Goals

• Making the client visible to services
• The bare minimum of “wiring” we need to layer into the specifications
• Moving from implicit to explicit routing of messages aimed at the client
• Allowing asynchronous messages to reach the client from a broad range of possibly deployed services.
Status

• Draft submitted
• A few minimal comments
• Time is ripe to solve this before we add new drafts such as Teleport into the specifications pile
• Good time to ensure WebSockets, etc. have our requirements
Region to Facades to Generality

- Single Region as proxy is simplest case
- Current approach creates a pair of facades, one for Region, one for “agent” services
- Allowing general deployment pattern adds new challenges
- Asynchronous notification delivery is the core challenge
- Supporting multiple protocols over time is a related challenge
Region as proxy to facade

- Routing is implicit
- As soon as there are regions which want to talk to back end services outside their own cloud, we are exposing some back end service interfaces
• How does the separate service deliver asynchronous updates – especially when the client is behind a firewall
• How do we manage if the separate service wants to use Hybi, XMPP, or other transport
If “client” is addressable (Not firewalled)

• Just need a URI to delivery your messages to
• Currently we don’t have one
• Client caps proposes to make that explicit
• URI also covers transport choices
• Ensure that this happens in core
If “client” is firewalled

- Need non direct address or reversed transport
- WebSocket (hybi), Comet style long polling, etc.
- Non direct address has implications
- WebSocket requires client to setup, and requires proper addressing to service setup happens