WEB-BASED ELECTRONICALLY SIGNED DOCUMENTS

INVENTORS
Donald G. Peterson
Duane E. Wald
Mike Borozdin

PRIORITY CLAIM

[0001] This application claims the benefit of U.S. Provisional Application Serial No. 61/354,155 filed June 11, 2010, the contents of which are incorporated by reference.

FIELD OF THE INVENTION

[0002] The present disclosure relates to methods and systems for electronic signatures and, more particularly, to methods and systems for managing Web-based electronic signature processes.

BACKGROUND OF THE INVENTION

[0003] In one approach to managing an electronic signature process, a document is emailed or otherwise transferred to a user (a "signer") who engages in a signature

process with respect to the document. Upon receipt, the signer signs the document, after which data from the signed document is transferred to a server or other location for storage.

[0004] The above-described approach suffers from a number of drawbacks. First, common implementations of this approach typically require specialized software, such as Portable Document Format ("PDF") processing software. Moreover, the PDF processing software often needs to be specially configured to facilitate the signature process. Second, PDF and its related processing software may suffer from security vulnerabilities that can be exploited by malicious parties for various purposes, thereby rendering the signature process unreliable. Third, such an approach typically statically defines the document for signature such that the document cannot easily be modified in light of newly received information, particularly when the document has been previously sent to the signer.

SUMMARY OF THE INVENTION

[0005] In one embodiment, a method in an electronic signature service for facilitating electronic signatures includes, without use of a PDF processing module, transmitting a URL or other identifier that identifies a template that specifies required electronic signature data; receiving a request based on the transmitted URL; in response to the received request, preparing a form based on the template; presenting the form within a Web browser; and receiving the required electronic signature data.

[0006] In other embodiments, systems and computer-readable media for facilitating electronic signatures by an electronic signature service are provided.

[0007] In further embodiments, client side methods, systems, and computer-readable media for interacting with an electronic signature service are provided.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Preferred and alternative examples of the present invention are described in detail below with reference to the following drawings:

[0009] FIGURE 1 illustrates an exemplary block diagram of an embodiment of an electronic signature service in accordance with the present invention;

[0010] FIGURE 2 is an exemplary flow diagram of an electronic signature service process performed by an embodiment of the present invention;

[0011] FIGURE 3 is an exemplary flow diagram of an electronic signature client process performed by an embodiment of the present invention;

[0012] FIGURES 4A-4N are user interface screens and controls provided by an exemplary embodiment of the present invention; and

[0013] FIGURE 5 is a block diagram of a computing system for practicing exemplary embodiments.

DETAILED DESCRIPTION

[0014] Embodiments described herein provide enhanced computer- and network-based methods and systems for obtaining and managing electronic signatures and, more particularly, for creating, managing, and utilizing a dynamic Web-based electronic signature process. Example embodiments provide an electronic signature service ("ESS") configured to facilitate creation of a template that specifies required electronic signature data. In one embodiment, the electronic signature service is further configured to provide

an identifier of the template, such as a uniform resource locator ("URL"), a uniform resource identifier ("URI"), a network file identifier or path, or the like. This identifier can be provided to a signer in various ways, such as by being placed on a Web page for access by a Web browser operated by the signer. When the ESS later receives a request based on the provided identifier, the ESS dynamically prepares a form based on the template. This form is transmitted to the signer, who in turn uses it to provide the required electronic signature data to the ESS. The ESS stores the provided electronic signature data for use by other components or parties.

[0015] FIGURE 1 illustrates an exemplary block diagram of an example embodiment of an ESS in accordance with the present invention. In particular, FIGURE 1 illustrates an ESS 100 that includes logic 102, a templates data repository 104 and a signature data repository 106.

[0016] A template creator user 110 uses a client device 120 (*e.g.,* a personal computer, a laptop computer, a smart phone) to create a template. The created template specifies, indicates, or identifies required electronic signature data, such as a signer name, a signature, a date, and the like. Template creation in one example embodiment is described further with reference to FIGURES 4A-4N, below. The created template is stored by the ESS 100 in the templates data repository 104.

[0017] In the illustrated example, the ESS 100 provides a uniform resource locator ("URL") to a Web server 130. The Web server 130 hosts Web pages and/or code modules (*e.g.,* server and/or client side code, servlets, plug-ins, scripts) that facilitate transactions that require electronic signatures. For example, the Web server 130 may host

a human resources Web site that facilitates requests for or related to personal time off, vacation time, sick leave, retirement account contributions, and other personnel matters. In other embodiments, the Web server 130 may host an online shopping facility, an online banking service, a government service, or the like.

[0018] A signer user 111 uses a client device 121 (*e.g.,* a personal computer, a laptop computer, a smart phone) to perform a transaction facilitated by the Web server 130. For example, the signer user 111 may use the client device 121 to submit a request for personal time off to the Web server 130. In response, the Web server transmits the URL that identifies the template to the client device 121. The transmitted URL may be embedded in a Web page as a link or other element (*e.g.,* button).

[0019] The client device 121 then transmits a request based on the received URL to the ESS 100. In one embodiment, the request is transmitted via HTTP, HTTPS, or some other protocol specified by the URL. The transmitted request may include all or some of the URL (*e.g.*, the hierarchical portion of the URL including a host name and path name) as well as data obtained by the client device 121 and/or the Web server 130. For example the client device 121 may initially present a form to gather or obtain data from the signer 111, such as a name, address, employee identification number, or other information. This obtained data may be incorporated or included as part of the transmitted request. In one embodiment, the obtained data is appended onto the URL (or portion thereof) as a query string (*e.g.*, as in an HTTP GET request) or other type of parameters (*e.g.*, embedded in the request as HTTP POST data).

[0020] In response to the request received from the client device 121, the ESS 100 dynamically prepares a form based on the request and/or the template identified by the request. The form may include or otherwise be customized based on additional data appended to the URL or otherwise transmitted along with the received request as discussed above. For example, if the signer's name and address were appended to the URL, then the form may be customized to include this information, thereby saving the signer 111 the steps of having to re-enter this information during the signature process. The prepared form may be, include, or specify user interface controls that are configured to obtain the required signature data from the signer 111. The prepared form is transmitted to the client device 121, where it is presented to the signer 111, such as within the context of a Web browser or other client application. The signer 111 then provides the required signature data via the prepared form to the ESS 100.

