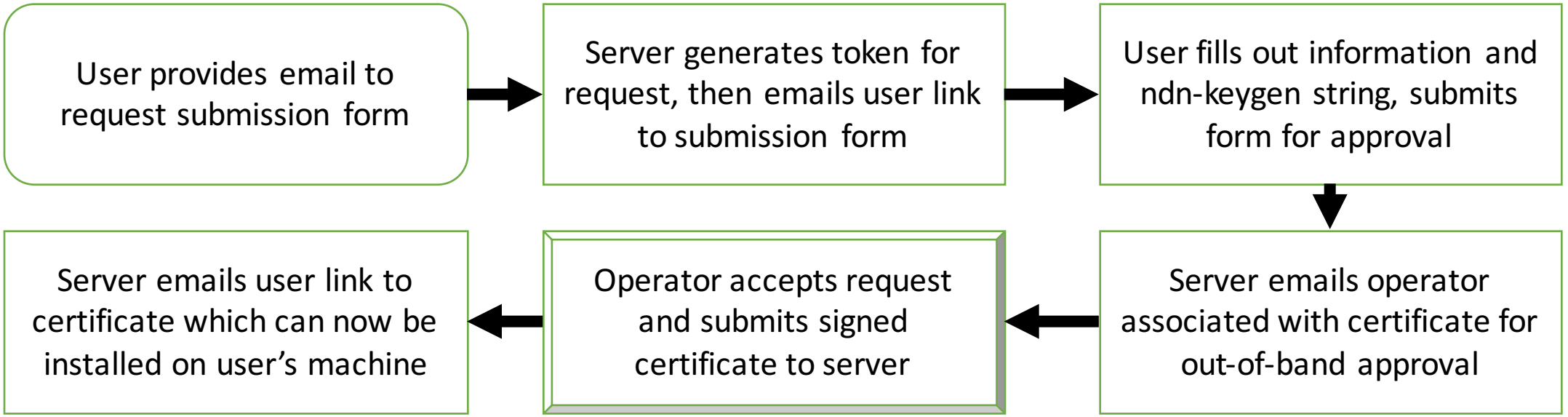
Proposal for NDN Automatic Certificate Issuance & Maintenance

Issuance/Maintenance Requirements

1. Provide support for short-lived certificates:
   - Limits data exposure if keys are compromised.
   - Certificates need to be automatically re-signed.
2. Utilize the NDN network when possible:
   - Users need certificates to access the network, so issuance is done over IP.
   - Renewal of existing certificates must use NDN.
3. Issuance and maintenance will mirror ACME protocols if possible:
   - ACME is an existing system to manage certificates.
   - NDN is inherently different from IP, so ACME/validation channels may be different depending on implementation.

Current Workflow

The current workflow is secured by verifying the email of the user. After the operator issues a certificate, the user requests this certificate from the server for installation. No further interaction between the user and server occurs.



Overview of ACME Workflow

- The ACME workflow has three major sections:
  I. Account Creation
     1. Client registers an identifier(s) with the server, providing contact information signed with an asymmetric key pair. (All further messages are signed with this key pair, referred to as the 'account key')
     2. The server replies with challenges for the client, such as email verification. Challenges differ based on the type of identifier. A webserver might be asked to provision a file.
     3. The client responds with a list of completed challenges, and the server validates them.
     4. The client polls the server until it replies with an authorization or rejection.
  II. Certificate Issuance
     I. Client sends a CSR to the server signed by the account key, which the server replies to with a signed certificate.
  III. Certificate Revocation
     I. Client sends a revocation request to the server signed by the account key. The server stops signing this certificate in the future and replies with confirmation of the revocation.

Certificate Accounts

- Every user agent has an associated account, represented by a generated public/private key pair:
  - A user may own distinct identities for each CA system (i.e. /ndn, /mhealth).
  - Every identity is associated with the same account.
  - We assume that the account key is secure and will not be compromised.
- Accounts are bound to the certificates and stored in the SignatureInfo field of the NDN Certificate:
  - Used by the CA to verify identity before renewing a certificate.
  - *May store other meta-info about account in SignatureInfo (TBD).*

Certificate Request Process

- Certificates are requested through a provided ‘user agent’ program running on the device requesting a name:
  - Users will specify a module when running the program, which corresponds to the test-bed they are requesting the name under.
  - Modules will provide information specific to the CA system, such as (1) location of the CA, (2) namespace constraints, (3) supported challenges.
- The process mirrors the ACME protocol:
  1. Users sends a certificate request to the CA for a name.
  2. CA responds with challenges to verify user owns the identity.
  3. User completes the challenge out-of-band, and informs the CA.
  4. The CA validates the challenges and returns a signed certificate.

Certificate Request Format

- A certificate request is generated by the module, and consists of:
  1. Requested namespace (i.e. for /ndn, this might be forced to user email)
  2. Certificate public key
  3. Account public key
  4. Challenges (chosen out of a set provided by the module)
  5. Version # of module (if out-of-date, the CA will reject the request)
- The request is encrypted with the public key of the CA (we assume this is known by the module):
  - This is required to protect against man-in-the-middle attacks; the CA has no way to authenticate the requester.

