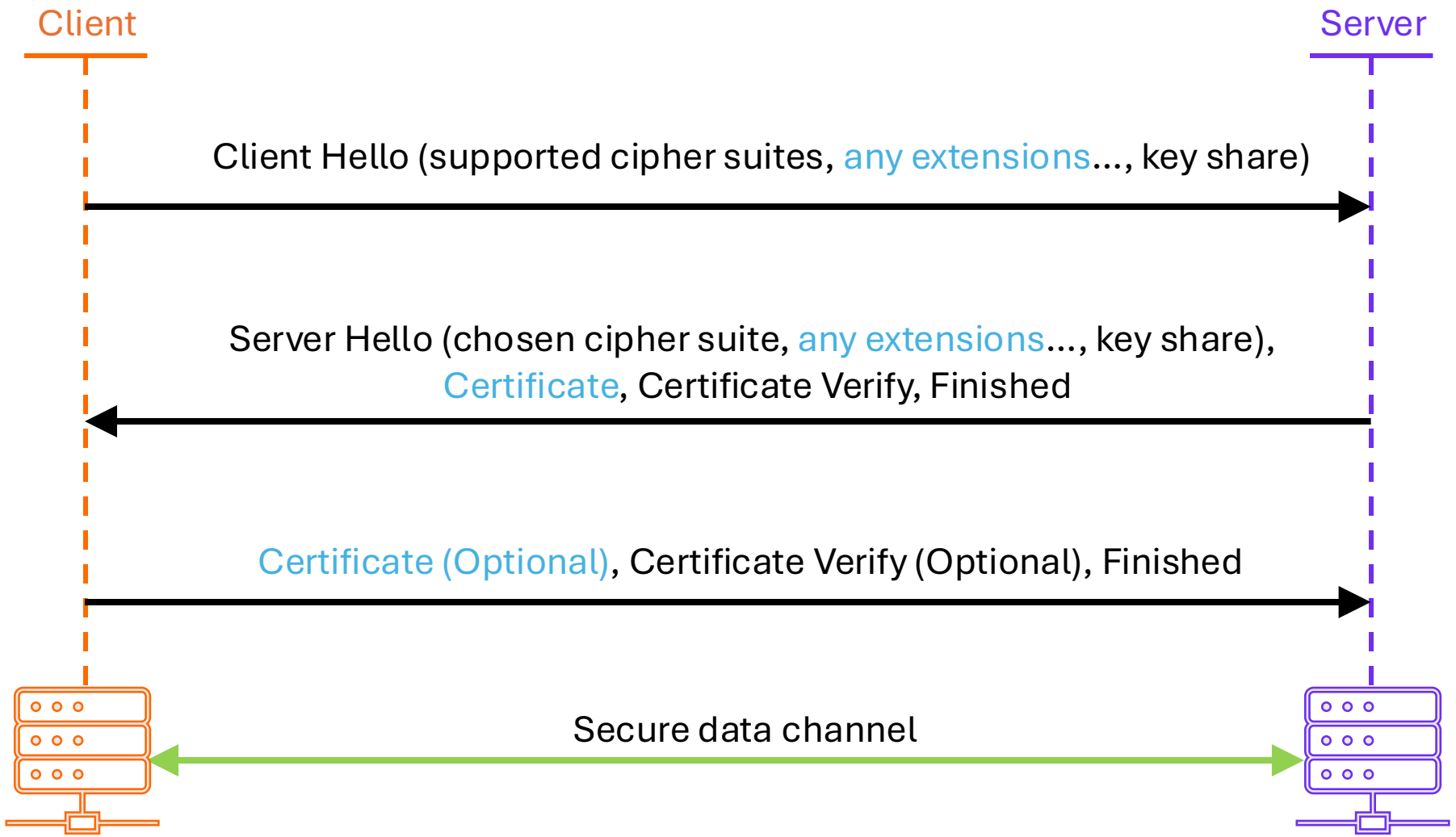
Remote attestation + TLS

Project update

Why Attested TLS (aTLS)?