[0021] The ESS 100 then stores the received required signature data in the signature data repository 106. The required signature data is then made available for access by other components or users. For example the template creator 110 or some other administrative user (*e.g.,* a human resources administrator) can access the required signature data to verify that the signer 111 has provided the required electronic signature.

[0022] In at least some embodiments, the techniques described herein do not require the use of a Portable Document Format ("PDF") processing module (*e.g.,* viewer, reader, editor) by any of the elements involved in a signature process. In the example of FIGURE 1, neither the client devices 120-121 nor the ESS 100 have or utilize a PDF module to perform any of the described acts or functions. In particular, the client device

121, in obtaining the required signature data does not use a PDF module (*e.g.*, to present and/or fill out a form). Instead, the client device 121 receives and renders the form only via well-known Web standards, such as HTTP, HTML, and the like. By not relying on the PDF format or related processing modules, the described approaches can advantageously be deployed in environments where PDF processing modules may not be available or configured to facilitate form-filling operations or the like, including feature phone or smart phone settings where PDF support may be minimal or entirely lacking.

[0023] Also, the techniques described herein can be employed to overcome or avoid problems related to "stale" or inconsistent forms. In prior approaches, forms may become stale or inconsistent with one another or associated templates because form documents may have long lifetimes that are independent of their associated signature processes and/or templates. For example a form document (*e.g.*, represented as a PDF file) may be transmitted via an email and stored for a period of time (*e.g.*, a week or longer) before the signer signs the form. In the meantime, however, the signature process associated with the form document may have changed (*e.g.,* to require additional information), resulting in an inconsistency between the originally transmitted form and the modified signature process.

[0024] Such problems with "stale" or inconsistent forms are avoided when using the presently described approach. In particular, because the ESS 100 uses a "late binding" approach that dynamically generates a form in response to a received request, modifications to the associated form template can be reflected in the generated form, even when the form template is modified after the URL is initially transmitted. In one

embodiment, after transmitting the URL that identifies a template, the template may be modified, such as to include additional required electronic signature data. Then, in response to a received request, the ESS 100 prepares the form to include controls for receiving the additional required electronic signature data. Thus, even though the template was modified after transmission of the URL, the modified aspects of the template were still reflected in the prepared form transmitted to the client.

[0025] FIGURE 2 is an exemplary flow diagram of an electronic signature service process performed by an embodiment of the present invention. In particular, FIGURE 2 illustrates a process that may be implemented by, for example, one or more elements of the ESS 100, such as the logic 102, described with reference to FIGURE 1. The process provides a dynamically generated form based on a received request that identifies a form template that specifies required electronic signature data.

[0026] The process begins at block 202, where it transmits a URL that identifies a template that specifies required electronic signature data. Typically, the transmitted URL is then embedded in a Web page or other document (*e.g.,* email) that can be accessed on a client device for purposes of performing a transaction that requires an electronic signature.

[0027] At block 204, the process receives a request based on the transmitted URL. The received request may include one or more arguments or other data along with at least a portion of the transmitted URL. For example, a script or other code executing on a client device may have obtained information about a signer user, and included that

information as arguments in the request, such as represented by a URL query string of one or more key value pairs.

[0028] At block 206, the process prepares a form based on the template and the received request. Preparing the form may include dynamically generating a form based on the template, where the form is populated to include arguments or other data that are part of the request received at block 204. The form may be represented, for example, as markup code (*e.g.,* HTML code) and/or instructions (*e.g.,* JavaScript instructions).

[0029] At block 208, the process presents the form. Presenting the form may include transmitting the form to a client device that is executing a Web browser. In other embodiments, other types of client applications may be utilized, such as a mobile application (*e.g.,* an "app") executing on a smart phone. The Web browser then renders the transmitted form and receives the required signature data as user inputs to the form.

[0030] At block 210, the process receives the required electronic signature data. For example, the process may receive the required electronic signature data from a Web browser that is executing on a client device and that has rendered the form transmitted as described with respect to block 208. The Web browser may transmit the form data to the process in response to a user selection, such as selection of a "submit" button or other control. The received required electronic signature data may be stored or in some cases transmitted to one or more recipients specified by the template or another mechanism.

[0031] The process may perform other operations in addition to or instead of those described above. For example, in one embodiment, the process does not utilize, directly or indirectly, a PDF processing module during any of its operations. As another

example, in some embodiments, the process may enforce usage restrictions specified by templates. For example, a usage restriction may specify a maximum number of times or frequency that a template can be used for electronic signature purposes. If the maximum number is exceeded, the process may refuse to prepare and transmit a form.

[0032] FIGURE 3 is an exemplary flow diagram of an electronic signature client process performed by an embodiment of the present invention. In particular, FIGURE 3 illustrates a process that may be implemented by or within a Web browser or other code module executing on the client device 121 described with respect to FIGURE 1. The process interacts with an electronic signature service process (*e.g.,* FIGURE 2) in order to implement an electronic signature process.

[0033] The process begins at block 302, where it receives a URL that identifies a template managed or stored by an electronic signature service. The template specifies electronic signature data that is required to represent a complete electronic signature. In other embodiments, other mechanisms for identifying the template may be used, such as network file system file or resource identifiers.

[0034] At block 304, the process transmits a request based on the received URL. In some embodiments, the request is dynamically generated by the process based on the received URL and includes at least a portion of the received URL as well as other data or arguments. For example, the request may be generated by a script executing on a Web browser that has collected data values from a user. In generating the request, the script may incorporate the collected data values into the request, such as by appending them to the URL as a query string comprising one or more key value pairs. The request may be

transmitted to the electronic signature service (*e.g.,* using the HTTP protocol) in order to obtain a form for collecting required electronic signature data.

[0035] At block 306, the process receives a form prepared based on the template. As discussed above, the electronic signature service prepares a form based on the transmitted request. The prepared form is then transmitted from the electronic signature service to the illustrated process.

[0036] At block 308, the process collects required electronic signature data by presenting the received form. For example if the form is represented as HTML, the process may render the form in a Web browser or similar rendering component. Alternatively, the form may be represented as a code module (*e.g.,* an applet or script). In such a case, the process presents the received form by executing or interpreting the code module. A user of the process can then use the presented form to input the required electronic signature data.

