TLS Using EAP Authentication

draft-nir-tls-eap-11

Yoav Nir
Hannes Tschofenig
Yaron Sheffer
Peter Gutmann

What is TLS-EAP?

- Modification of the TLS protocol to accommodate non-certificate client credentials.
- Uses EAP to transport an authentication protocol between a back-end authentication server and the client.
- Similar to the approach in IKEv2.

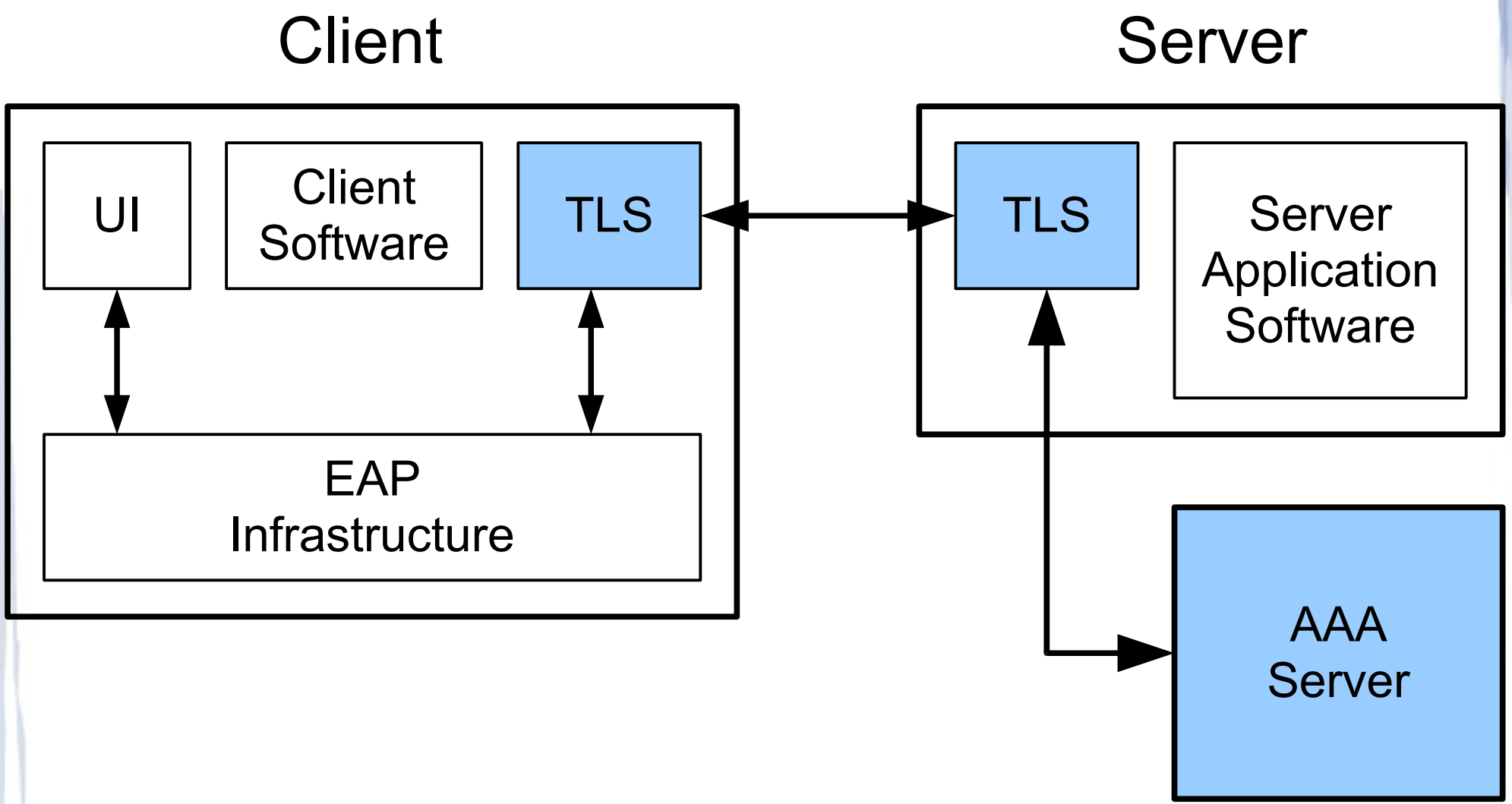
But why not TLS-SRP?

- RFC 5054 allows passwords in TLS.
- But, it requires that the SRP verifiers (salted hash of username and password) be stored on the server
- Does not work with a AAA back-end.
- Only for passwords, while EAP supports lots of exotic methods
  - Passwords, SIM cards, hardware tokens, etc.

