MIF API
draft-liu-mif-api-extension-06
Status

• Previous version was sketchy and incomplete
• Current version is an attempt at a complete API
• No review yet
• Would like some review
Provisioning Domains

- API exposes interfaces
- All actions are in terms of provisioning domains
- Need more clarity: what is a provisioning domain
  - Is link local addressing a separate pd?
  - Is IPv4 a separate pd?
- What about RA vs stateful DHCP?
API Model

- Application
  - High level API
    - MIF API
      - Communications API
        - Network Link API
          - Interface 1
          - Interface 2
          - Interface 3
Application

- The ultimate customer of the API
- Can use API directly as well
- May receive connections as well as originating
- Web browser, peer-to-peer app, etc
High level API

- E.g. connecttoname()
- E.g. happy eyeballs
- Easy for app implementors to use
- Requires some or all features of MIF API
- Not used after connection established
MIF API

- Abstract API, no language bindings
- Too low level for most apps
- Provides sufficient tools to implement HL API
- Does not constrain HL API implementations
- Message-passing, not synchronous
- Analogous to unix routing socket
Communications API

- How application communicates after connecting

- MIF API and HL API yield a handle that is then used with this API

- Not otherwise used directly by MIF API

- Must be possible to do MIF API and communication API at the same time without interference

- e.g. BSD Socket API
Network Link API

- This is typically the OS IP stack
- Controls interfaces
- Sends and receives packets
- Handles routing
- I've also stuffed DHCP and ND in here for brevity
- Should I make that more explicit?
What's next?

- Is this working group work?
- How much should we flesh out what's happening under the covers, if at all?
- Comments, flames, suggestions, offers of help?