

Outbound Discovery and High Availability

Jonathan Rosenberg
Cisco Systems

Problem Statement

- Support discovery of outbound proxies for SIP outbound
- Support mid-dialog failover for SIP outbound

Interesting Side Effects

- Mechanisms that resulted also ended up solving
 - Domain certs
 - Connect reuse

Three Orthogonal Functions

- Mechanism for discovery of outbound proxies through REGISTER
- Connection stickiness for mapping URI to connections
- Usage of connection stickiness with SIP outbound to achieve HA

Three Orthogonal Functions

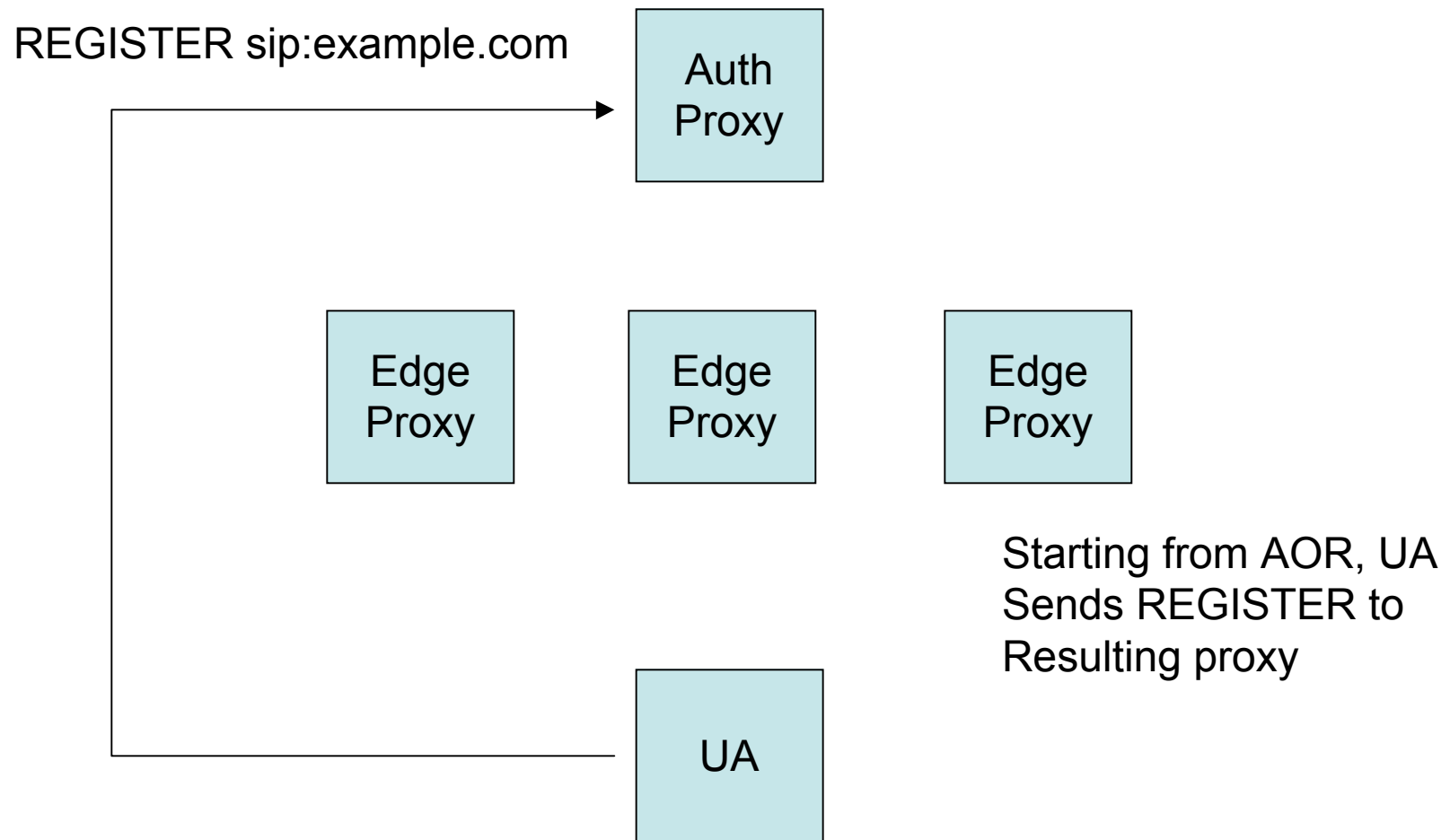
- Mechanism for discovery of outbound proxies through REGISTER

