P2PSIP Concepts and Terminology

draft-willis-p2psip-concepts-04

Philip Matthews
David Bryan
Eunsoo Shim
Dean Willis

Major Changes This Rev

- Terminology
- Locating and Joining an Overlay
- Role of the Overlay as Distributed Database
- NAT Traversal and Transport Service
- Credentials
- Client Models

Terminology Changes

∆ Overlay Name

Was “Overlay-ID”

user-friendly name.

∆ Client

definition cleaned up, deleted “peer subset”.

+ Service

definition added.

+ Service Name

added user-friendly name.

∆ Users

named human, aka “User Name”

- ∆ Resources

stored in overlay by resource ID.

Labeled by user or service name.

- P2PSIP UA

deleted

+ Joining Peer

+ Bootstrap Peer

+ Admitting Peer

∆ Admission vs Insertion

clarified

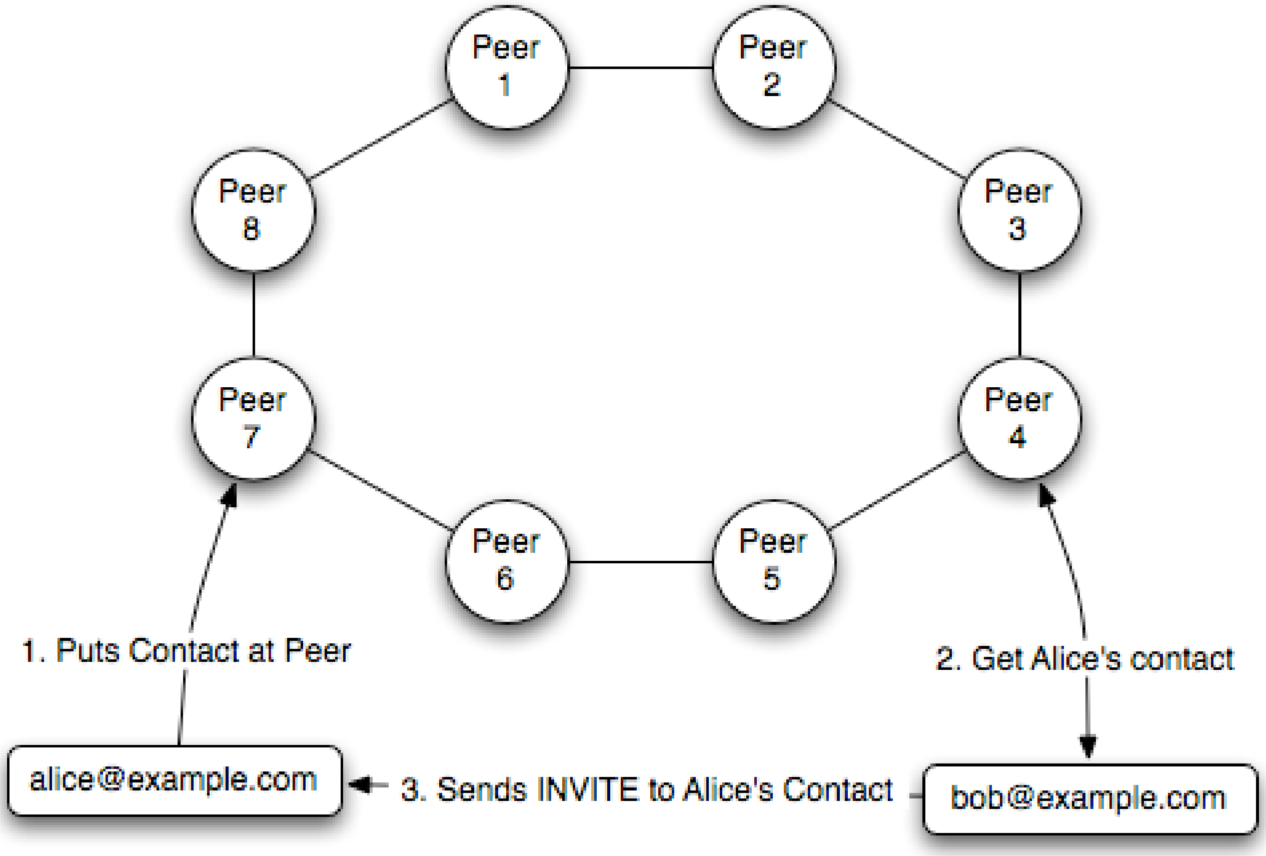
Locating and Joining an Overlay

- Current rev has more detailed model
- Several Roles Defined
  - Joining Peer
    - The one coming in
  - Bootstrap Server
    - Easy-to-find introducer, finds a Bootstrap Peer
  - Bootstrap Peer
    - finds appropriate Admitting Peer
  - Admitting Peer
    - Performs insertion into database
    - May stay on as link or referrer
- Roles can be combined in implementation