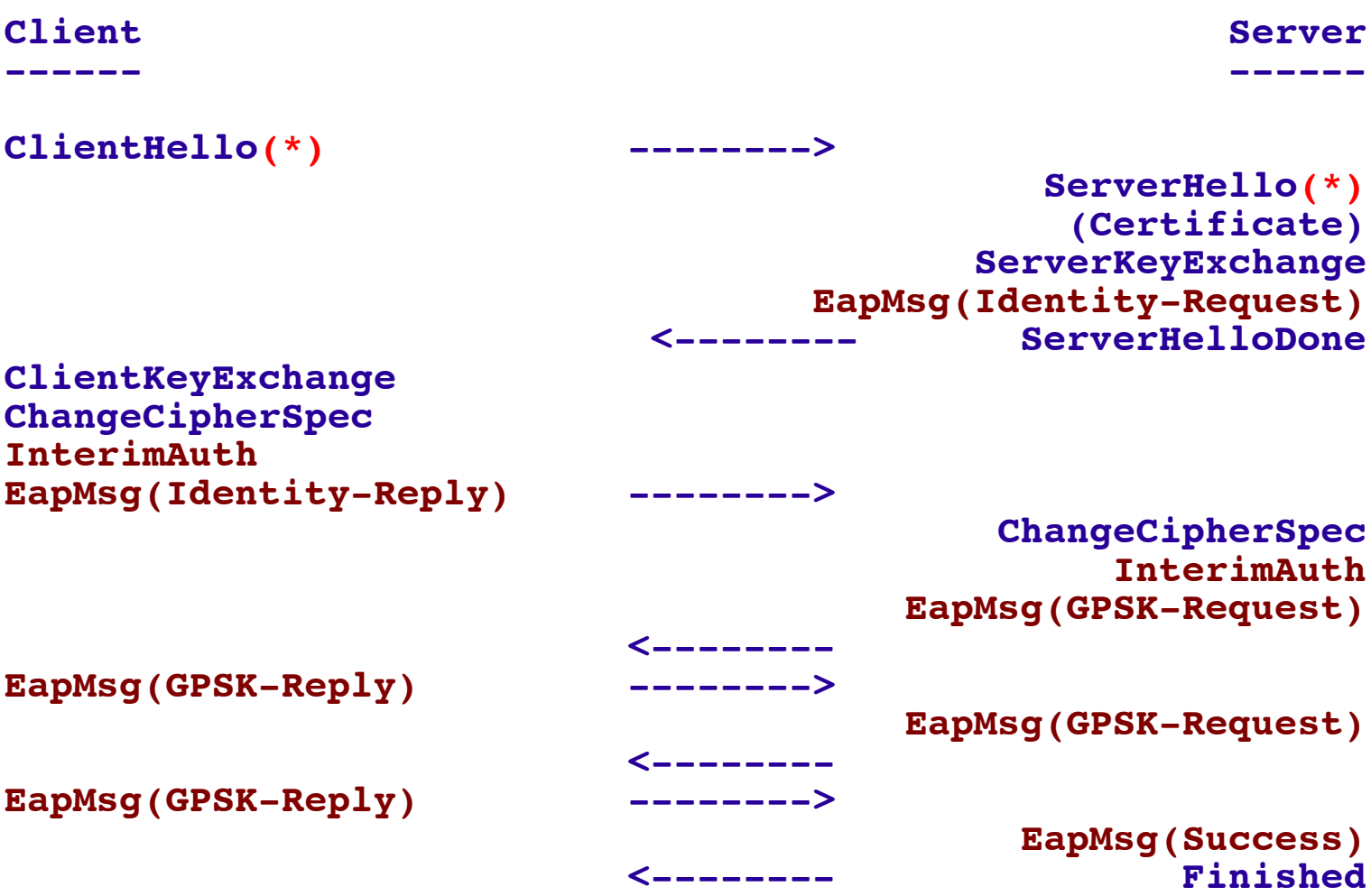
Where might this be useful?

- HTTP authentication next generation web sites (Bar BoF tonight at 8:00 in Karlin II)
  - Seeking a secure alternative to web forms
- “SSL VPN” clients
- Web services
- SMTP / IMAP / POP3 and other applications

Operating Environment



Protocol Overview



Security Considerations

- The EAP messages are protected by the TLS record layer.
- Key-generating EAP methods generate a shared key called MSK, which also goes to the TLS server.
- The MSK is used to sign the Finished message, which binds the TLS session to the EAP authentication.

