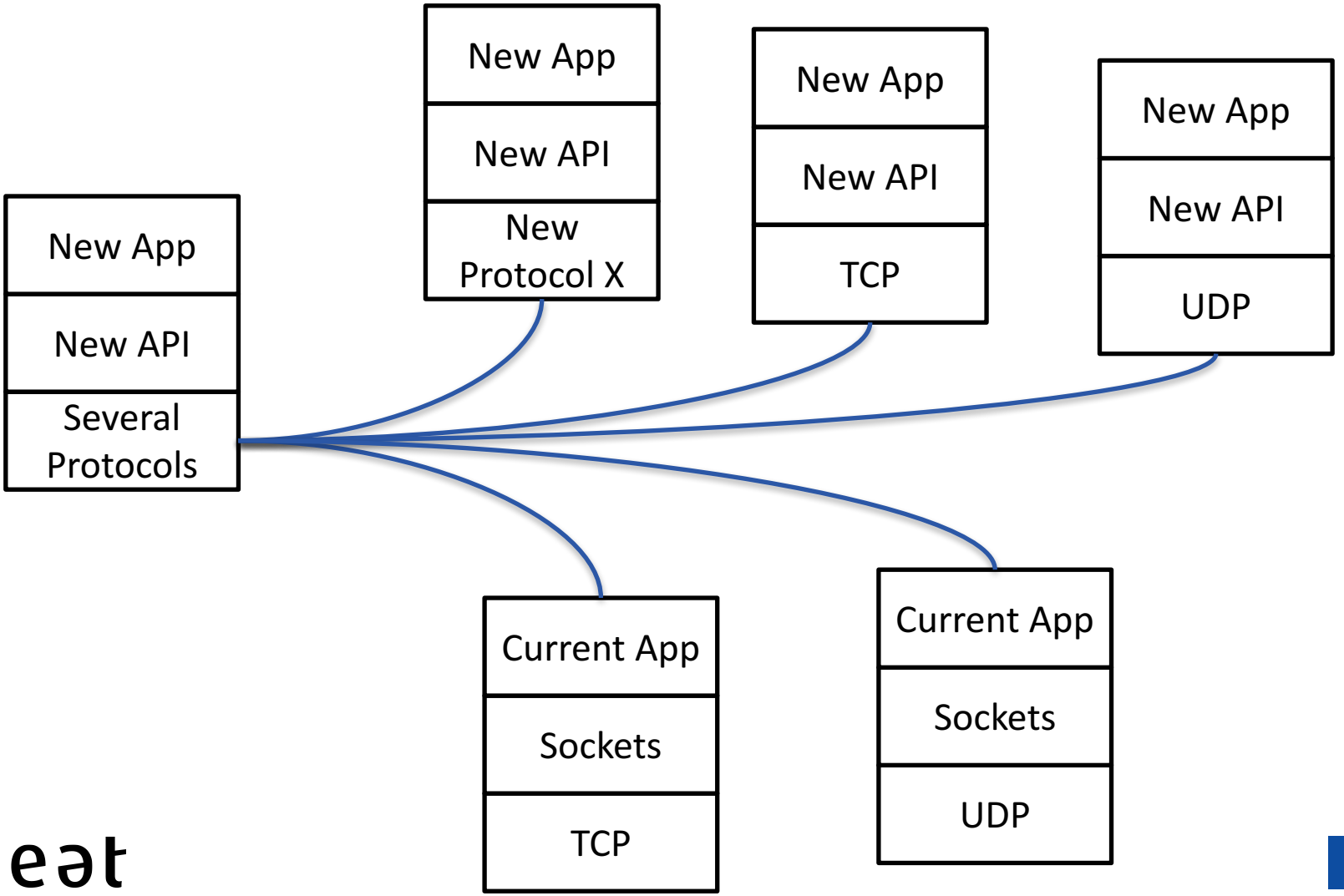
Falling back

Michael Welzl

neat

Horizon 2020
European Union funding
for Research & Innovation

Who can we talk to?



neat

Application-Framed (AFra-)Bytestream

- From minset TAPS draft:
  "require TCP fallback for SCTP"
  => can have various ways to send messages, but only receive streams

- **AFra-Bytestream:** only applications delimit data
  - Transport framing unnecessary: TCP apps already (need to) do this

- Uniform access to protocol-independent transport layer
  - Can implement all typical messaging things in transport
  - No significant receiver overhead: each AFra-block always stays intact

- Possible implementation: just a normal bytestream interface
  - Optional information from app to transport about frame boundaries
  - Receiver app can easily identify out of order blocks, lost blocks,…

What is a flow?

- From minset TAPS draft:

  No application-specific knowledge involved in decision to use multiple connections or multiple streams of an association
  - To the app, “Multi-streaming” is only a flow grouping concept

- Suggest to only expose “flows” with “group” concept
  - 3 flows can e.g. be 3 streams of one association or 3 TCP connections
  - Flows have properties: flow group number, flow priority number

- Note: this affects establishment / teardown semantics
  - E.g., streams of an association can just begin to send
    => connect() without data not guaranteed to do anything on the wire