

Post Sockets

An Abstract Programming Interface for the Transport Layer
draft-trammell-taps-post-sockets-03

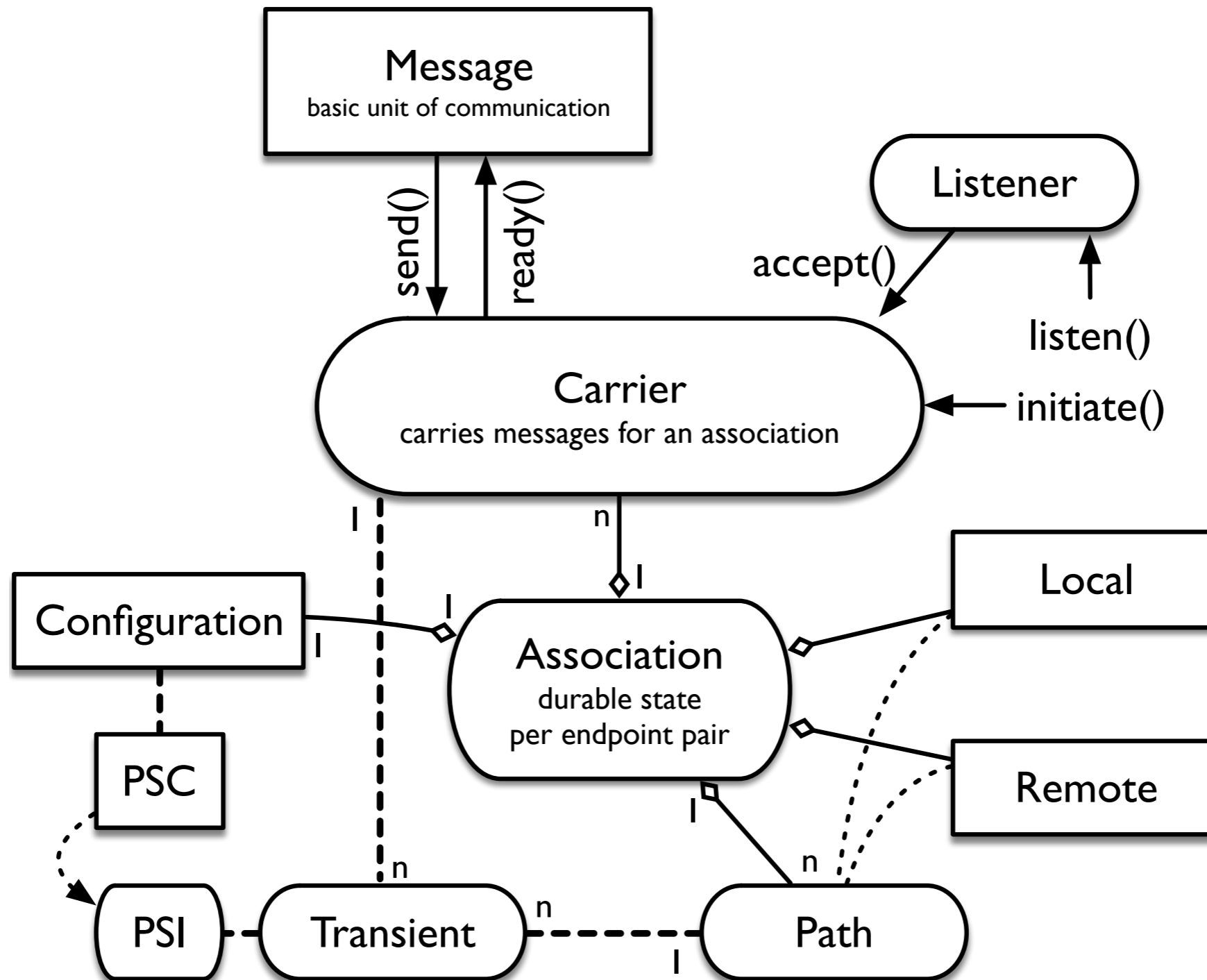
Brian Trammell, Colin Perkins, Tommy Pauly,
Mirja Kühlewind, and Chris Wood

TAPS WG — IETF 100 Singapore — 14 November 2017

Changes since -01

- Terminology: **policies are now called *configurations*** and explicitly bound to associations (§2.6)
 - These are layered (see IETF 99 presentation).
 - Explicit separation of Protocol Stack Configuration (PSC) to specify which protocols are acceptable.
 - PSC may contain protocol-specific tweaks.
- **New API dynamics section:** how it all fits together (§3.2)
 - More detailed examples to follow implementation work.

Abstractions, updated



Open Issues and Next Steps

- We're working on the details now.
 - Define a simple protocol-independent carrier state machine
 - How to represent certain transport-specific interactions (e.g. SCTP-PR never-retransmit semantics, handling of carrier multiplexing in multistreaming environments)
- see <https://github.com/mami-project/draft-trammell-post-sockets/issues>
- Is it time to bring this work into the TAPS WG?

An observation on charter item 3

- Critical mass of AbstractPI (NOT AN API) proposals before the WG
 - ...minset, post-sockets, neat, crypto-sep, intents, happy eyeballs, racing guidelines...
- Our view:
 - Requirements come from minset.
 - We should take the creative leap beyond that to design an abstract API that makes sense and is forward-looking, rather than just collating the existing options in the least-bad way.
 - That creative leap is some combination of the proposals at hand.
 - That combination should happen within the WG.