[0037] At block 310, the process transmits the collected electronic signature data to the electronic signature service. As discussed above, the electronic signature service then stores the electronic signature data for use or access by other users or system components.

WEB POWERFORMS EMBODIMENT

[0038] The following describes an exemplary embodiment referred to herein as "Web PowerForms." Web PowerForms facilitates the creation of an online-signing process, seamlessly integrated into a Web portal, intranet, email, document or other

technology using only, in an embodiment, user interface tools. No programming is required to create this integrated experience.

[0039] Web PowerForms provides a user the ability to click a URL link, be placed in a form collecting their name and email address, and then be immediately placed in a Web user interface to complete a document signing. Web PowerForms further includes the ability for the technology to dynamically add the appropriate recipient information, place these data in the document to be signed, and present a customized, integrated document signing.

[0040] When the user clicks the URL link, a "virtual sender" initiates sending of a pre-described document. This allows the on-demand creation of a customized document for signature that is virtually prepared and sent in an automated fashion by the pre-specified sender.

[0041] Document signing can be directly embedded in a system that maintains the URL link. Also, in Web PowerForms, a template indicating parties, roles and order of document recipients describes the signing process. Furthermore, progress and status of a signing workflow is visible to the "virtual sender" from an online- or desktop-based console.

[0042] On completion of a document signing, results (*e.g.,* signature data) are available for download by the virtual sender. Results and completed documents are also visible to the virtual sender via the console. Completed documents may be sent via email attachment to all parties.

EXAMPLE PROCESS ACCORDING TO AN EMBODIMENT

STEP 1: CREATE A TEMPLATE

[0043] As illustrated in FIGURE 4A, this step specifies the following: 1) the recipients of the document including the user initiating the document signing, 2) the signing order, 3) the subject and message, 4) security options and requirements, and 5) required data, signatures, initials, dates and other data to be collected in the signing process.

STEP 2: CREATE A WEB POWERFORM FROM THE TEMPLATE

[0044] As illustrated in FIGURE 4B, the template created in Step 1 is identified as the instructions for a Web PowerForm. The virtual sender is identified, security options are specified and the method for signing is indicated. At the end of this step the Web PowerForm exists for signing.

STEP 3: GET LINK FROM POWERFORM LIBRARY

[0045] As illustrated in FIGURES 4C and 4D, this step creates the URL link that can start the Web PowerForm signing process. The URL extracted in this step can be inserted into a portal, intranet, email, document or other technology capable of launching a Web URL.

STEP 4: PUBLISH THE LINK FOR USERS

[0046] The URL to the Web PowerForm is added to a portal, intranet, email, document or other technology, such as is illustrated in FIGURE 4E, which the user can access to start the signing process.

STEP 5: USER CLICKS THE LINK AND ACTIVATES THE WEB POWERFORM

[0047] As illustrated in FIGURE 4F, clicking the Web PowerForm link from the portal, intranet, email, document or other technology launches the signing process. The first step of the signing experience is to identify the user and any other required, un-addressed participants by supplying a name and email address.

STEP 6: USER COMPLETES AND SIGNS THE WEB POWERFORM

[0048] As illustrated in FIGURE 4G, once the user and required parties are identified, the user is presented with the document, customized with the recipients, data and other content specified in the template in Step 1. The user adopts a signature, reviews the document, supplies the requested data, adds any required signatures and completes the signing.

STEP 7: DATA IS RETURNED TO THE ELECTRONIC SIGNATURE SERVICE AND MADE AVAILABLE FOR DOWNLOAD

[0049] After completion, all signing data, document and signing history are made available for download from the Web PowerForm Library. These data can be downloaded on a document basis or for all documents submitted on a specific Web PowerForm.

[0050] Completed documents are delivered to all parties as attachments to the completion notification email as well as being made available in the console.

WEB BASED POWERFORMS

OVERVIEW

[0051] PowerForms allows users to create "self-help" transactions that do not require a send from an electronic signature service, such as may be provided by DocuSign. This is optionally advantageous for account openings, Web-based forms and documents sent via email. An embodiment does not require the user to have known

software products, such as Adobe Acrobat® Professional, and to know this software which is complex.

[0052] With Web PowerForms, PowerForms are made much more flexible by allowing a PowerForm not to have a PDF that launches it. By converting a stored Template to a PowerForm, the service generates some extra information about how the PowerForm is to launch, etc., and the service can launch it with a URL. (This is similar to a PDF version, but the link is not to a PDF that starts the process, but rather it is directly to a template structure.)

[0053] The impact of this approach is to enable much more rapid deployment of Web-based signing for account openings, and other files, reducing cost and complexity for the customer, and removing the need for products such as Adobe Acrobat®.

FEATURES

[0054] Web PowerForms make a standard Template behave like a PowerForm. In fact, the term PowerForm may be used to describe all PDF-based or Web-based PowerForms.

ATTRIBUTES OF A WEB POWERFORM:

[0055] 1) Can be invoked by a signer via a URL. This URL may be sent in an email, posted on a Web site, or embedded into a PDF document. The URL may be overloaded with data to populate the PowerForm fields and to populate role information for recipients

[0056] 2) Clicking the URL creates a new transaction and opens a secure browser to begin the signing process.

[0057] 3) The signer who invokes the PowerForm may provide any additional recipient data. In other words, if there are five signers in the Template workflow, the signer invoking the PowerForm may default to being the first signer. They may provide information about who the other recipients are as part of their signing. This can occur up front, and may only happen if there is more than one undefined role. (This is different than the current PDF PowerForm.)

[0058] 4) Arguments may be able to be passed into the PowerForm to provide information to complete form fields or assign roles. This allows an integration to collect HTML form information and generate a string with arguments to pre-populate the PowerForm (*e.g.,* name, email for each role, field values, access code, ID check, phone auth). (This is different than the way the current PDF PowerForms work.)

[0059] 5) In general, in some prior approaches, templates are uploaded; there is no way to grab a template from a template library, etc. An embodiment is able to default to obtaining templates from the template library, not via uploading. If a user has the template in the library, the user can make a PowerForm from it. If the template PDF has form fields, when the user creates a PowerForm from it, the user can elect to use them or not by selecting PDF or Web as the type of PowerForm he is creating. When the service creates a PowerForm, the option to create "PDF" or "Web" PowerForm can direct how it is constructed. If "PDF" is selected, the PowerForm can have the hyperlink to begin signing in a PDF process, and the PDF data fields can be usable. If "Web" is selected, then it can go directly to the Web, and NOT use the PDF to collect data.