- Connection stickiness for mapping URI to connections

- Usage of connection stickiness with SIP outbound to achieve HA

This is the general purpose tool that also solves
Connect reuse and domain-certs problems

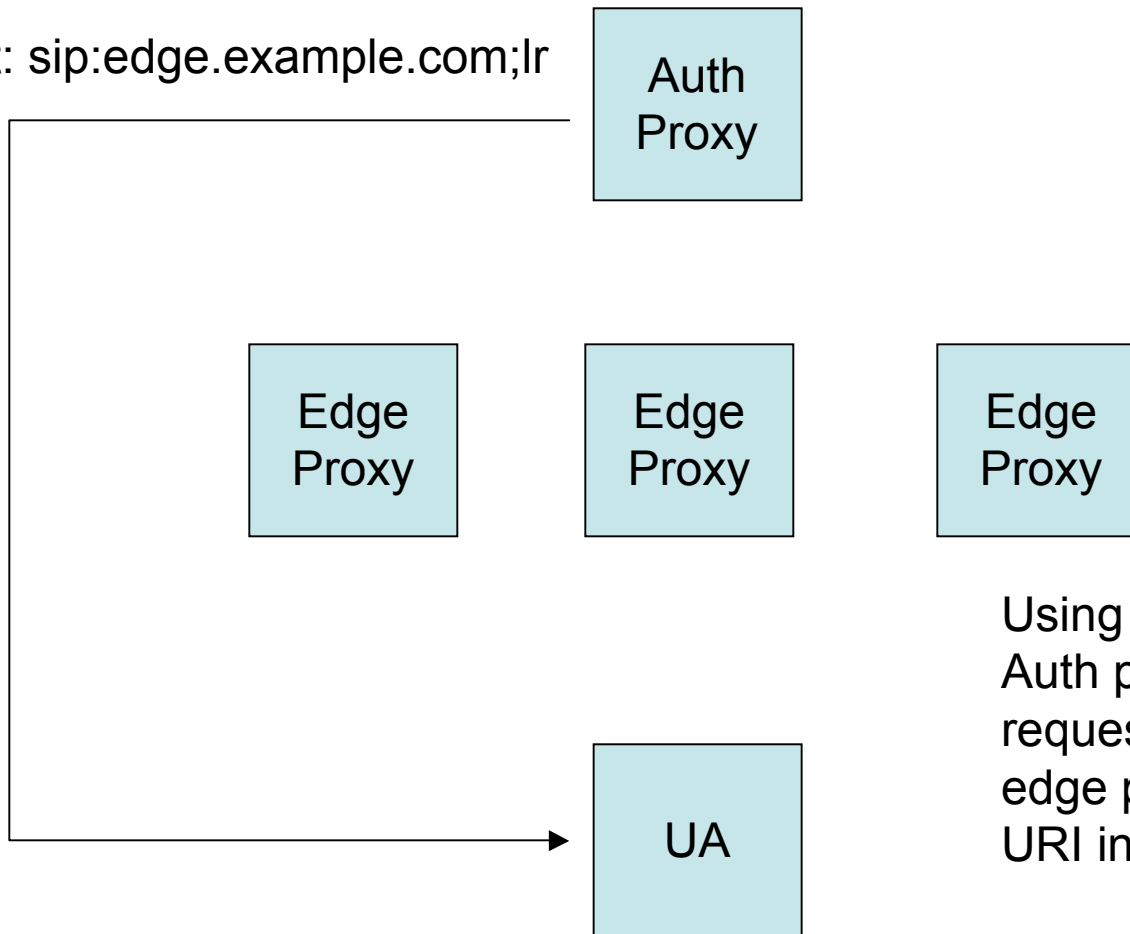
Discovery: Home Case



Discovery: Home Case

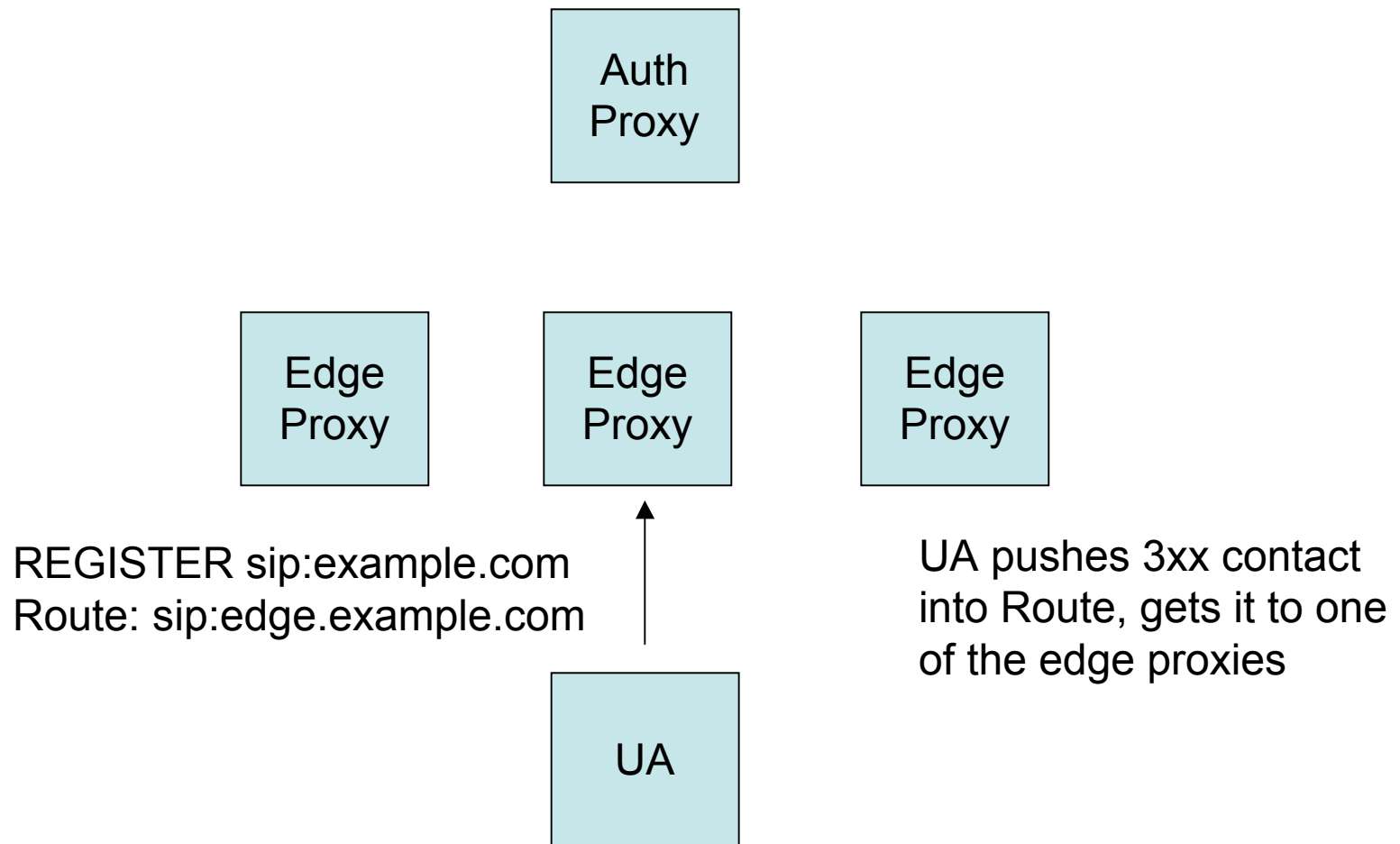
300

Contact: sip:edge.example.com;lr

