **1206**, and computing platform **1200** can be implemented in a client-server arrangement, peer-to-peer arrangement, or as any mobile computing device, including smart phones and the like. Such instructions or data may be read into system memory **1206** from another computer readable medium, such as storage device **1208**. In some

examples, hard-wired circuitry may be used in place of or in combination with software instructions for implementation. Instructions may be embedded in software or firmware. The term “computer readable medium” refers to any tangible medium that participates in providing instructions to processor **1204** for execution. Such a medium may take many forms, including but not limited to, non-volatile media and volatile media. Non-volatile media includes, for example, optical or magnetic disks and the like. Volatile media includes dynamic memory, such as system memory **1206**.

[0113] Known forms of computer readable media includes, for example, floppy disk, flexible disk, hard disk, magnetic tape, any other magnetic medium, CD-ROM, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, RAM, PROM, EPROM, FLASH-EPROM, any other memory chip or cartridge, or any other medium from which a computer can access data. Instructions may further be transmitted or received using a transmission medium. The term “transmission medium” may include any tangible or intangible medium that is capable of storing, encoding or carrying instructions for execution by the machine, and includes digital or analog communications signals or other intangible medium to facilitate communication of such instructions. Transmission media includes coaxial cables, copper wire, and fiber optics, including wires that comprise bus **1202** for transmitting a computer data signal.

[0114] In some examples, execution of the sequences of instructions may be performed by computing platform **1200**. According to some examples, computing platform **1200** can be coupled by communication link **1221** (e.g., a wired network, such as LAN, PSTN, or any wireless network, including WiFi of various standards and protocols, Bluetooth®, NFC, Zig-Bee, etc.) to any other processor to perform the sequence of instructions in coordination with (or asynchronous to) one another. Computing platform **1200** may transmit and receive messages, data, and instructions, including program code (e.g., application code) through communication link **1221** and communication interface **1213**. Received program code may be executed by processor **1204** as it is received, and/or stored in memory **1206** or other non-volatile storage for later execution.

[0115] In the example shown, system memory **1206** can include various modules that include executable instructions to implement functionalities described herein. System memory **1206** may include an operating system (“O/S”) **1232**, as well as an application **1236** and/or logic module(s) **1259**. In the example shown in FIG. 12, system memory **1206** may include any number of modules **1259**, any of which, or one or more portions of which, can be configured to facilitate any one or more components of a computing system (e.g., a client computing system, a server computing system, etc.) by implementing one or more functions described herein.

[0116] The structures and/or functions of any of the above-described features can be implemented in software, hardware, firmware, circuitry, or a combination thereof. Note that the structures and constituent elements above, as well as their functionality, may be aggregated with one or more other structures or elements. Alternatively, the elements and their functionality may be subdivided into constituent sub-elements, if any. As software, the above-described techniques may be implemented using various types of programming or formatting languages, frameworks, syntax,

applications, protocols, objects, or techniques. These can be varied and are not limited to the examples or descriptions provided.

[0117] In some embodiments, modules **1259** of FIG. 12, or one or more of their components, or any process or device described herein, can be in communication (e.g., wired or wirelessly) with a mobile device, such as a mobile phone or computing device, or can be disposed therein.

[0118] In some cases, a mobile device, or any networked computing device (not shown) in communication with one or more modules **1259** or one or more of its/their components (or any process or device described herein), can provide at least some of the structures and/or functions of any of the features described herein. As depicted in the above-described figures, the structures and/or functions of any of the above-described features can be implemented in software, hardware, firmware, circuitry, or any combination thereof. Note that the structures and constituent elements above, as well as their functionality, may be aggregated or combined with one or more other structures or elements. Alternatively, the elements and their functionality may be subdivided into constituent sub-elements, if any. As software, at least some of the above-described techniques may be implemented using various types of programming or formatting languages, frameworks, syntax, applications, protocols, objects, or techniques. For example, at least one of the elements depicted in any of the figures can represent one or more algorithms. Or, at least one of the elements can represent a portion of logic including a portion of hardware configured to provide constituent structures and/or functionalities.

