Transport Services Strawman Architecture
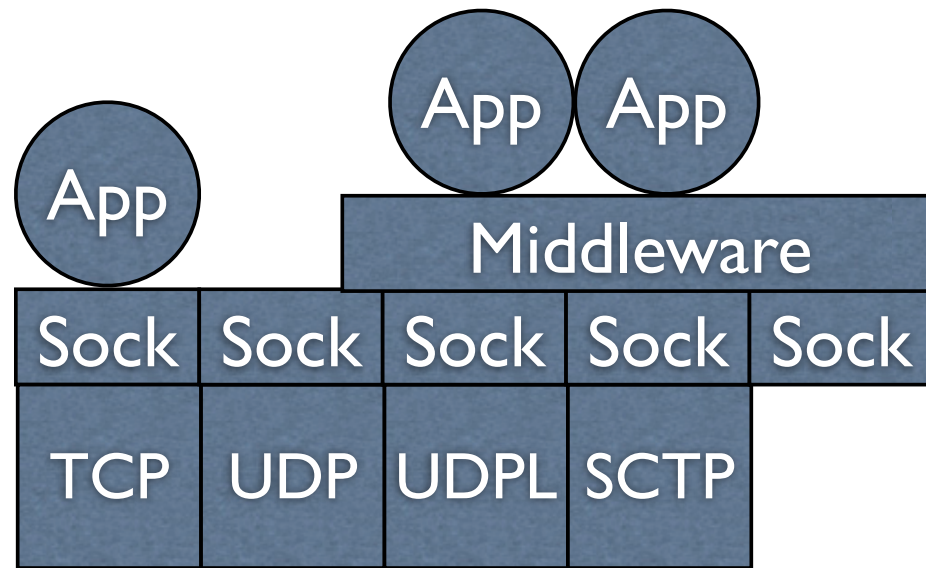
draft-montpetit-transport-strawman-00

M.J. Montpetit
I. Zhovnirovsky
B. Reuther
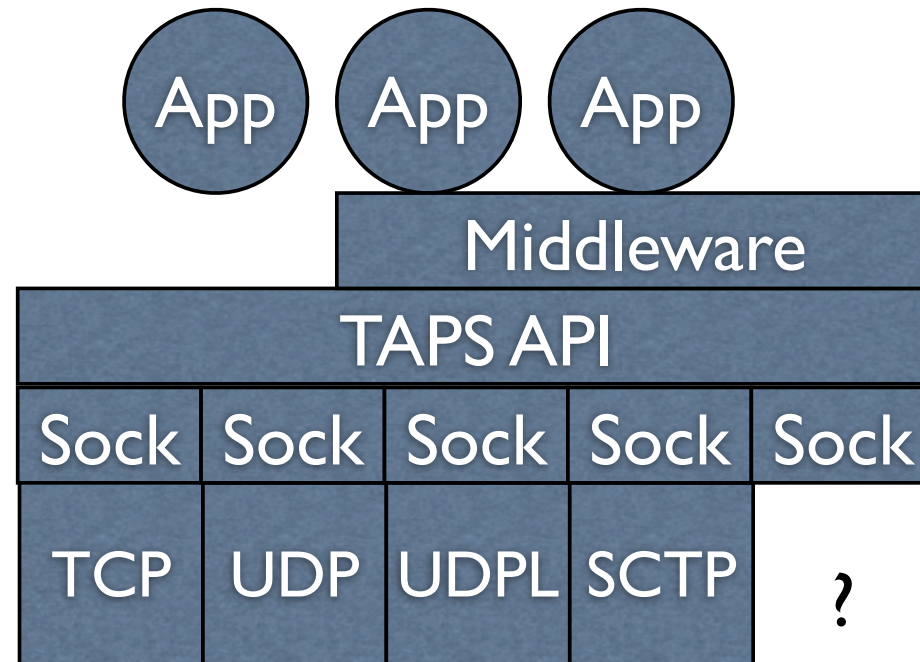
Gorry Fairhurst (presenter)

Current architecture

IETF has many options on offer, but people programming to this often just choose one, don't want to support many, or do not even know