[0060] 6) Attributes that can be edited:

[0061] a) Template. Opens the template for both PDF and Web PowerForms.

[0062] b) Sender. Assign who the PowerForm is "owned" by—who it can report status back to, etc.

[0063] c) Usage settings. Maximum number of times it can be used and number remaining.

[0064] d) Frequency. How often it can be used; e.g., limit to every X minutes by the same signer.

[0065] e) Status. Active or inactive. If inactive, the link clicked may say "This document has been suspended and is not currently active. Please contact the sender."

[0066] f) Access Mode. Settings are Direct and Email. NOTE: When using Direct, the template may have some other form of authentication enabled such as access code, phone auth, or ID check to ensure the signer is identified. When the link is clicked, the PowerForm opens immediately and processes as follows:

[0067] (1) Ask for any other recipients. If they are required, the recipients may be provided (*e.g.,* name, email)

[0068] (2) Execute any authentication requested

[0069] (3) Signing Process

[0070] (4) Once complete, go to the URL for completed envelopes. An embodiment may add the URL for the completed signing event to be defined as part of the template.

[0071] h) Recipients. Optional edit of recipients that may also be accomplished in other "edit" modes. Some embodiments may display routing order, roles, etc.

[0072] i) PDF owner Password. This protects the ability to edit the PDF in the case of a PDF PowerForm.

[0073] j) Document name

[0074] k) Email Subject

[0075] l) Email Text

[0076] m) An embodiment may include Instructions. These instructions may be used on the starting page for the signer. If instructions are entered, this may be the "Introduction" shown after the signer clicks the URL. This may enable more hand-holding for the signer, and in the event the template has more than one role, it may be displayed on the same page as where the roles are asked for. This way, a user can define some text, such as "Welcome to Envoy Mortgage. You are applying for a loan. If you will be co-signing, please enter the name & email of the co-signer as well as your name and email."

[0077] n) If the Template contains Required Fields with the name "NAME" and "EMAIL" for the First Role, then the service may not ask the user to provide their name and email up front. The service can instead do it with the template.

[0078] 7) Data Management

[0079] a) It may be possible to download all the data collected by a PowerForm by all its uses.

[0080] b) It may be possible to filter the download all the data by date range.

[0081] c) It may be possible to delete data collected by the PowerForm, either all data, or by date range.

[0082] d) In a PDF PowerForm, data from all PDF fields may be gathered. In a Web PowerForm all data from SecureFields may be gathered. If a PDF has both PDF and Web SecureFields, all may be gathered.

[0083] e) Performance. It may be able to download 1000 records of 100 fields in 10 seconds. Other embodiments have different performance characteristics.

[0084] f) Format of CSV file. In an embodiment, each PowerForm may have only one record/row in a CSV file vs. each having duplicate rows.

[0085] g) PowerForm subject, signer name, signer email, date/time completed (or none if not complete), all field names in the PDF/SecureFields. For each signer the value they entered or the word "NA" if not assigned to that person.

[0086] 8) Console listing

[0087] a) The console listing PowerForms may have the following columns. They may be sortable, date searchable, and searchable by date used, sender, name, signing mode, status, type.

[0088] i) Name (Template name). Some embodiments compress "name" and "template" into one

[0089] ii) Sender. Identity of who it is assigned to

[0090] iii) Auth. Direct, email

[0091] iv) Used. The number of records collected by the PDF. (This value changes if the user deletes records.)

[0092] v) Remaining. The number left before the limit (if set) is hit

[0093] vi) Last used date

[0094] vii) Status (active, inactive)

[0095] viii) TYPE: PDF or Web

[0096] b) Act on a PowerForm

[0097] i) In the console the user may be able to use buttons or menus to do the following:

[0098] (1) Get form data from one selected form (download data dialog)

[0099] (2) Download

[00100] (3) Email. Creates email with hyperlink

[00101] (4) Links. Displays the hyperlink for the PowerForm

[00102] (5) Edit. Opens edit view

[00103] (6) Template. Opens the template to be edited

[00104] c) Create a new PowerForm

[00105] i) Not all templates are PowerForms. To make a PowerForm, the user selects a template, and clicks "create PowerForm" menu/button

[00106] (1) This opens the "edit" panel for a PF, and allows the user to assign the "PowerForm attributes"

[00107] (2) PDF or Web Form

[00108] (3) Mode

[00109] (4) Owner

[00110] (5) Usage settings

[00111] (6) Displays list of roles, and which are required or not

[00112] (7) Email subject and Name (for subsequent signers)

[00113] ii) Difference between Template and PowerForm

[00114] (1) PowerForms have more attributes than Templates

[00115] (2) Mode, Times uses, multiple data collection, etc.

[00116] Plan mappings and settings

[00117] 1) According to an embodiment:

[00118] a) Available to Standard and Enterprise Accounts

[00119] b) Feature enabled as a plan item in the Billing Plan UI and the PlanItemsMaster table.

[00120] c) Service Admin can enable PowerForms

[00121] d) User Admin can enable PowerForms, *e.g.,* some users can use PowerForms, some can author them.

[00122] e) Users can turn on/off PowerForms

[00123] 2) Launch for existing customers: normally this may be off

[00124] 3) Launch for new customers: normally this may be on.

DOCUSIGN POWERFORMS

[00125] DocuSign® PowerForms embody at least some of the described techniques and combine the capabilities of PDF forms and market-leading DocuSign electronic signing and workflow technology. DocuSign provides an example electronic

signature service, sometimes referred to as the "DocuSign System." Forms that were previously completed, signed and returned in hard copy via fax or overnight delivery can be widely distributed, conveniently completed and securely signed and returned electronically.

[00126] DocuSign PowerForms add these capabilities to existing editable PDFs:

[00127] • PowerForms can be distributed via email or the Web with a unique, secure URL automatically generated by the DocuSign System

[00128] • All data provided via the PDF form may be returned and preserved within the DocuSign system and can be readily integrated into existing applications

[00129] • Centralized creation and management of PowerForms, including control of all workflow elements—authentications, approvals and routing order.