[0119] For example, modules **1259** or one or more of its/their components, or any process or device described herein, can be implemented in one or more computing devices (i.e., any mobile computing device, such as a wearable device, such as a hat or headband, or mobile phone, whether worn or carried) that include one or more processors configured to execute one or more algorithms in memory. Thus, at least some of the elements in the above-described figures can represent one or more algorithms. Or, at least one of the elements can represent a portion of logic including a portion of hardware configured to provide constituent structures and/or functionalities. These can be varied and are not limited to the examples or descriptions provided.

[0120] As hardware and/or firmware, the above-described structures and techniques can be implemented using various types of programming or integrated circuit design languages, including hardware description languages, such as any register transfer language (“RTL”) configured to design field-programmable gate arrays (“FPGAs”), application-specific integrated circuits (“ASICs”), multi-chip modules, or any other type of integrated circuit. For example, modules **1259** or one or more of its/their components, or any process or device described herein, can be implemented in one or more computing devices that include one or more circuits. Thus, at least one of the elements in the above-described figures can represent one or more components of hardware. Or, at least one of the elements can represent a portion of logic including a portion of a circuit configured to provide constituent structures and/or functionalities.

[0121] According to some embodiments, the term “circuit” can refer, for example, to any system including a number of components through which current flows to perform one or more functions, the components including

discrete and complex components. Examples of discrete components include transistors, resistors, capacitors, inductors, diodes, and the like, and examples of complex components include memory, processors, analog circuits, digital circuits, and the like, including field-programmable gate arrays (“FPGAs”), application-specific integrated circuits (“ASICs”). Therefore, a circuit can include a system of electronic components and logic components (e.g., logic configured to execute instructions, such that a group of executable instructions of an algorithm, for example, and, thus, is a component of a circuit). According to some embodiments, the term “module” can refer, for example, to an algorithm or a portion thereof, and/or logic implemented in either hardware circuitry or software, or a combination thereof (i.e., a module can be implemented as a circuit). In some embodiments, algorithms and/or the memory in which the algorithms are stored are “components” of a circuit. Thus, the term “circuit” can also refer, for example, to a system of components, including algorithms. These can be varied and are not limited to the examples or descriptions provided.

[0122] Although the foregoing examples have been described in some detail for purposes of clarity of understanding, the above-described inventive techniques are not limited to the details provided. There are many alternative ways of implementing the above-described invention techniques. The disclosed examples are illustrative and not restrictive.

1. (canceled)

2. A method, comprising:

activating a moderator application at a computing platform configured to host a syndication of electronic messages, the moderator application being configured to filter a plurality of queues, each of the plurality of queues having a subset of the electronic messages;

receiving a gesture as a user input signal configured to determine whether to present an electronic message at an interface, whereby the interface is implemented as a portion of a computerized tool in data communication with the moderator application;

characterizing content of the electronic message to determine one or more attributes to form characterized attributes;

analyzing the characterized attributes of the electronic message to identify one or more classification values representative of message attribute data;

arranging the electronic messages relative to other electronic messages to form a queue of electronic messages for moderation as a function of the message attribute data;

receiving one or more gestures configured to cause presentation of a first electronic message at the interface; filtering the queue of the electronic messages with which to facilitate an action automatically of the first electronic message; and

performing an automatic action in accordance with the first electronic message.

3. The method of claim 2, comprising:

providing the interface to include a viewable area bounded by a distance over which one or more images of the electronic messages in which the queue may be accessed, the distance being indicated by the gesture received as the user input.

4. The method of claim 2, further comprises:

detecting positions of each of the electronic messages comprises:

detecting an input associated with a position of at least the first electronic message on the interface, the position being configured to indicate a state that, when the first electronic message is displayed at the position on the interface, activates the automatic action by the moderator application.

5. The method of claim 2, further comprises:

performing the automatic action to approve automatically the first electronic message; and posting the first electronic message to a message network computing system or to a content source computing system.

6. The method of claim 2, wherein determining to perform the automatic action further comprises:

determining not to perform the automatic action.

