an electronic postmark. In some cases, this is done by hashing the voter information. The ballot is then mailed to the voter, who casts his or her votes, records the votes onto the blockchain via the app or website and optionally mails the ballot back. The election officials can tally the vote using the electronic or paper ballot received from the voter.

[0100] FIGS. **6-13** depict various data flow diagrams of some exemplary operations in various embodiments of the vote by mail system.

[0101] FIG. **6** displays a message flow diagram demonstrating an embodiment of how a voter could register with the blockchain powered voting system **100**. Voter **601** inputs a desired user name, password, and/or other information into user interface **131**. Voter **601** can also include a "secret," for example the answer to a security question, such as the name of first pet that can also be used to identify the voter. In some embodiments, voter **601** can also input information necessary to verify the identity of the voter, such as a driver's license number, social security number, or address.

[0102] User interface **131** sends that information to the blockchain API **602**, which then forwards the information on to identity manager **603**. Identity manager **603** can then request that verification authorities **605** verify the identity of voter **601**. In some embodiments, verification authorities **605** are government entities such as the Social Security Administration, state motor vehicle departments, the Federal Bureau of Investigation, and the United States Postal Service that can use the submitted information to verify the identity of voter **601**. In some embodiments, the blockchain API **602**, the identity manager **603**, and other components of the system can be a part of or embodied in the blockchain access layer **101** or other component of the systems described herein.

[0103] Once verification authorities **605** have verified the identity of voter **601**, the verification authorities send the results to identity manger **603**. If verification authorities **605** confirm the identity of the voter **601**, identity manager **603** will generate a unique user identifier (UUID) for the voter **601** and create a public/private key pair for the voter **601**. The public/private key pair will later be used to sign transactions on the blockchain. Once the keys have been generated, identity manger **603** sends the keys for storage into key store **604**. In some embodiments, key store **604** can also store the user IDs and secrets of voters, as well as digital tokens that can be used to identify voters instead of user names and secrets. Finally, identity manager **603** generates a UUID and a hash of the user's password and sends the information to the blockchain API **602**. Blockchain API **602** then sends the unique user identifier and private key back to the voter **601**.

[0104] FIG. **7** depicts a message flow diagram demonstrating an embodiment of how a voter could log on to the voting systems of the current disclosure. In some embodiments, voter **601** will log in to the system by submitting his or her UUID, password, and secret to the user interface **131**, which will then forward it on to blockchain API **602**. In some embodiments, blockchain API **602** then requests that identity manager **603** verifies the user ID and secret. In some embodiments, the identity manager **603** then verifies the user ID secret or token with key store **604**, which then responds back with the results to identity manger **603**, which then forwards the results to blockchain API **602**. Blockchain API **602** then returns a success or error message back the voter **601**.

[0105] FIG. **8** depicts a message flow diagram demonstrating an embodiment of how a voter can request and fill out an ballot with the system. First the voter **601** submits its specific identification number, e.g. ID1, to the identity manager **603** and has its identity verified by identity manager **603** in the manner previously described. The voter **601** then requests an ballot from user interface **131**, which then returns the ballot. The voter then fills out/updates the ballot using user interface **131** before submitting the ballot to blockchain API **602**. Blockchain API **602** then confirms that the voter exists within the registered voter database service **142** and receives a success or failure message back. If the attempt at confirmation is successful, the blockchain API then stores the submitted ballot onto the blockchain. In some embodiments, the blockchain API stores the submitted ballot on to the blockchain by updating the information associated with the specific voter ID, e.g. ID1. In some embodiments, the voter-ballot database **154** can also determine if multiple votes were received from a particular voter and note this on the blockchain as well. The voter-ballot database **154** can then determine whether actions should be taken to deal with the multiple votes, such as marking the voter for a fraud review or determining which of the votes to count.

[0106] FIG. **9** depicts a message flow diagram demonstrating how a voter can request to register for an election. First, user interface **131** displays a list of elections for voter **601**. Voter **601** then submits a request to register for an ballot to blockchain API **602** through user interface **131**. In some embodiments, this request contains a specific user ID (ID1) and a unique alphanumeric election ID (E1). The blockchain API **602** then registers the voter's request to blockchain **801**. In some embodiments, the blockchain API **602** registers the user ID and election ID to the blockchain **801**. In some embodiments, blockchain API **602** also transmits the user ID and election ID to the United States Postal Service **904** or other entity that will eventually be responsible for mailing the physical ballot. Next, blockchain **801** sends acknowledgement back to blockchain API **602** which forwards it on to voter **601** through user interface **131**. Blockchain API **602** then notifies registrar **903** that the voter **601** has registered for an election. In some embodiments, this notification contains the user ID and election ID. Next, registrar **903** then informs the blockchain API **602** that the user is verified and approved for the election through user interface **131**. In some embodiments, this message includes the relevant user ID and election ID. The blockchain API **602** then stores this approval onto blockchain API **602**. Blockchain API **602** then generates a token for the voter **601**. In some embodiments, the token is generated by the user ID, authorization, and election ID. In some embodiments, the blockchain API **602** generates the token in conjunction with a random alphanumeric sequence issued by token engine **901**. The blockchain API **602** stores the association between the token (T1), the user id (ID 1), and the election id (E1) in vault **902**, such as a vault database **153** or similar structure. In some embodiments, the blockchain API **602** sends the information to the United States Postal Service **904** or other entity that will eventually be responsible for mailing the physical ballot. In some embodiments, the United States Postal Service **904** or other entity can use this information to generate a barcode, EPM®, or other computer or machine readable identifier that will be printed on the paper ballot that will allow the user to later scan the ballot. In some embodiments, the barcode or other computer or machine readable identifier is