[00130] DocuSign PowerForms work in conjunction with the DocuSign System to allow a user to create transactions that do not require the user to send documents from DocuSign.

[00131] In an embodiment, there are two types of PowerForm users: PowerForm Senders and PowerForm Administrators.

[00132] **PowerForm Senders:** PowerForm senders are allowed access to the PowerForms library and can download and send PowerForms that are assigned to as the Sender. Senders can also see completed, signed PowerForms that they sent, whether by email or other.

[00133] **PowerForm Administrators:** PowerForm administrators have the same access as PowerForm senders, but can also create and edit PowerForms. Additionally,

PowerForm administrators can extract the completed form data set, in XML or CSV format, from completed PowerForms.

CREATING A POWERFORM

[00134] In an embodiment, only PowerForm Administrators can create new PowerForms. If a user is not a PowerForm Administrator, he or she may not use this section.

CREATING A NEW POWERFORM

[00135] FIGURE 4H shows a console configured to facilitate the creation of a new PowerForm template.

[00136] **1.** The first step in creating a PowerForm is creating a DocuSign template. Follow the normal steps for creating a template. Templates that may be used for PowerForms preferably include only one document, although they can have any number of pages. When creating a template a user can use a PDF form created (or one that has been received from a third party) that has active form fields. The form fields are automatically converted to SecureFields when the PDF is added to a template. Once the DocuSign template is created, the PowerForm can now be created.

[00137] **2.** Go to the Manage PowerForms page PowerForm tab in one of the following manners: (a) In the DocuSign Console, click **PowerForms** and then click **New PowerForm**. The user is taken to the PowerForm tab on the Manage PowerForms page. (b) In the DocuSign Console, click **PowerForms** and then click **PowerForms Library**. The user is taken to the Manage PowerForms page. Click the New PowerForm link. (c) In

the DocuSign Console, click **PowerForms** and then click **PowerForms Library**. The user is taken to the Manage PowerForms page. Click the PowerForm tab.

[00138] **3. Upload** the DocuSign template:

[00139] **4. Click Template Library** and locate the DocuSign template.

[00140] **5.** To use a DocuSign Professional template, click the "here" link, click **Browse...** and then locate the DocuSign template. DocuSign Professional templates have a file extension of .dpd and are located, by default, in My Documents\DocuSign Professional\Templates. In some embodiments, only one template may be uploaded per PowerForm. **Note:** If a template used by a PowerForm is edited and saved at a later time, the changes may be automatically reflected in the PowerForm. Once a template is uploaded, the Recipients, Document Settings and Sender sections may be filled out with the information from the template.

[00141] **6.** Review template information. The information in the Document Settings and Sender section can be edited, but the information in the Recipients may be read-only and may be not editable. Any necessary changes to the recipients may be made in the DocuSign template.

[00142] The Recipients section, shown in FIGURE 4I, shows the routing order, recipient role or name, email address, recipient type (signer, certified delivery, carbon copy etc.), any access code associated with the recipient and if there is a security check (ID Check, Phone Authentication or False for no check) for the recipient.

[00143] If any of this information is incorrect, the user can correct the template by clicking the template name link to open the template wizard, as illustrated in FIGURE 4J. After saving the changes to the template, the user may be returned to the PowerForm tab.

[00144] **7.** Review Sender information. The default sender for a PowerForm may be the PowerForm Administrator creating the PowerForm. The sender associated with the PowerForm may be the person that is notified by email when a PowerForm is completed. The sender may be also the person who sees the PowerForm-generated envelopes in the Sent Items folder in their DocuSign Console.

[00145] To **change** the PowerForm sender, click the Change link to the right of the sender name, as illustrated in FIGURE 4K. The list of all DocuSign senders associated with the account appears. Select the sender and **click Select** to continue.

[00146] **Note:** In one **embodiment**, only a single sender may be associated with the PowerForm. If multiple senders need to be associated with a PowerForm, create multiple copies of the PowerForm (one copy for each sender) and associate the sender with the PowerForm.

[00147] **8.** Set the Usage Settings, as illustrated in FIGURE 4L:

[00148] • **Maximum number of times the document can be used:** The PowerForm administrator can **specify** the number of times the PowerForm can be used (signed). This can limit an account's exposure to excessive envelopes usage by specifying a maximum number of times the PowerForm can be used. A PowerForm administrator can edit this number anytime after the PowerForm is created, even if it has reached a previously entered use limit. Each time a recipient clicks **Begin Signing** for a document,

the number of Uses remaining may be reduced by one. If a recipient attempts to sign a PowerForm that has reached its maximum number of uses, an error message may be displayed to the recipient saying the document is not available for signing.

[00149] • **Frequency the same signer can use the document:** The PowerForm administrator can specify how often the same recipient can sign the same PowerForm. This may be accomplished by specifying a length of time between signings. For example, if a PowerForm is configured to limit use to every 365 days, then the same recipient may only be able to sign the PowerForm once every year. If the recipient attempts to sign the PowerForm more frequently than allowed, an error message may be displayed to the recipient saying the document is not available for signing.

[00150] • **Document Status:** This sets whether a document can be sent to recipients. The default status for a PowerForm may be **Active**, meaning the PowerForm can be sent and signed by recipients. If the status is changed to **Inactive**, then the PowerForm may not be emailed or accessed by a recipient (even if they are entering the PowerForm URL or HTML link). If a recipient attempts to sign an inactive PowerForm, an error message may be displayed to the recipient saying the document is not active and to contact the sender.

[00151] **Signing Mode:** In an embodiment, PowerForms supports two types of signing modes: **Email** and **Direct**.

[00152] The **Email** signing mode may be used to verify the recipient's identity using email authentication before they can sign a document. After a recipient enters their email and clicks **Begin Signing**, an email with a validation code for the PowerForm may

be sent to the recipient. If the recipient did not provide a valid email address, they may not be able to open and sign the document.

[00153] The **Email** signing mode can also be used if someone other than the recipient will be completing the PowerForm. For example, a call center representative completes the PowerForm while they are on the phone with the signer and provides the signer's name and email address in the PowerForm. When the representative clicks **Begin Signing**, the recipient receives an email notification to sign the document and they are not able to edit the information entered into the PowerForm by the representative.

[00154] The **Direct** signing mode does not require any verification. After a recipient enters their email and clicks **Begin Signing**, a new browser opens and the recipient can immediately begin the signing process.

