Post Sockets:
An API for the FTL

Brian Trammell / Mirja Kühlewind, ETH Zürich

3. MAMI Plenary, Seville, 6 February 2017

mami

measurement and architecture for a middleboxed internet

measurement | architecture | experimentation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688421.The opinions expressed and arguments employed reflect only the authors' view. The European Commission is not responsible for any use that may be made of that information.

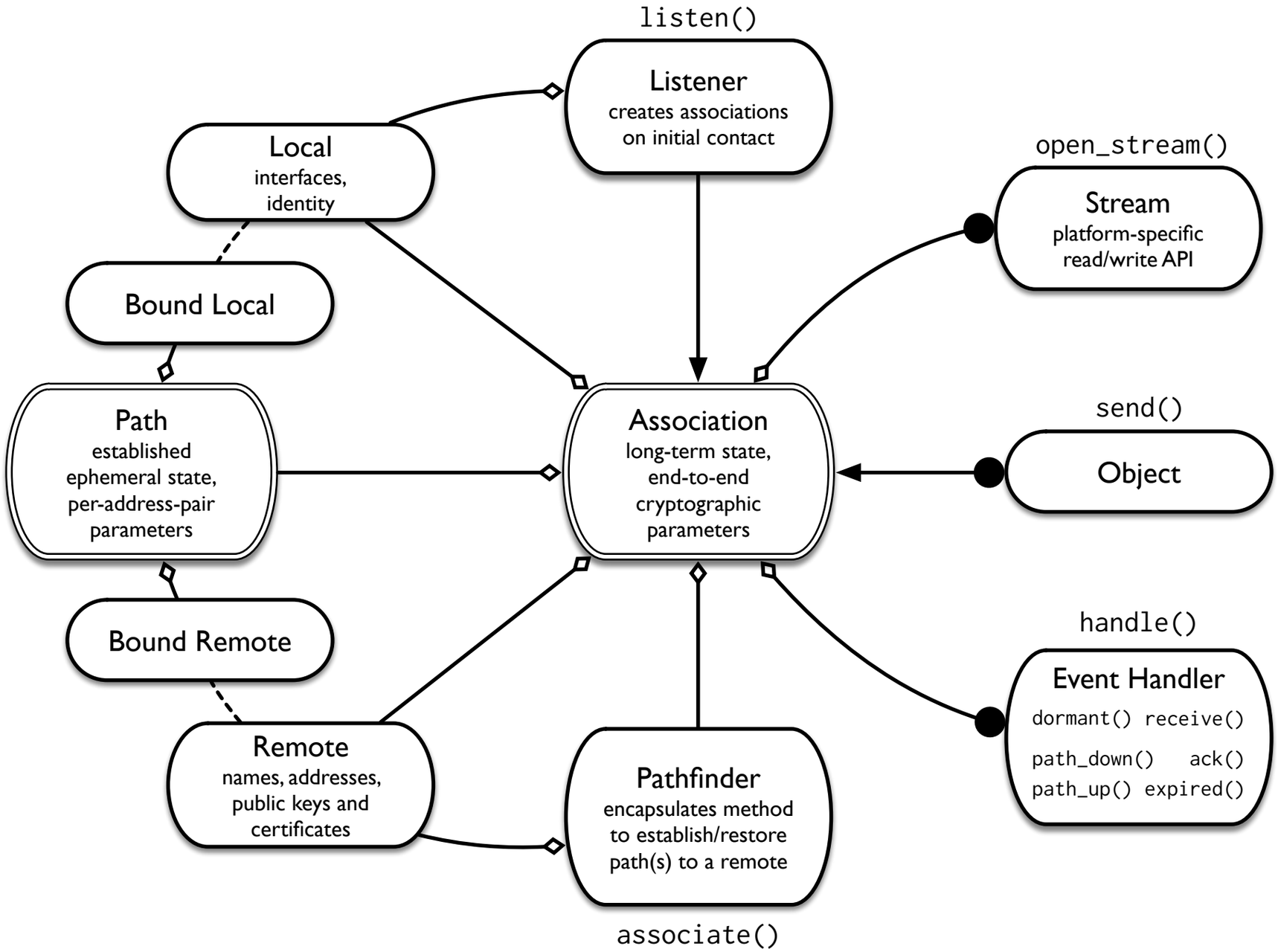
Supported by the Swiss State Secretariat for Education, Research and Innovation under contract number 15.0268. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the Swiss Government.

A Few Insights about Network Applications

- ***Applications deal in objects*** (messages) of arbitrary size
  - Files, assets, media frames, etc. may *depend* on each other, but have no strict ordering.
  - “Real” streams exist too, when data source has unknown length, not easily divisible.
  - Most “streams” are simply the result of applying sequential-file logic to networking
- The network of the future is ***explicitly multipath***.
  - Applications must have access to the properties of these paths.
  - (And may be able to communicate with the path about these properties: MCP)
- Future transports must ***guarantee security properties***.
  - Path elements must not be able to see transport-layer metadata.
- Message reception is ***inherently asynchronous***.
  - Present scalable programming models enable async IO.
  - Core kernel/user interface may remain `select()`-based for performance.

Abstract Programming Interface

Classes and Entry Points



Abstract Programming Interface

Object and Stream properties

- Objects and streams have a **niceness**
  - Nicer `send()`s/`write()`s yield to less nice
- Objects have a **deadline**
  - An object will be cancelled if it cannot be realistically received before this deadline
  - Infinite-deadline objects are fully reliable
- Objects may have **antecedents**
  - other objects which should be sent before

Transport Independence

- Only two requirements for transport on the wire:
  - Framing for objects
  - Some (non-address) way to identify associations
- Assumption that the transport protocol provides encryption for payload confidentiality and public header integrity protection.
- Can make use of other transport features on demand:
  - Multipath load balancing and migration
  - Multistreaming for objects and streams
  - PLUS for path property exposure
- Object properties (niceness, deadline, dependencies) are sender-side only; path properties can be derived locally too.

Work to do

- Complete design
  - Post Sockets Secret Cabal Meeting in Zürich next week
  - Consider impact of mobile connection racing, interface- v. path-bound properties (e.g. work in IETF MIF), etc.
  - Define path properties beyond “up/down”
    - Defined properties: interface cost/preference
    - Measurable properties: RTT/loss rate
    - Exposed properties: Lo/La, etc. via PLUS
- Pilot implementation and experimentation
  - Current incomplete sketch in Go at https://github.com/mami-project/postsocket
  - Build atop QUIC/PLUS in quic-go fork.
  - Build simple framed-TCP implementation to show flexibility on impaired paths?