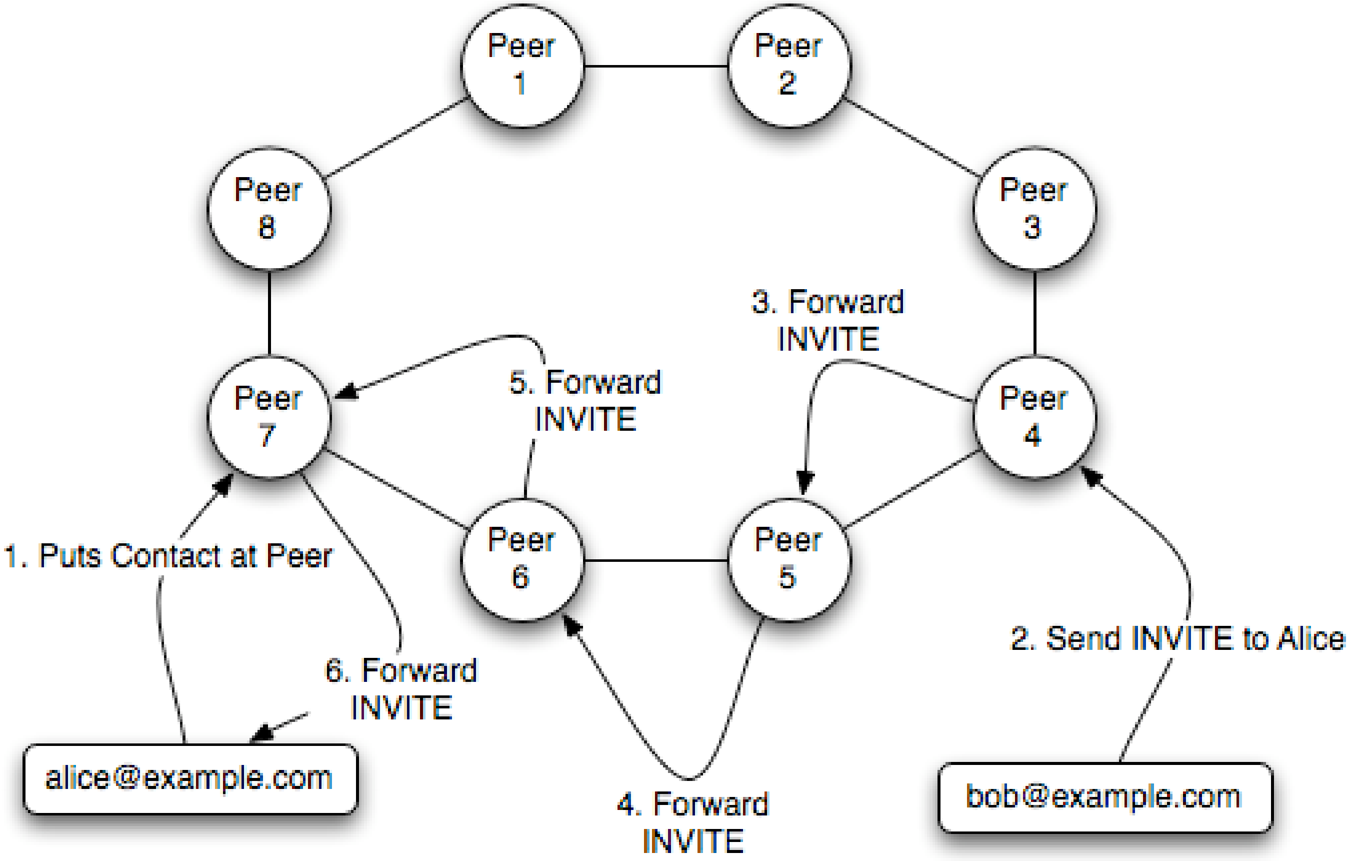
Role of Overlay as Distributed Database

- Overlay provides a distributed database
  - Database stores information about resources
  - P2PSIP will standardize resource representation
- Database stores resource records keyed by ID
  - Users, whose names hash to resource IDs
  - Services, whose names hash to resource IDs
  - Other types of resources?
- How does overlay work to pass messages?
  - Store contact, retrieve contact, and use with SIP
  - Store contact, pass message along peers to contact
  - Store “serving proxy’s” peer ID, use it to route SIP