- Remote Attestation (RA) is a powerful tool in understanding the security state of a workload
  - … but cannot be used on its own to secure comms with the workload.
- Secure channel establishment can bootstrap a comm channel to a known network identity
  - … but cannot probe its current state.
- If we want to know the state of our secure channel peer, we need to integrate the two mechanisms.
- The TLS handshake is the most widely used secure channel establishment protocol, so the prime target for enablement work.

TLS handshake overview



* Extension points

Our goals

- Follow established best-practices.
  - e.g., negotiation of features and parameters.
- Ensure secure integration between TLS and RA.
- Enable as many topologies as possible
  - Background check & passport.
  - Client and/or server as attester.
- No modifications to core TLS protocol.
  - Exclusively use predefined extension points.

... and non-goals

- No new RA primitives or APIs
- Not designed for web use-cases
- Not intended as a universal solution

Desirable properties

- **Negotiation of RA parameters**
  - TLS Client and Server must be able to signal willingness to engage in RA.
  - Relying Party (RP) must be able to convey a nonce.
- **Freshness of attestation evidence**: for background check deployments, it should be possible to prove the freshness of evidence (e.g., via a nonce).
- **Refresh of attestation credentials**: RA can track runtime state changes, so being able to check credentials repeatedly is important.
- **Cryptographic binding between TLS and RA sessions**: there must be some way to verify that the endpoint of the secure channel is the same entity represented by the attestation credential.

