Renegotiation

Framing slides
Proposal

Design Space: Level 1

A) Renegotiation
   - Maybe with profiling
B) Stop and start again
C) Change some things
   - Just Master Secret (rekeying)
   - Master Secret and Authentication
   - PFS characteristics?
D) New connection please

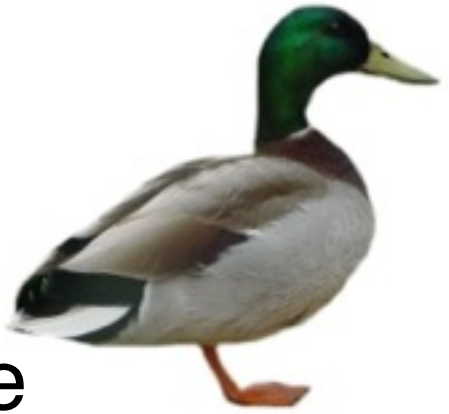
A) it ain't broke

- Various arguments for this:
  - We've never had any issues with renegotiation; those issues aren't protocol problems
  - It's useful; I use it all the time
  - Argument from inertia
- Option to have applications enforce immutability on the various properties that they care about (like peer credentials)

B) hiccup

- Keeps the TCP connection open for a new TLS connection
- Basically all the properties of renegotiation
  - Period of “dead air” while the handshake completes
- Arguments for are that it simplifies analysis
- Arguments against keeping renegotiation are generally problems of usage, not correctness
  - Those apply equally here
  - And this has worse performance characteristics

C) rekey



- Everything is intrinsically immutable
  - Except the master secret
  - And maybe we have additive authentication
- Provides for long-lived connections
  - Rekeying keeps the consultants happy
  - At a cost in complexity

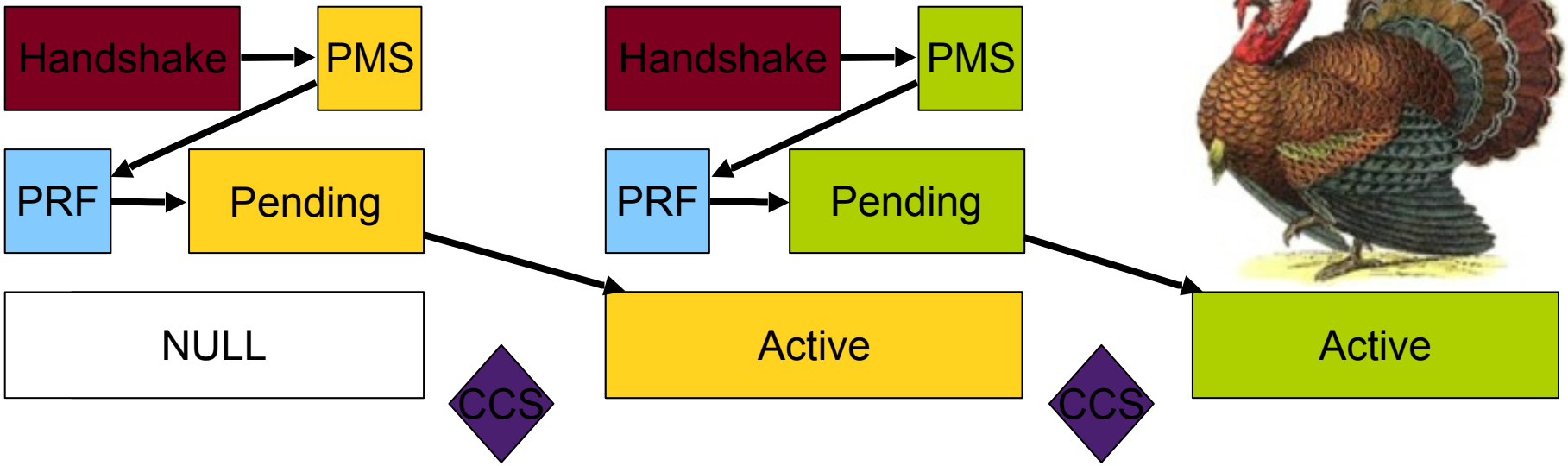
D) start over

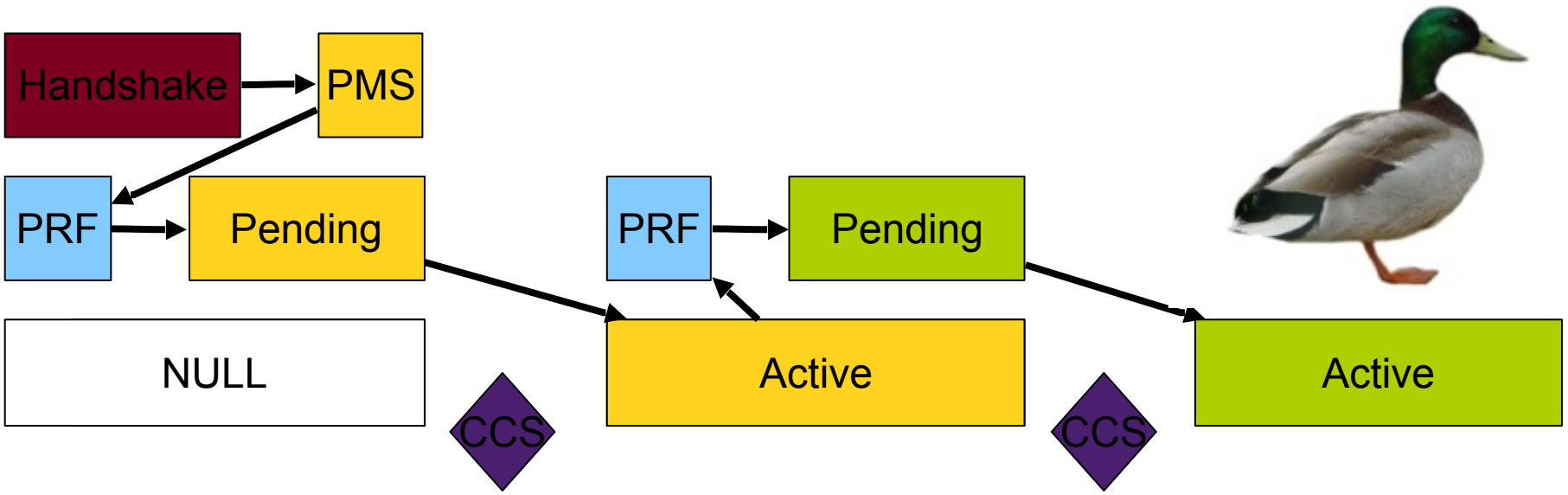
- Everything is immutable
  - Have to start over to change anything
  - No rekeying
- Keys are good for long enough to address most use cases

Proposal

- Rekeying
- Consider addon authentication

Rekeying process





(P)FS*

- Goal: prevent a leak of session keys or a machine compromise from revealing old communications for the same sessions
  - Potentially useful for long-lived sessions
- Idea:
  - Require a reciprocal ChangeCipherSpec so that at most two master secrets are active (usually one)
  - Using the PRF means that the previous MS can't be recovered from the current one
- A new (EC)DHE exchange is possible
  - At a cost in complexity and time
  - Didn't seem worth it