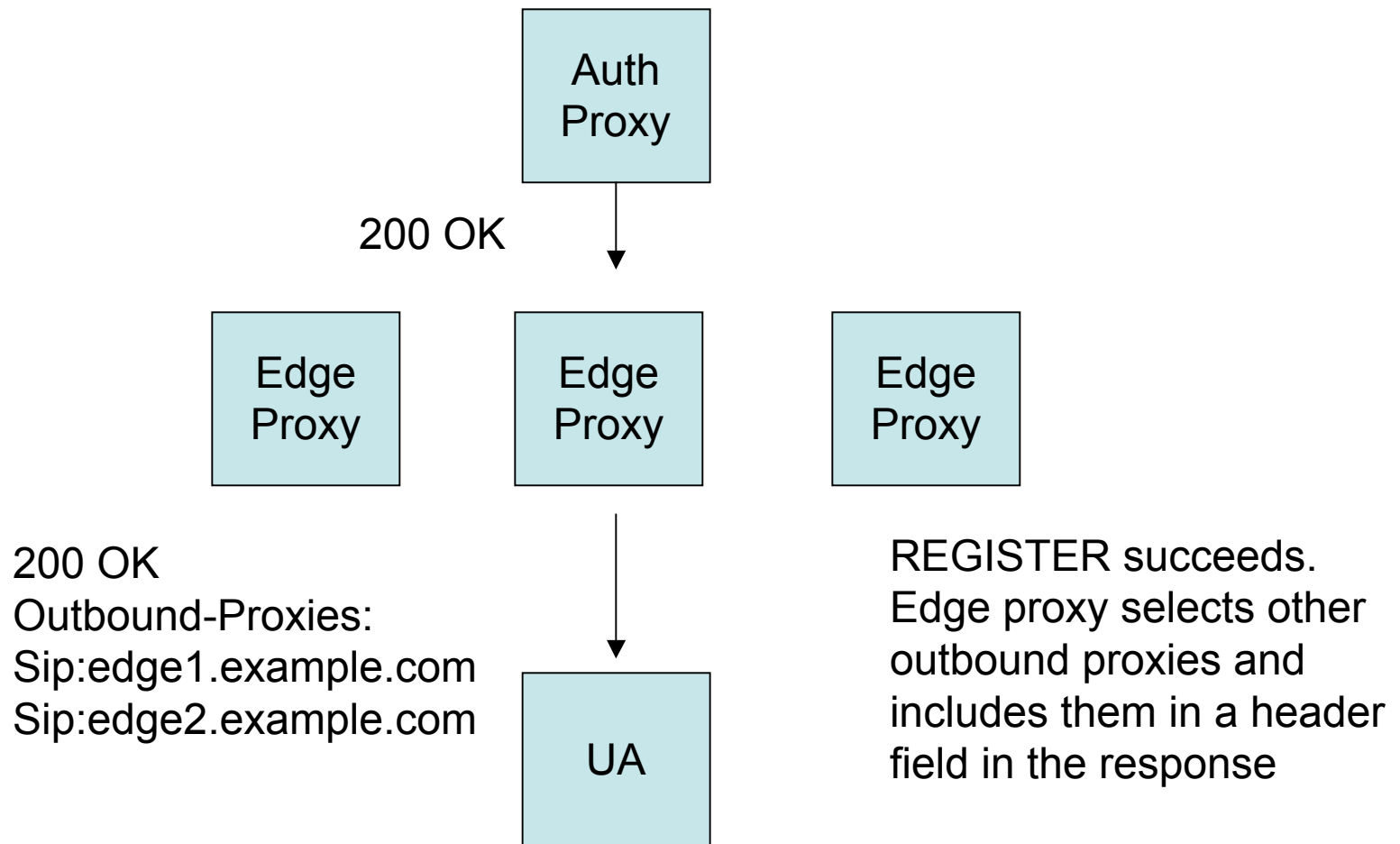


Using loose routing principles, Auth proxy redirects (since request didn't come from edge proxies). Note cluster URI in Contact