Design space

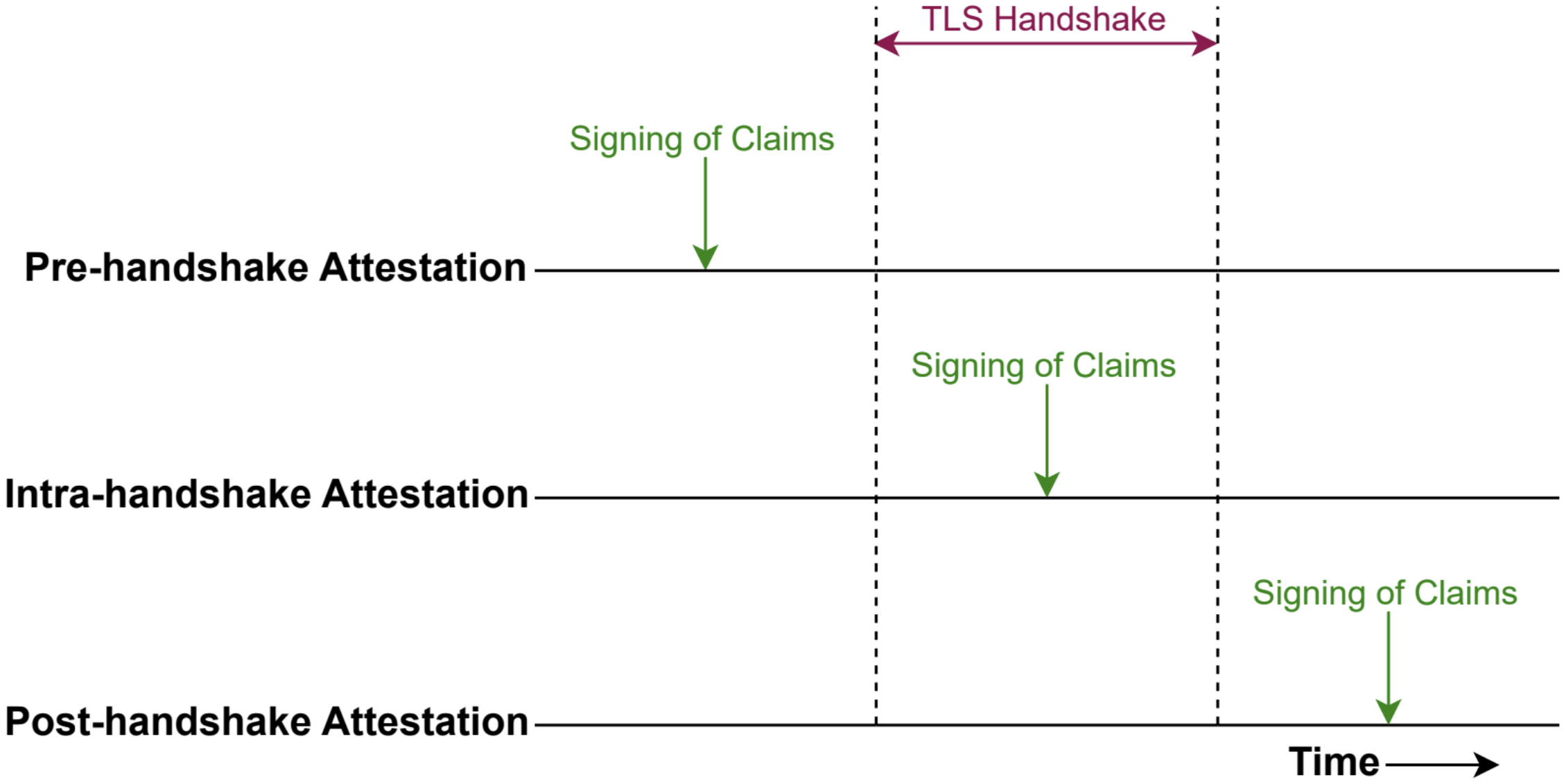


Diagram courtesy of Muhammad Usama Sardar

Existing designs

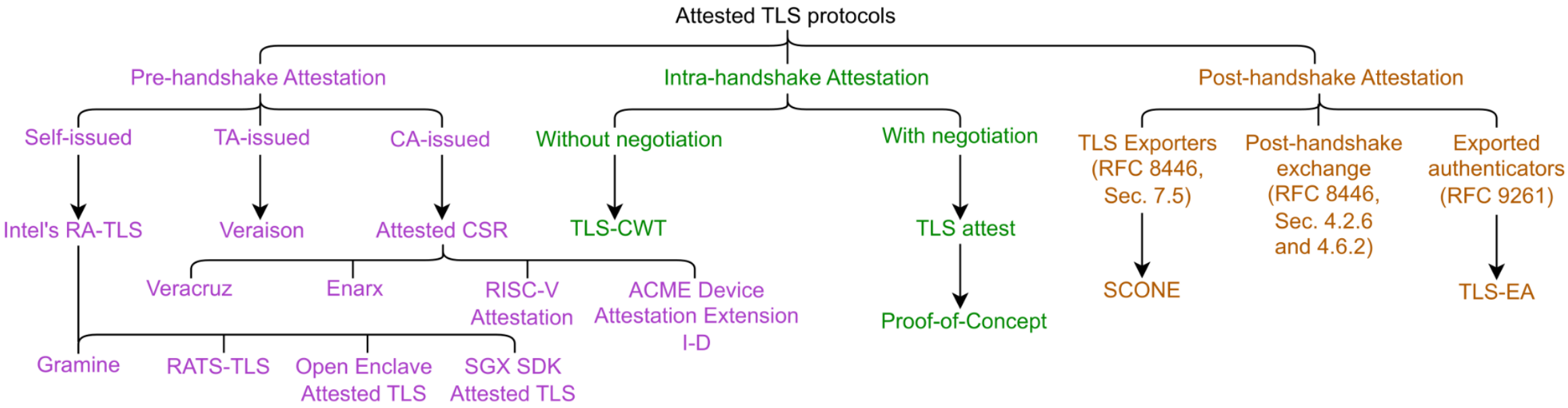
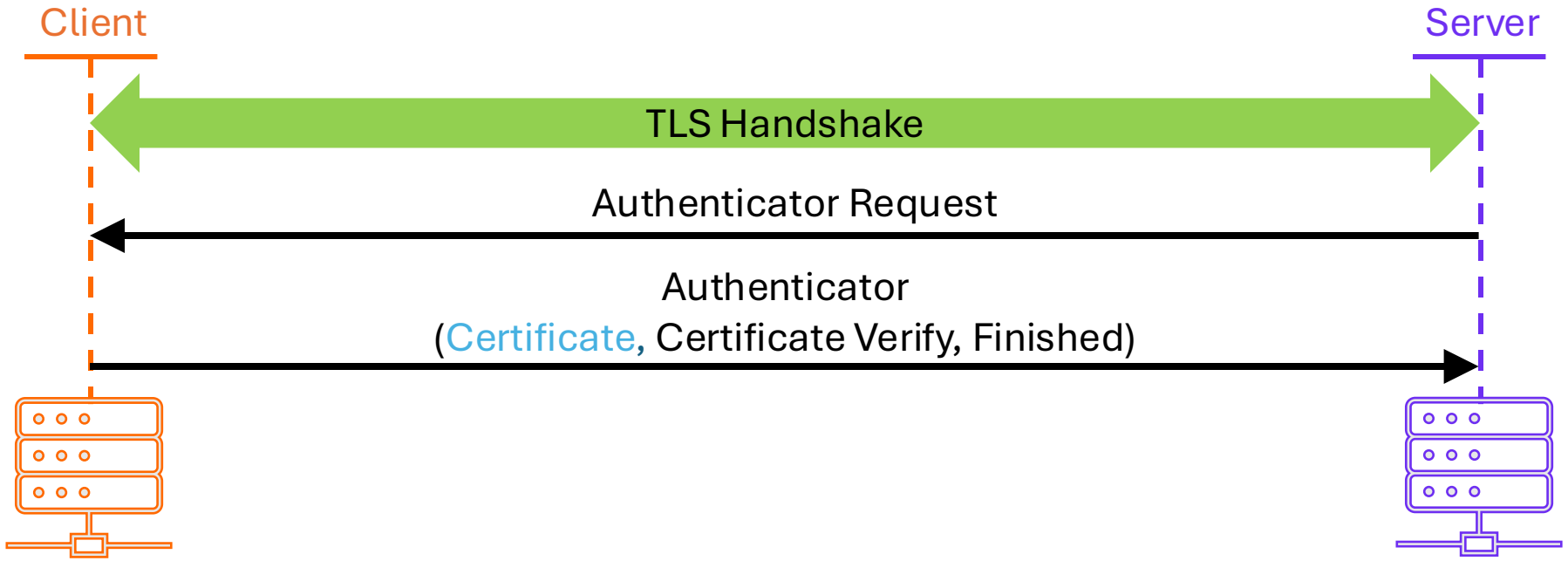