[00155] Since the recipient's identity is not verified using email authentication, it is strongly recommended that the **Direct** signing mode only be used when the PowerForm is accessible behind a secure portal where the recipient's identity is already authenticated or another form of authentication (for example: access code, phone authentication or ID check) is specified for the recipient in the DocuSign template.

[00156] **9.** Review and set the Document Settings, as illustrated in FIGURE 4M:

- **Document Name:** This may be a required field. This is the name of the PowerForm. The default name of the PowerForm may be the DocuSign template name.

- **Email Subject:** This may be a required field. This is the email subject for any emails sent with this PowerForm. The default email subject may be the Message subject from the DocuSign template.

- **Email Text:** This may be a required field. And text is provided for the PowerForm. This text may be used in emails sent to signers who are notified via email.

• **Signing Instructions:** These may be additional instructions that are shown on the starting page for the recipient. These instructions are optionally advantageous if the recipient accesses the PowerForm by a method other than email. If instructions are entered, they may be shown as an introduction after the recipient accesses the PowerForm.

[00157] **10.** Save the PowerForm. After adding the DocuSign template and making adjustments to the settings for the PowerForm, click **Save** to save the PowerForm. The user is then returned to the **PowerForms** Library tab and the new PowerForm is listed in the PowerForms Library. To exit the PowerForm tab without saving the PowerForm, click the PowerForm Library tab.

SENDING POWERFORMS

[00158] In an embodiment, there are multiple ways that a PowerForm can be sent to a recipient, but sending process typically begins by viewing the PowerForms Library.

[00159] PowerForm administrators can see all PowerForms for all senders in their account. PowerForm Senders may only see the PowerForms for which they are the sender. This is a form of access control that restricts PowerForm Senders to only the PowerForms they are allowed to use.

[00160] The PowerForms in the library can be sorted by clicking on the column headings. PowerForms can also be searched based on various criteria, for example, by date last used/signed, document name, sender name, signing mode, or status.

[00161] PowerForms can be sent by email or by using a URL link or HTML settings or by setting them up embedded in a Web page.

SENDING BY EMAIL

[00162] **1.** In the DocuSign Console, click **PowerForms** and then click **PowerForms Library**. The user is taken to the Manage PowerForms page PowerForms Library tab.

[00163] **2.** Select the PowerForm to send by clicking the check box adjacent to the PowerForm name.

[00164] **3.** Click **Email**, a new email is opened using a default email client. The Email Subject and Text from the PowerForm is automatically entered in the email and the URL link to the PowerForm is added at the end of the email text. The user can review and edit the subject text for the email.

[00165] **4.** Send the email.

SENDING BY ATTACHMENT

[00166] This is similar to sending a PowerForm by email, but in this case the PowerForm is attached to an email rather than being linked from it. **Note:** In an embodiment, this method may only be used if the PowerForm type is PDF.

[00167] **1.** In the DocuSign Console, click **PowerForms** and then click **PowerForms Library**. The user is taken to the Manage PowerForms page PowerForms Library tab.

[00168] **2.** Select the PowerForm to send as an attachment by clicking the check box adjacent to the PowerForm name.

[00169] **3.** Click **Download** and save the PowerForm to the user's system.

[00170] **4.** Create a new email using an email client and attach the PowerForm to the email.

[00171] **5.** Fill out and send the email normally.

SENDING BY URL

[00172] **1.** In the DocuSign Console, click **PowerForms** and then click **PowerForms Library**. The user is taken to the Manage PowerForms page PowerForms Library tab.

[00173] **2.** Select the PowerForm to send by clicking the check box adjacent to the PowerForm name.

[00174] **3.** Click **Links**, a dialog box is displayed with a URL link to the PowerForm and HTML snippet. The user can copy and paste the URL in an email or the user can copy the HTML snippet and add it to a Web page.

POWERFORMS LIBRARY

[00175] The PowerForms Library provides a common location for storing and accessing PowerForms.

[00176] PowerForm administrators can see all PowerForms for all senders in their account. PowerForm Senders may only see the PowerForms for which they are the sender. This is a form of access control that restricts PowerForm Senders to only the PowerForms they are allowed to use.

VIEWING THE POWERFORM LIBRARY

[00177] When a PowerForm administrator and PowerForms users view their PowerForms Library, they may see the following information for each PowerForm:

- Name (Template)—the PowerForm name and template name
- Sender—the sender associated with the PowerForm
- Auth—the Signing Mode (Email/Direct) for the PowerForm

- Used—the number of times the PowerForm has been used
- Remaining—the remaining number of times the PowerForm can be used
- Last Used—the date the PowerForm was last sent or signed
- Status—if the PowerForm is Active or Inactive
- Type—the type of document in the PowerForm: PDF indicates the source of the PowerForm is a PDF file with active form fields, Web indicates the source of the PowerForm is a DocuSign template.

[00178] The PowerForms Library list can be sorted by clicking on the column headings. The PowerForms Library can also be searched by date last used/signed, document name, sender name, signing mode, or status.

RETRIEVING FORM DATA

[00179] If the PowerForm has been signed, a PowerForm administrator can download the form data for the PowerForm, as follows:

[00180] **1.** From the PowerForms Library tab, select the PowerForm to download data from by clicking the check box adjacent to the PowerForm name and click **Form Data** the Form Data dialog box appears.

[00181] **2.** Optionally, filter by selecting a From and To date range. After entering or selecting the date range, click **Apply Filter** to apply the date range filter, as shown in FIGURE 4N.

[00182] **3.** Click **Download XML** or **Download CSV** to download the information as an XML or CSV file.

EDITING A POWERFORM

[00183] PowerForm administrators can edit a PowerForm at any time. When editing a PowerForm the PowerForm administrator can replace or edit the DocuSign

template associated with the PowerForm, change the document and usage settings and change the sender for the PowerForm.

[00184] **1.** From the PowerForms Library tab, Select the PowerForm to edit by clicking the check box adjacent to the PowerForm name and click **Edit.** The PowerForm tab with the PowerForm information is displayed.

[00185] **2.** Edit the PowerForm information in the same manner as creating a new PowerForm.

[00186] **3.** Click **Save** to save changes.

