content server/database **40** are provided to and obtained by the email gateway **12***b*. The publication of the content in the content server **40** by the email gateway **12***b* is one step of a content processing chain that occurs in the gateway. Such a content processing chain is the assembly of a set of content processors with each performing one dedicated operation on the content. The result being a structured format of non-RSS information that can be provided by the email gateway **12***b* to the blog infolet **16**.

[0086] The inbound user request **100** that has been changed by the email gateway **12** into structured non-RSS information contains the URLs for each piece of content published to the content server **40** instead of the actual content submitted by the user. A part of the content is identified by the email gateway **12***b* as being a command line and is sent along in the request (i.e., the structured non-RSS information). Furthermore, the types of content in the request that are published in the content server **40** and the types of content that should be propagated directly to the blog infolet **16** can be configured in the email gateway's content processing chain.

[0087] After taking into account user defined preferences that have been stored in a user profile database **44**, the blog infolet **16** stores a blog information item (e.g., a blog entry) into the blog database **42**. The blog entry contains the URLs that point to the user supplied content now residing on the content server **40**. In some embodiments of the invention, a confirmation of receipt of the blog information item containing an identifier of a new blog entry is provided back to the user from the blog infolet **16**, through the email gateway **12***b* and by the SMTP relay chains **106** and **104**. Conversely, still referring to FIG. **2**A, if a user request, indicating that the blog information item or the entire blog is desired, is sent to the blog infolet **16** in the server platform **10**, then the user preferences are accounted for via the profile database **44**. Then the blog information is retrieved from the blog database **42**. The blog infolet **16** then provides the retrieved blog information, in accordance with the user's service profile and the user session delivery context, to the email gateway **12***b*. The retrieved blog information will contain the appropriate URLs of any data associated with the content at the time of original content submission. Delivery context is the set of all the attributes constraining the content delivery process during a user session. It includes the static device profile, user preferences, dynamic network capacity information, etc.

[0088] Since SMTP messaging requires the actual content to be embedded in the message, the email gateway **12***b* retrieves the associated data from the content server **40** using the URLs while composing the outgoing MIME message. The outgoing MIME message is sent via the SMPT relay chain **106**, **104** to the user.

[0089] FIG. **2**B is an exemplary block diagram of the architecture of SMIL retrieval through the HTTP gateway **12***a*, a blog infolet **16** and associated databases. When a user retrieves content of a blog via a SMIL client **200**, the SMIL document (presentation) returned by an exemplary platform **10** contains, due to SMIL specifics, only the URLs of the requested content. It is the user agent that needs to retrieve the actual content, using the URLs, from the content server **40** (shown via the dotted arrows **202** between the SMIL device and the content server/database).

[0090] SMIL clients **200** can access the platform **10** via the HTTP gateway **12***a* and presently can only be used for retrieval of content. The blog infolet **16** executes a content selection step in this situation because content such as Word documents, PowerPoint presentations, etc. (in general application/*content type) cannot be a part of a SMIL presentation.

[0091] FIG. **2**C is an exemplary block diagram of an architecture of a voice gateway **12***d*, a blog infolet **16** and associated databases. A few things are specific to voice based interactions, although they do not radically change the flow of information. In the context of blogging by voice, whether a user publishes an audio recording or retrieves a blog, all interactions (navigation menus) are presented via the voice telephony platform **300**. When content is being requested by a voice device **302** (e.g., mobile phone, telephone, VoIP phone and the like), the content is provided by the HTTP gateway **12***a* via a VXML file that is interpreted in the voice telephony platform. The voice gateway **12***d* is used only when the blog infolet **16** autonomously decides to push out a phone call to the end user, for example, in the case where a user is subscribed for some events to be automatically delivered by the platform **10** as a notification service **304** (shown by the dotted line).

[0092] The voice telephony platform **300** manages calls via PSTN/Wireless/VoIP networks **304**. The voice telephony platform basically operates as a VXML engine. The voice telephony platform **300** retrieves VXML content, which directs the dialog with the user, via HTTP from HTTP gateway **12***a*. The HTTP gateway **12***a* recognizes the HTTP requests as coming from a telephony platform **300** and creates the appropriate delivery context for the session; most importantly, the HTTP gateway **12***a* requests that the blog infolet **16** presents its functionality through VXML.

[0093] When content (a voice recording) is submitted by the user for publication in, for example, the user's blog, the content will appear to the HTTP gateway **12***a* as a multipart encoded form. Audio recordings can be converted to text using an external resource (ASR) and provided to and stored in the content database **40**, if needed. Similar to other gateways **12**, as part of the content processing, the content is published to the content server **40** and corresponding URLs are sent to the HTTP gateway **12***a*. This is true for all user interactions going through the HTTP gateway.

[0094] An exemplary method for publishing a wired or wireless telephone blog submission may be performed as follows. First, a user dials into a predetermined number in order to request publication of a voice originated blog information. The user may then be authenticated via a voice or user entered PIN. Alternatively, the user's device ID may automatically authenticate the user. A menu, via voice, via a popup screen, or other user interface is provided to the user's voice device **302** by the HTTP gateway **12***a*. Next the user selects the desired option from the menu, for example, "record a voice entry for my blog", and proceeds to speak into the voice device so that the content is recorded into the HTTP gateway **12***a*. The HTTP gateway **12***a* stores the content on the content server/database **40** and in return receives a URL providing the content's location. The blog infolet **16** is informed of the URL for the content published in the content server **40**. And, the blog infolet sends a successful response to the user's voice device **302** via the HTTP gateway **12***a* and voice telephony platform **300**. The user may also be queried to provide descriptors for annotations to the content that will aide in the categorization and searching of the content.