Diagram courtesy of Muhammad Usama Sardar

Current direction

- Initial proposal focused on integrating RA directly into the TLS handshake.
  - Use extension points to negotiate the functionality and convey credentials.
  - However, the integration had a high(er) risk of interference with the IETF TLS WG.
- Current proposed approach moves into the post-handshake space.
  - Uses Exported Authenticators framework to perform RA and bind it to the existing TLS session.
  - Responsibility will now fall onto the application layer to manage this request.
  - Better fulfills our required properties.



IETF status

- Two internet drafts available, both in “alpha” status:
  - intra-handshake attestation.
  - post-handshake attestation.
- Long-running attempt to get the TLS WG to adopt our work.
  - No appetite and no forthcoming interest in the functionality.
  - Internet Architecture Board (IAB) statement on RA had a chilling effect.
- Effort culminated in a BoF at IETF 123 in July.
  - Successfully advocated for the creation of a new Working Group (WG) to tackle this topic.
  - Still in the process of establishing the charter and kicking things off.
  - IETF 124 will most likely see the first meeting for the WG.

TODO

- Kickstart the new IETF WG and set initial direction based on our accumulated experience
  - Produce a comprehensive document covering our design rationale
  - Continue formal analysis work to iron out any gaps in security.
- Implement a prototype for the post-handshake flavour of aTLS
  - Engineering effort could be significant, as popular TLS libraries don’t support functionality we require.
- **Call for engagement**: join us in forming a community interested in deploying this standard when it ships, and in providing feedback along the way!