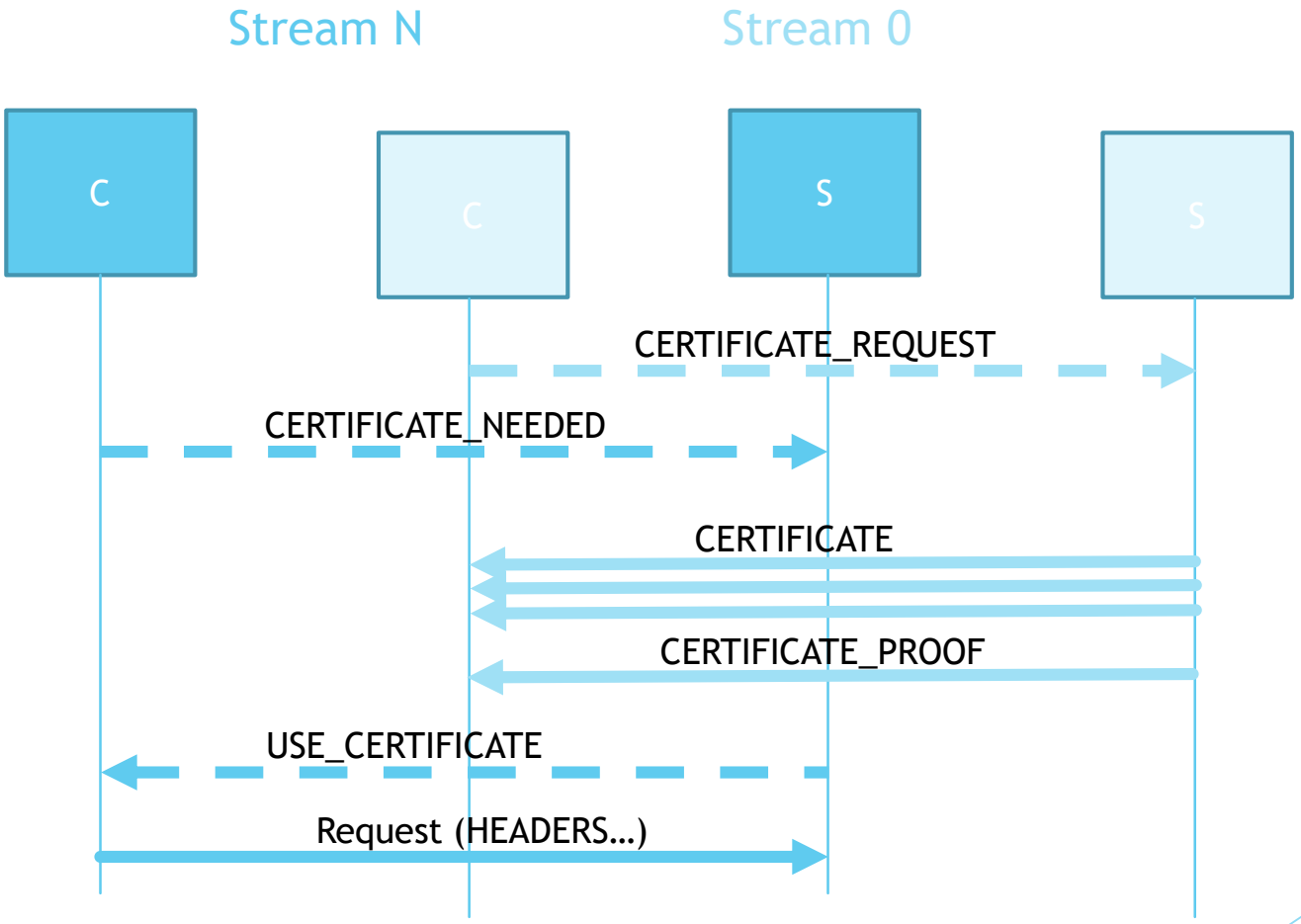
Secondary Certificates

…in HTTP?