Response to Certificate Request

- The response to a certificate request:
  1. Requested namespace
  2. Challenge information
- The response is encrypted with the account key to verify that the user owns the account private key.
- Depending on the type of challenge, the ‘challenge information’ field will contain different information:
  - If the challenge is an email, this field might be empty as the token is sent over email (separate channel).
  - If the challenge is modifying a DNS entry, this field will contain the record that they must add to the DNS.
  - Future challenges may use this field in other ways (TBD).

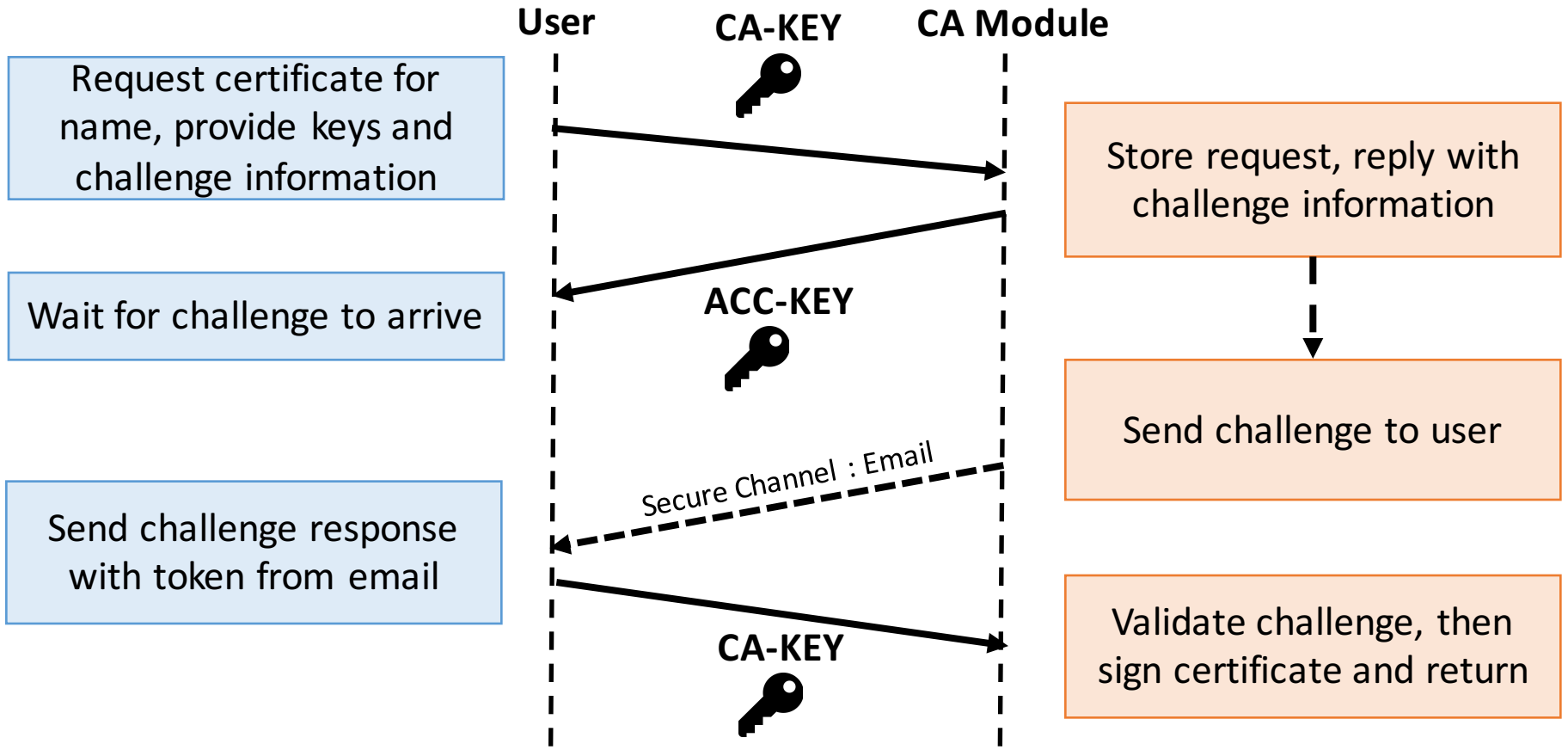
Challenge Response

- After completing the challenge, the user agent program will send a message to the CA to begin validation.
- Format of the challenge response:
  1. Requested namespace
  2. Token/secret key
- The challenge response is again encrypted with the CA's public key.
- The token/secret key field in the challenge response will vary depending on what challenges are issued:
  - If the challenge was an email, it will contain a token that must be provided here.
  - If the challenge was a DNS record, this field may be empty and will instead signal to the CA that it must probe the network to check for this record.