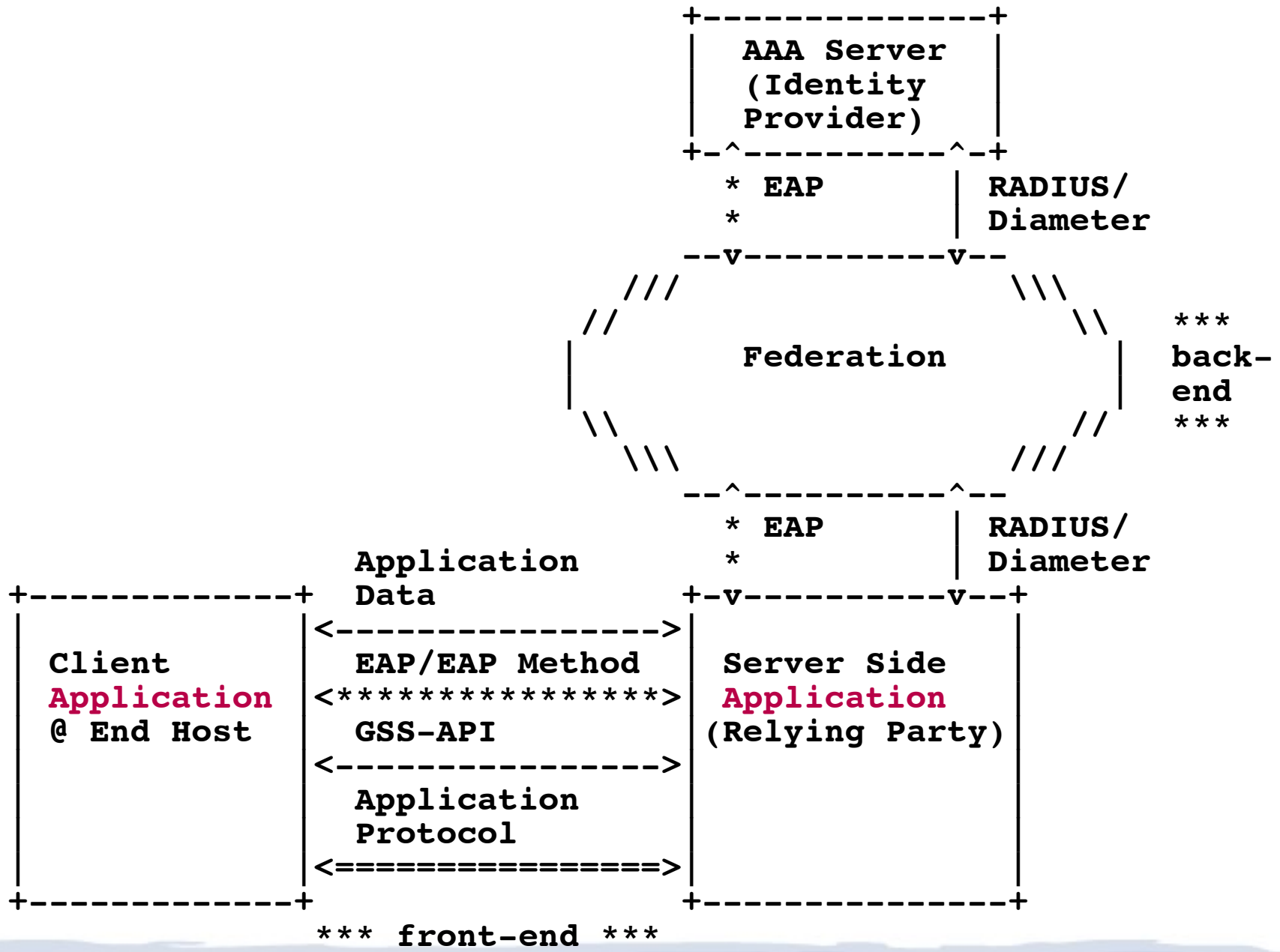
Questions?

Backup Slides

EAP Applicability Statement

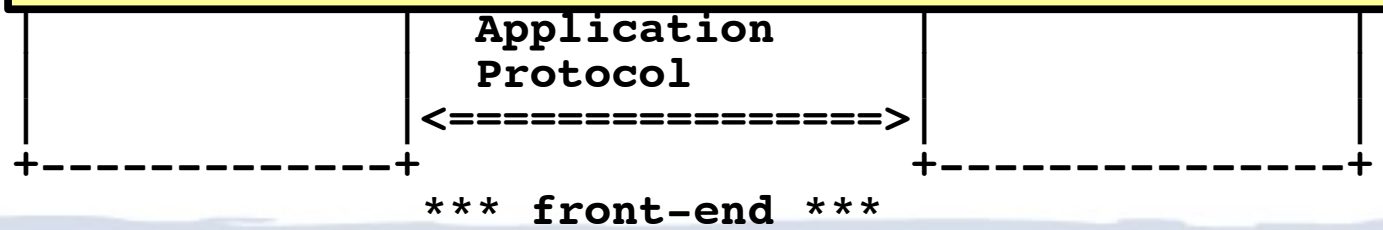
- When we published the first version of this draft, people pointed us to section 1.3 of RFC 3748
- `EAP was designed for use in network access authentication, where IP layer connectivity may not be available. Use of EAP for other purposes, such as bulk data transport, is NOT RECOMMENDED.`
- So, is EAP applicable to TLS?