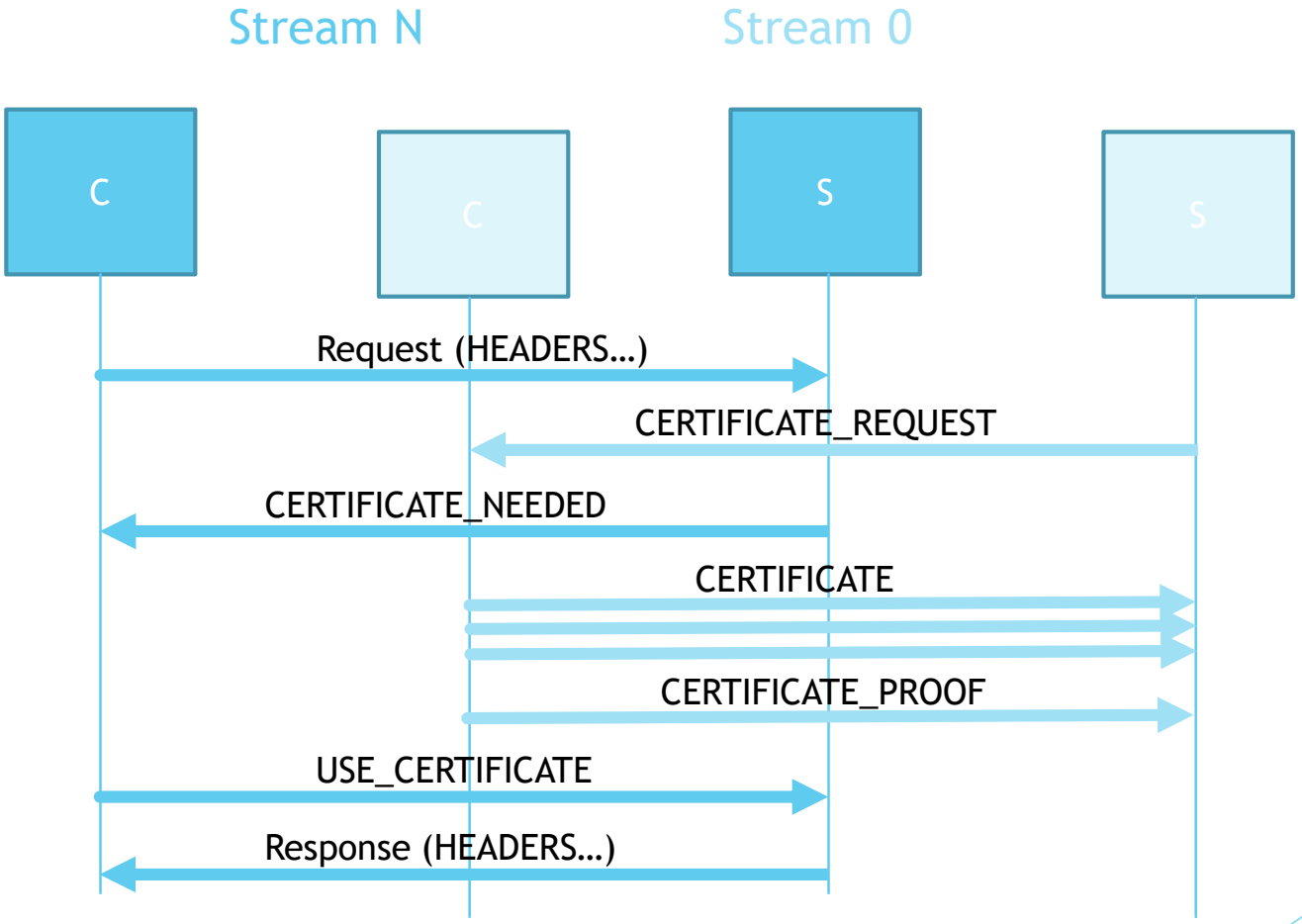
Why is HTTP doing certs?

- TLS: One server identity, one client identity
  - HTTP/2 multiplexing – different client identities for different resources
- Client certs
  - HTTP/2 prohibits renegotiation
  - Most TLS 1.2 implementations can't do renegotiation while application data flows
    - Spec doesn't mandate this, but is deployment reality
  - TLS 1.3 *might* improve this
- Multiplexing
  - HTTP/2 connection coalescing currently only works if the server cert has all possible names
    - Forces servers to use mega-certs with large numbers of SANs
  - Desire to support coalescing across origins while using discrete certificates

Server Certificate



Client Certificate



But again, why in HTTP?

- Could replace Stream 0 exchange with TLS 1.3 Post-Handshake Auth
  - PHA would require more capabilities
    - Already capable of exchanging a client certificate; multiple?
    - Could this be used to exchange additional server certificates?
  - HTTP on-stream frames look about the same
  - HTTP layer needs to retrieve identifiers for the exchanges to reference
- What about TLS 1.2?
  - Backport something?
  - Leave alone; carrot to migrate to 1.3