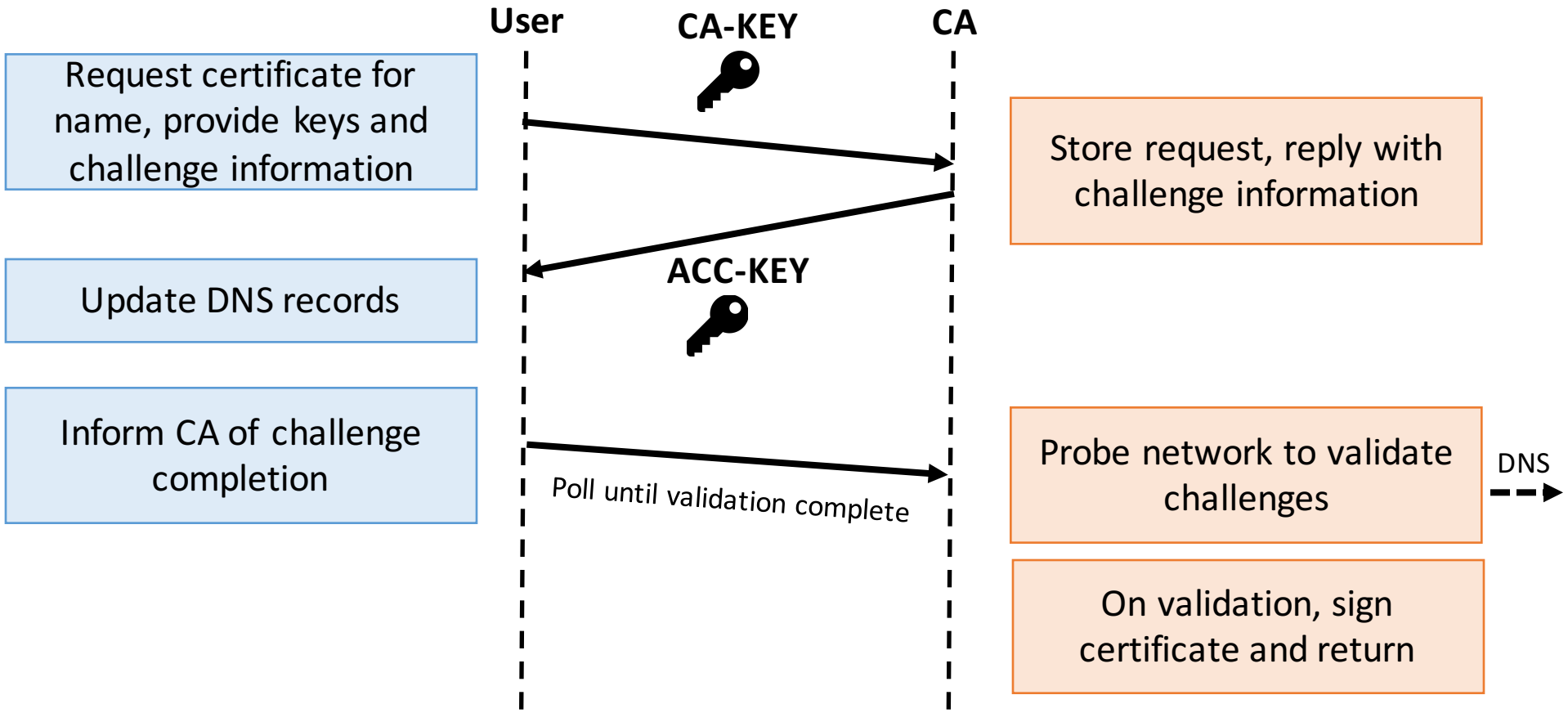
Certificate Signing

- The CA will first validate the challenge response before signing.
  - Validation can either be instant (if the token is provided) or may take time (if the CA needs to probe the network to validate).
- The format of the certificate response:
  1. Signed certificate, bound with account key.
  2. Name of certificate on CA system for future acquisition.
  3. Rollover deadline (discussed in later slide)
- If validation will take time, the CA will instead respond with a ‘pending validation’ and the user program will poll until it is complete.
- The name provided by the certificate response will be the name to which the CA will publish renewed certificates.
- Validated certificates will be stored by the CA and will automatically be re-signed until the rollover deadline.

Message Flow (Email Challenge)



Message Flow (DNS Challenge)



Certificate Renewal/Revocation

- Certificates are kept alive by a negative evidence scheme:
  - The CA automatically re-signs certificates when they are about to expire.
  - Revocations are (mostly) explicit, and require a revocation request.
- The CA will renew a certificate if:
  - The certificate is still valid and no revocation request has been made.
  - The certificate's rollover deadline has not passed.
- Simple revocation request format, signed with account key:
  - Name of certificate

Certificate Rollover

- All CA's will enforce key rollovers for certificates:
  - For security, the certificate key itself ought to be refreshed from time to time, but can be longer-lived than the certificate validity period.
  - The CA will require that a new certificate key be provided within some period after the initial certificate has been issued (i.e. 6 months).
  - If the user does not request a rollover, the certificate will not be re-signed.
- Within the allotted period, a rollover request must be made with:
  - Name of certificate
  - New certificate key (*maybe signed with old certificate key?)*
- The rollover request itself is signed with the user's account key.
- No revalidation is required, the new certificate is created and the rollover period is updated automatically.