Now there's ABFAB

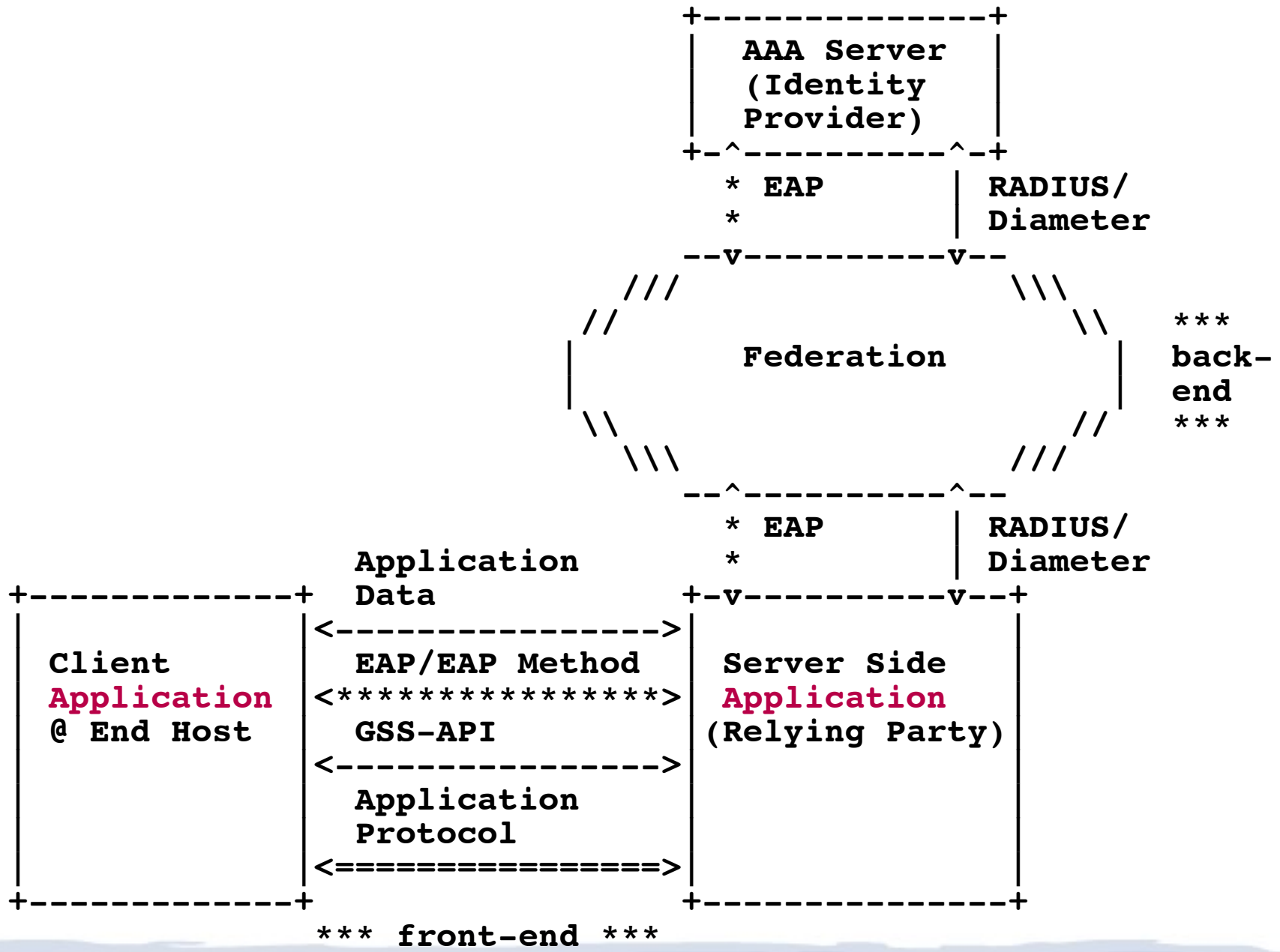


Now there's ABFAB

- ABFAB is about client access to ***applications***, not network access.
- Authentication is tunneled from a client that may be anywhere, through an application server, to an identity provider that is somewhere else.



Now there's ABFAB



Why not Finished and ext-Finished?

- There was a suggestion that we call InterimAuth “Finished”, and make a new name for the last handshake message, such as “EAP-Finished” or “extended Finished”
- RFC 5246 says this in section 7.3:
  - `In response, the server will … send its Finished message under the new Cipher Spec. At this point, the handshake is complete, and the client and server may begin to exchange application layer data.`
- I think this semantic of “Finished” is more important than the exact key that's used.