REPLACING A POWERFORM TEMPLATE

[00187] If the PowerForms administrator wants to update the DocuSign template associated with the PowerForm, they can do so by clicking on the template file name and downloading the template to their local machine. They can then edit the template in DocuSign Professional and then upload the new template.

[00188] **1.** From the PowerForms Library tab, select the PowerForm to replace the template for by clicking the check box adjacent to the PowerForm name (the user can select multiple PowerForms in the list) and click **Replace Template** the Template Selection dialog box appears.

[00189] **2.** Select a new template for the PowerForm and click **Save**.

[00190] **3.** Click **Save** to save the change.

[00191] When a new DocuSign template is uploaded for the PowerForm, all signers who have an older version of the PowerForm will be presented with the updated PDF document to sign and will be processed based on the updated template (*e.g.*,

modified authentication methods, routing orders, etc.). Note: If the underlying PDF document is replaced, the PowerForms administrator making the change may need to make sure the new version has the same fields (and field names) as the old version or the PowerForm may fail.

COPYING A POWERFORM

[00192] A PowerForm administrator can create multiple copies of a PowerForm for the purposes of assigning the PowerForm to different senders. A unique copy of the PowerForm will be created for each sender.

[00193] **1.** From the PowerForms Library tab, select the PowerForm to copy by clicking the check box adjacent to the PowerForm name (the user can select multiple PowerForms in the list) and click **Copy,** the list of all DocuSign senders associated with the account appears.

[00194] **2.** Select the sender and click **Select** to continue.

[00195] **3.** Click **Save** to save the change.

DELETING A POWERFORM

[00196] PowerForm administrators can delete PowerForms from the PowerForms Library. However, any data associated with the PowerForm may be lost when the PowerForm is deleted. If the user wants to retain the PowerForm data, the user can retrieve and save the data or just set the PowerForm status to Inactive.

[00197] **1.** From the PowerForms Library tab, select the PowerForm to delete by clicking the check box adjacent to the PowerForm name (the user can select multiple

PowerForms in the list) and click **Delete,** the system asks for confirmation of the deletion of the PowerForm.

[00198] **2.** Click **OK** to delete the PowerForm or **Cancel** to cancel the action.

EMBEDDING A POWERFORM IN A WEB PAGE

[00199] The user can select the appropriate PowerForm within the library and click on the "Links" button. A dialog will be displayed with a URL link to the PowerForm and HTML snippet. The user can copy and paste the URL in an email or the user can copy the HTML snippet and add it to a Web page.

EXAMPLE COMPUTING SYSTEM

[00200] FIGURE 5 is a block diagram of a computing system for practicing example embodiments of an electronic signature service 100. In the embodiment shown, the computing system 500 comprises a computer memory ("memory") 501, a display 502, one or more Central Processing Units ("CPU") 503, Input/Output ("I/O") devices 504 (*e.g.,* audio processor, video processor, keyboard, mouse, CRT or LCD display cards or drivers, and the like), other computer-readable media 505, and network connections 506.

[00201] The electronic signature service 100 is shown residing in memory 501. The electronic signature service 100 interacts with client devices 120-121 and the Web server 130 via a network 550. In other embodiments, some portion of the contents and some of or all of the components of the electronic signature service 100 may be stored on and/or transmitted over the other computer-readable media 505. The components of the electronic signature service 100 preferably execute on one or more CPUs 503 to manage electronic signature processes, as described herein. Other code or programs 530 (*e.g.,* a

Web browser or server, and the like) and potentially other data repositories, such as data repository 520, also reside in the memory 501, and preferably execute on one or more CPUs 503. Of note, one or more of the components in FIGURE 5 may not be present in any specific implementation. For example, some embodiments may not provide other computer-readable media 505 or a display 502.

[00202] The memory 501 also includes a user interface manager 511 and an electronic signature service application program interface ("API") 512. The user interface ("UI") manager 511 provides a view and a controller that facilitate user interaction with the electronic signature service 100 and its various components. For example, the user interface manager 511 provides interactive graphical user interface elements such as those described with respect to FIGURES 4A-4N. As discussed, such user interfaces allow users to create and manage templates that represent electronic signature processes.

[00203] The API 512 provides programmatic access to one or more functions of the electronic signature service 100. For example, the API 512 may provide a programmatic interface to one or more functions of the electronic signature service 100 that may be invoked by one of the other programs 530 or some other module. In this manner, the API 512 facilitates the development of third-party software, such as user interfaces, plug-ins, news feeds, adapters (*e.g.,* for integrating functions of the electronic signature service 100 into Web or mobile applications), and the like. In addition, the API 512 may be in at least some embodiments invoked or otherwise accessed via remote entities, such as by code executing on a remote client device, to access various functions of the electronic signature service 100. For example, an application on the client device

121 may upload signature data via the API 512. The API 512 may also be configured to provide code modules that can be integrated into third-party applications and that are configured to interact with the electronic signature service 100 to make at least some of the described functionality available within the context of other applications.

[00204] In an example embodiment, components/modules of the electronic signature service 100 are implemented using standard programming techniques. For example, the electronic signature service 100 may be implemented as a "native" executable running on the CPU 503, along with one or more static or dynamic libraries. In other embodiments, the electronic signature service 100 may be implemented as instructions processed by a virtual machine that executes as one of the other programs 530. In general, a range of programming languages known in the art may be employed for implementing such example embodiments.

[00205] In addition, the embodiments described above may also be structured in various ways, including but not limited to, multiprogramming, multithreading, client-server, or peer-to-peer, running on one or more computer systems each having one or more CPUs. Some embodiments may execute concurrently and asynchronously, and communicate using message passing, pipes, signals, or other communication techniques. Also, other functions could be implemented and/or performed by each component/module, and in different orders, and by different components/modules, yet still achieve the described techniques.