Discovery: Home Case



Discovery: Home Case



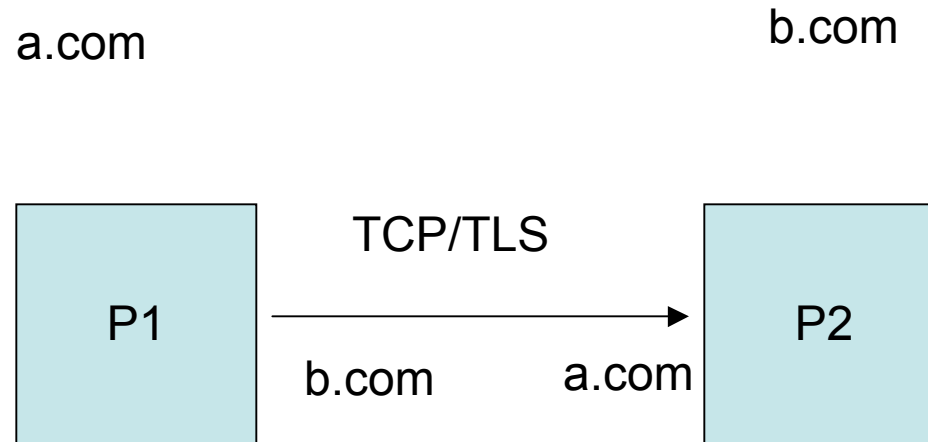
Important Notes

- This mechanism allows the edge proxy to select the other outbound proxies
 - Essential for HA – see later
 - Readily allows different outbound proxy sets per subscriber
- Have not yet discussed Path or how to properly set outbound proxy URI to achieve HA
- Discovery mechanism itself only requires UA loose route in addition
- Draft also discusses case where UA has discovered proxy via DHCP or other means

Connection Stickiness

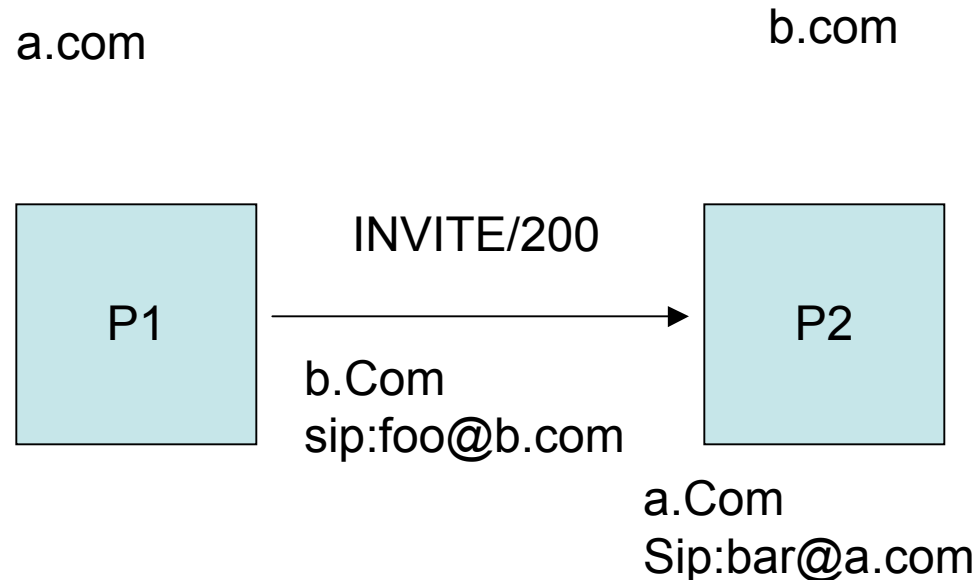
- Basic Idea
 - Alternate model of mapping URI to connections and flows, instead of DNS
 - UA maintains a table of URIs bound to each connection/flow
 - When sending a request, a “most specific match” operation is done to find the right connection