7. The method of claim 2, further comprising:

detecting accessibility of the first electronic message; and performing an action.

8. The method of claim 2, further comprising:

detecting a position of a second electronic message; and determining to perform a supplemental action.

9. The method of claim 8, further comprising:

re-displaying the second electronic message in the interface to facilitate review of the second electronic message.

10. The method of claim 9, wherein the supplemental action includes one of rejecting the second electronic message, modifying the second electronic message, and forwarding the second electronic message.

11. The method of claim 2, further comprising:

determining if a non-compliant message component is included in at least one of the electronic messages; and modifying the least one of the electronic messages if the non-compliant message component is detected to generate a modified electronic message.

12. The method of claim 2, further comprising:

using the one or more classification values as tags that include metadata.

13. The method of claim 2, further comprising:

characterizing a class of electronic messages to select and predictively apply one or more actions to an unmoderated message.

14. The method of claim 13, further comprising:

implementing any number of statistical analytic programs, machine-learning applications, deep-learning applications, neural networks applications, and natural language processing applications to determine the class of electronic messages.

15. A system, comprising:

a repository configured to store data associated with an electronic message; and

a processor configured to execute programmatic instruction configured to:

activate a moderator application at a computing platform configured to host a syndication of electronic messages, the moderator application being configured to filter a plurality of queues, each of the plurality of queues having a subset of the electronic messages;

receive a gesture as a user input signal configured to determine whether to present an electronic message at an interface, whereby the interface is implemented as a

portion of a computerized tool in data communication with the moderator application;

characterize content of the electronic message to determine one or more attributes to form characterized attributes;

analyze the characterized attributes of the electronic message to identify one or more classification values representative of message attribute data;

arrange the electronic messages relative to other electronic messages to form a queue of electronic messages for moderation as a function of the message attribute data;

receive one or more gestures configured to cause presentation of a first electronic message at the interface;

filter the queue of the electronic messages with which to facilitate an action automatically of the first electronic message; and

perform an automatic action in accordance with the first electronic message.

**16.** The system of claim **15**, wherein the processor is further configured to:

provide the interface to include a viewable area bounded by a distance over which one or more images of the electronic messages in which the queue may be accessed, the distance being indicated by the gesture received as the user input.

**17.** The system of claim **15**, wherein the processor is further configured to:

detect positions of each of the electronic messages comprises:

detect an input associated with a position of at least the first electronic message on the interface, the position being configured to indicate a state that, when the first electronic message is displayed at the position on the interface, activates the automatic action by the moderator application.

**18.** The system of claim **15**, wherein the processor is further configured to:

perform the automatic action to approve automatically the first electronic message; and

post the first electronic message to a message network computing system or to a content source computing system.

**19.** The system of claim **15**, wherein the processor is further configured to:

determine not to perform the automatic action.

**20.** The system of claim **15**, wherein the processor is further configured to:

detect a position of a second electronic message;

determine to perform a supplemental action; and

re-display the second electronic message in the interface to facilitate review of the second electronic message, wherein the supplemental action includes one of rejecting the second electronic message, modifying the second electronic message, and forwarding the second electronic message.

**21.** A non-transitory computer readable medium having one or more computer program instructions configured to perform a method, the method comprising:

activating a moderator application at a computing platform configured to host a syndication of electronic messages, the moderator application being configured to filter a plurality of queues, each of the plurality of queues having a subset of the electronic messages;

receiving a gesture as a user input signal configured to determine whether to present an electronic message at an interface, whereby the interface is implemented as a portion of a computerized tool in data communication with the moderator application;

characterizing content of the electronic message to determine one or more attributes to form characterized attributes;

analyzing the characterized attributes of the electronic message to identify one or more classification values representative of message attribute data;

arranging the electronic messages relative to other electronic messages to form a queue of electronic messages for moderation as a function of the message attribute data;

receiving gestures configured to cause presentation of a first electronic message at the interface;

filtering the queue of the electronic messages with which to facilitate an action automatically of the first electronic message; and

performing an automatic action in accordance with the first electronic message.

\* \* \* \* \*