[00206] Furthermore, in some embodiments, some or all of the components of the electronic signature service 100 may be implemented or provided in other manners, such

as at least partially in firmware and/or hardware, including, but not limited to one or more application-specific integrated circuits ("ASICs"), standard integrated circuits, controllers (*e.g.,* by executing appropriate instructions, and including microcontrollers and/or embedded controllers), field-programmable gate arrays ("FPGAs"), complex programmable logic devices ("CPLDs"), and the like. Some or all of the system components and/or data structures may also be non-transitorily stored as contents (*e.g.,* as executable or other machine-readable software instructions or structured data) on a computer-readable medium (*e.g.,* as a hard disk; a memory; a computer network or cellular wireless network or other data transmission medium; or a portable media article to be read by an appropriate drive or via an appropriate connection, such as a DVD or flash memory device) so as to enable or configure the computer-readable medium and/or one or more associated computing systems or devices to execute or otherwise use or provide the contents to perform at least some of the described techniques. Some or all of the system components and data structures may also be stored as data signals (*e.g.,* by being encoded as part of a carrier wave or included as part of an analog or digital propagated signal) on a variety of computer-readable transmission mediums, which are then transmitted, including across wireless-based and wired/cable-based mediums, and may take a variety of forms (*e.g.,* as part of a single or multiplexed analog signal, or as multiple discrete digital packets or frames). Such computer program products may also take other forms in other embodiments. Accordingly, embodiments of this disclosure may be practiced with other computer system configurations.

[00207] Embodiments of the invention are operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

[00208] Embodiments of the invention may include or be implemented in a variety of computer-readable media. Computer-readable media can be any available media that can be accessed by a computer and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer-readable media may comprise computer storage media and communication media. Computer storage media include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can accessed by computer. Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a

modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of the any of the above should also be included within the scope of computer-readable media.

[00209] According to one or more embodiments, the combination of software or computer-executable instructions with a computer-readable medium results in the creation of a machine or apparatus. Similarly, the execution of software or computer-executable instructions by a processing device results in the creation of a machine or apparatus, which may be distinguishable from the processing device, itself, according to an embodiment.

[00210] Correspondingly, it is to be understood that a computer-readable medium is transformed by storing software or computer-executable instructions thereon. Likewise, a processing device is transformed in the course of executing software or computer-executable instructions. Additionally, it is to be understood that a first set of data input to a processing device during, or otherwise in association with, the execution of software or computer-executable instructions by the processing device is transformed into a second set of data as a consequence of such execution. This second data set may subsequently be stored, displayed, or otherwise communicated. Such transformation, alluded to in each of the above examples, may be a consequence of, or otherwise involve, the physical

alteration of portions of a computer-readable medium. Such transformation, alluded to in each of the above examples, may also be a consequence of, or otherwise involve, the physical alteration of, for example, the states of registers and/or counters associated with a processing device during execution of software or computer-executable instructions by the processing device.

[00211] Embodiments of the invention may include or otherwise be implemented according to elements described in co-pending and commonly owned U.S. Appl. No. 12/176,265 filed July 18, 2008, which is hereby incorporated by reference as if fully set forth herein.

[00212] This patent application is intended to describe one or more embodiments of the present invention. It is to be understood that the use of absolute terms, such as "must," "will," and the like, as well as specific quantities, is to be construed as being applicable to one or more of such embodiments, but not necessarily to all such embodiments. As such, embodiments of the invention may omit, or include a modification of, one or more features or functionalities described in the context of such absolute terms.

[00213] While the preferred embodiment of the invention has been illustrated and described, as noted above, many changes can be made without departing from the spirit and scope of the invention. Accordingly, the scope of the invention is not limited by the disclosure of the preferred embodiment.

CLAIMS

1. A method in an electronic signature service, comprising:

without use of a Portable Document Format processing module,

creating a template that specifies required electronic signature data;

transmitting a URL that identifies the template;

receiving a request based on the transmitted URL;

in response to the received request, preparing a form based on the template;

presenting the form within a Web browser; and

receiving the required electronic signature data.

2. The method of claim 1 wherein the received request includes one or more arguments along with a portion of the transmitted URL, and wherein preparing the form includes populating the form with the included arguments.

3. The method of claim 1 wherein the request is based on a URL that is dynamically generated by a module executing on a client device that executes the Web browser.

4. The method of claim 3 wherein the module executing on the client device is a script executed by the Web browser.

5. The method of claim 1 wherein the request is based on a URL that includes argument values received from data collection performed by a client device that executes the Web browser.

6. The method of claim 1, further comprising:

after transmitting the URL that identifies the template, modifying the template to include additional required electronic signature data; and

in response to the received request, preparing the form to include controls for receiving the additional required electronic signature data.

7. A system comprising:

an electronic signature service computer configured, without use of a Portable Document Format processing module, to:

transmit a URL that identifies a template that specifies required electronic signature data;

receive a request based on the transmitted URL;

in response to the received request, prepare a form based on the template;

present the form within a Web browser; and

receive the required electronic signature data.

8. The system of claim 7 wherein the transmitted URL is included in a Web site accessed by the Web browser.

9. The system of claim 7, wherein the electronic signature service computer is further configured to transmit the received electronic signature data to one or more recipients specified by the template.

10. The system of claim 7, wherein the template specifies usage restrictions, and wherein the electronic signature service computer is further configured to enforce the usage restrictions.

11. The system of claim 10, wherein the usage restrictions specify a maximum number of times the template can be used for electronic signature purposes.

12. The system of claim 7, further comprising:

a client device that executes the Web browser and that is configured, without use of a Portable Document Format processing module, to:

receive the transmitted URL;

dynamically generate the request based on the transmitted URL, by including one or more arguments in the generated request; and

transmit the request to the electronic signature service computer.

13. The system of claim 12, wherein the client device is further configured to execute a module in the Web browser that dynamically generates the request based on the transmitted URL.

14. The system of claim 13, wherein the module is a script executing in the Web browser.

15. The system of claim 12 wherein the client device is further configured to:

collect data values from a user of the client device; and

include the collected data values as the one or more arguments in the generated request.

16. A computer-readable storage medium that includes instructions configured, when executed by a client device, to perform a method for interacting with an electronic signature service, the method comprising:

without use of a Portable Document Format processing module,

receiving a URL that identifies a template that specifies required electronic signature data;

transmitting a request based on the received URL;

receiving a form prepared based on the template;

collecting the required electronic signature data by presenting the received form in a Web browser executing on the client device; and

transmitting the collected data to the electronic signature service.

17. The computer-readable storage medium of claim 16, wherein method further comprises executing a module in the Web browser that dynamically generates the request based on the transmitted URL.

18. The computer-readable storage medium of claim 17 wherein the module is a script executing in the Web browser.

19. The computer-readable storage medium of claim 16 wherein the method further comprises:

collecting data values from a user of the client device; and

including the collected data values as the one or more arguments in the generated request.