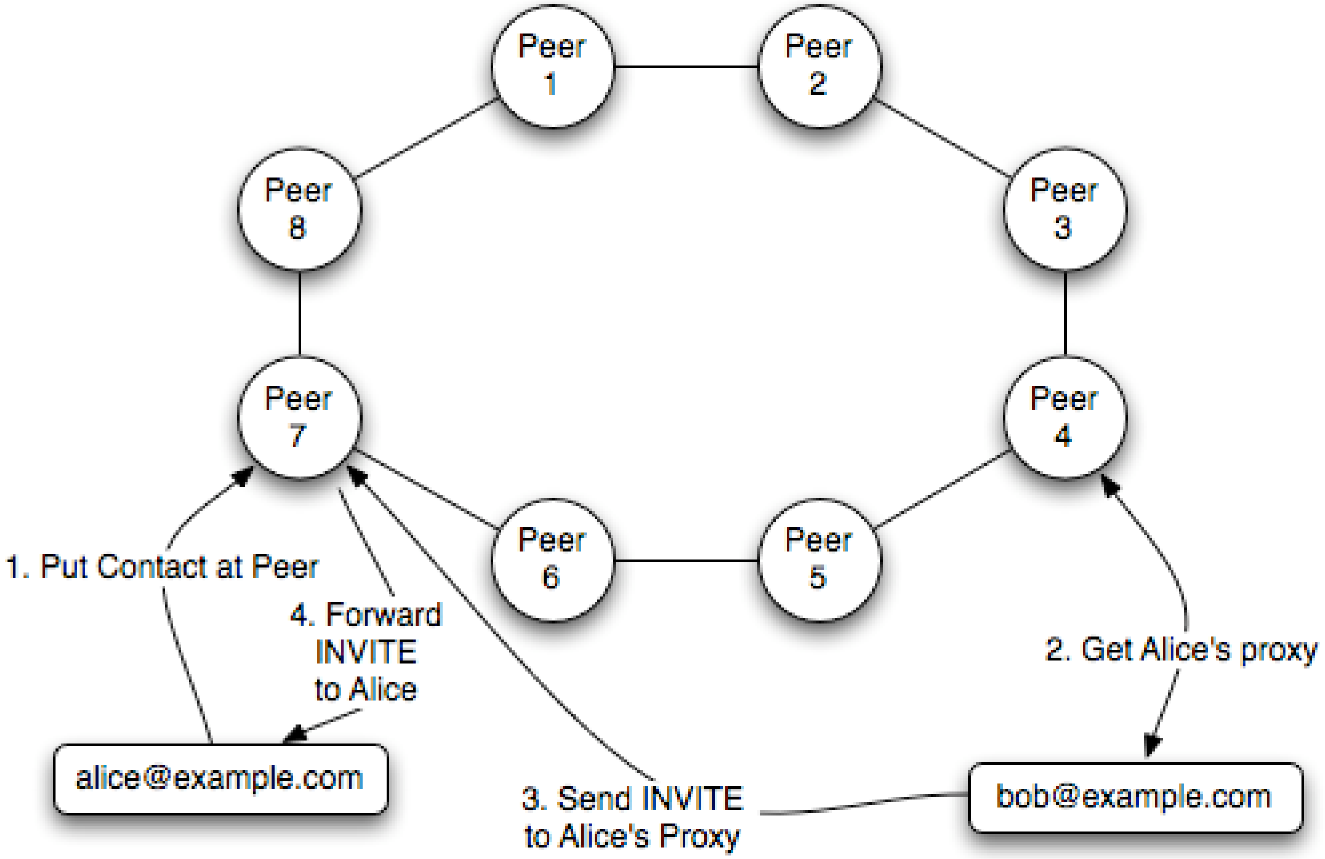
Overlay Stores Contact



Overlay Routes to Contact



Overlay Stores "Proxy" Peer



NAT Traversal and Transport Service

- Two possible approaches now described
  - Super (public) and ordinary (natted) peer
  - Fully-distributed, aka Partial mesh of persistent connections with edge routing
- Note: Transport discussion later in this meeting.

Credentials

- Resources, especially Users and Services, have credentials.
- Used in authentication and authorization decisions.
- Peers hold and show credentials for the users and services they are representing.
- Are peer credentials a discrete class?
- How do service credentials work?

Client Models

- Requirement for a client still under debate.
- Three models proposed:
  - Client attaches to a peer, which does all the work for it
  - Client attaches to peer and acts as storage auxiliary for that peer
  - Client interacts with distributed database, but doesn’t act as part of database
- Further discussion later in this meeting.

Additional Questions

- Selecting between multiple peers offering same service
- Visibility of messages to intermediate peers
- Hybrid Domains

Futures

- Should we publish this as a concepts and terminology info draft or is this draft becoming the “Framework” draft called for by our charter?