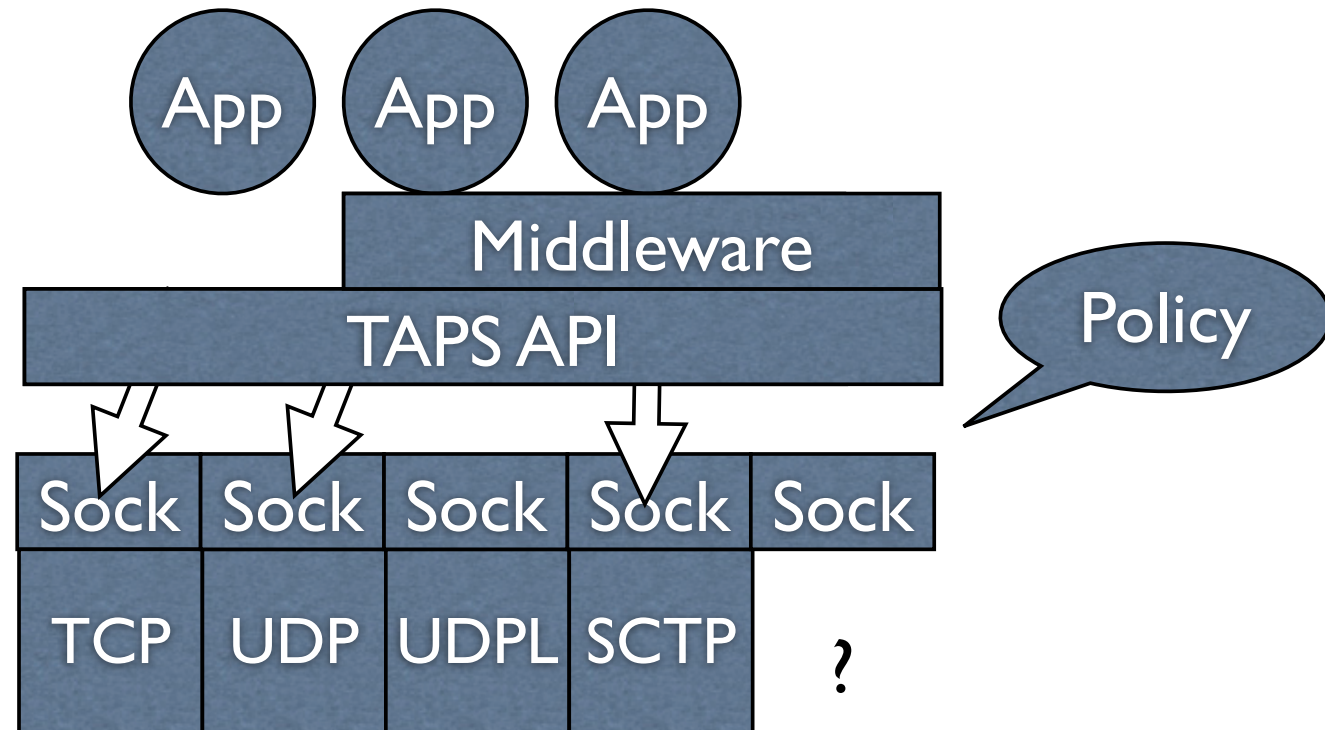# Possible architecture for choosing a transport

App App App

Middleware				
TAPS API				
Sock	Sock	Sock	Sock	Sock
TCP	UDP	UDPL	SCTP	?

Two models for deployment:

… change apps

… use middleware/library and change

Selecting the *Most Appropriate* Transport

Select using:

- application requirements
- other information

Goal 1: Choice of protocol can become transparent for application

Key thing is that binding to transport is **_at runtime_**.

- Based on what is available at both ends
- Based on what the app wants to do
- Based on what we know of the path characteristic
- Based on what we know of the platform
- Based on admin policies / provisioning domain

…some configuration objects may be global to a node, others are local to the interface.

Goal 2: Needs to work if this is only implemented at the sender

Only *requires* a sender-side change.

If we can also change the receiver, then we can do *much* more... but this is no starting point.

Goal 3: *Faster* choosing of an appropriate transport.

Various ways of finding what transport works for connecting to an endpoint.

There are *more* transport options, this is not getting easier.

Multiple interfaces often available.

Evolution will now be possible!