Example: Proxy to Proxy



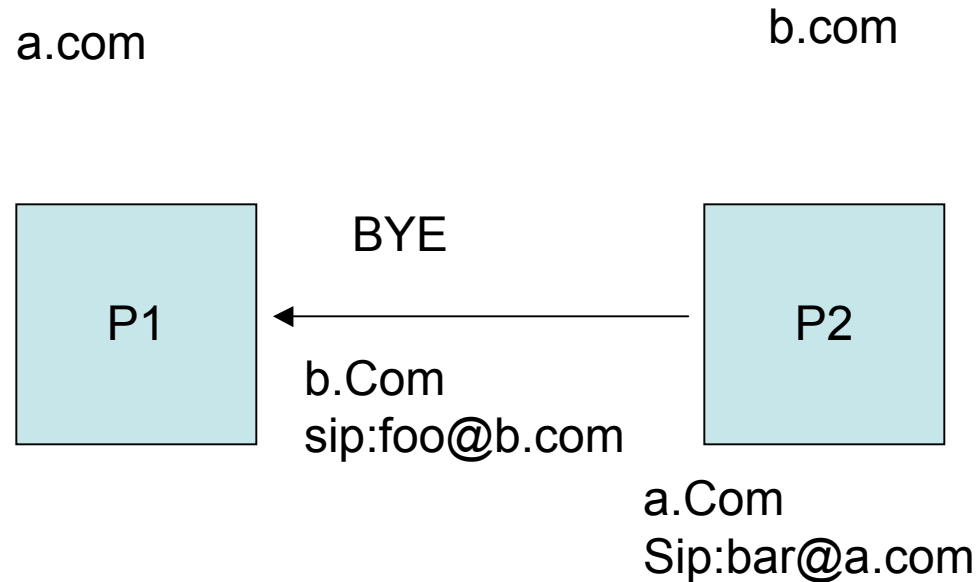
a.com wants to send a request to b.com. It finds no match for an existing connection. So, it opens a new TCP connection and begins TLS. Mutual TLS ensues. A.com links the domain b.com with that connection, and b.com links the domain a.com with its connection

Example: Proxy to Proxy



INVITE goes through, 200 OK comes back. P1 notices a downstream record-route in the 200 OK. It is sip:foo@b.com. Since this domain matches the domain of the connection on which it was sent, sip:foo@b.com is added to the list of URI for this connection. Similarly, P2, in the INVITE, noticed an upstream RR of sip:bar@a.com, and added that to its URI map.

Example: Proxy to Proxy



BYE arrives at P2. Route header has sip:bar@a.com. This is a match for the URI for the existing connection, so it is reused.

General Algorithm

- Bind a URI to a connection when
 - Of form sip:<domain> when the other side offers a TLS cert
 - When I open a new connection/flow to a URI as a result of RFC 3263
 - When a Path URI matches the domain linked to that connection
 - When a Record-Route URI matches the domain linked to that connection
- Use a connection
 - First find a URI that matches based on all URI parameters
 - Next find a URI that matches based on RFC 3261 match rules
 - Next find a URI whose domain alone matches
 - But use RFC 3263 to see if you should open another connection
 - Otherwise open a new connection

Important Notes

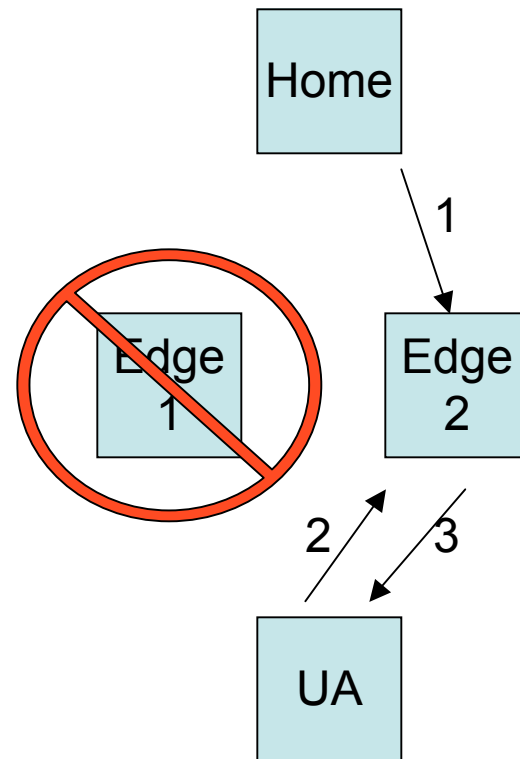
- URI to connection mapping table is state
 - Does not need to be replicated however
- State management requires element to track dialog and registration lifetimes
 - Though result of being wrong is not a big deal
- For cases where an element “forgets” and opens a connection to the wrong downstream host
 - Downstream element knows which is right host and redirects
 - Upstream element caches results of new connection attempt
 - Allows it to gracefully rebuild connection table

And Finally.....

- These two techniques
 - Proxy Discovery
 - Connection Stickiness
- Can be used in tandem with proper selection of URI to achieve mid-dialog HA for SIP Outbound

What is needed

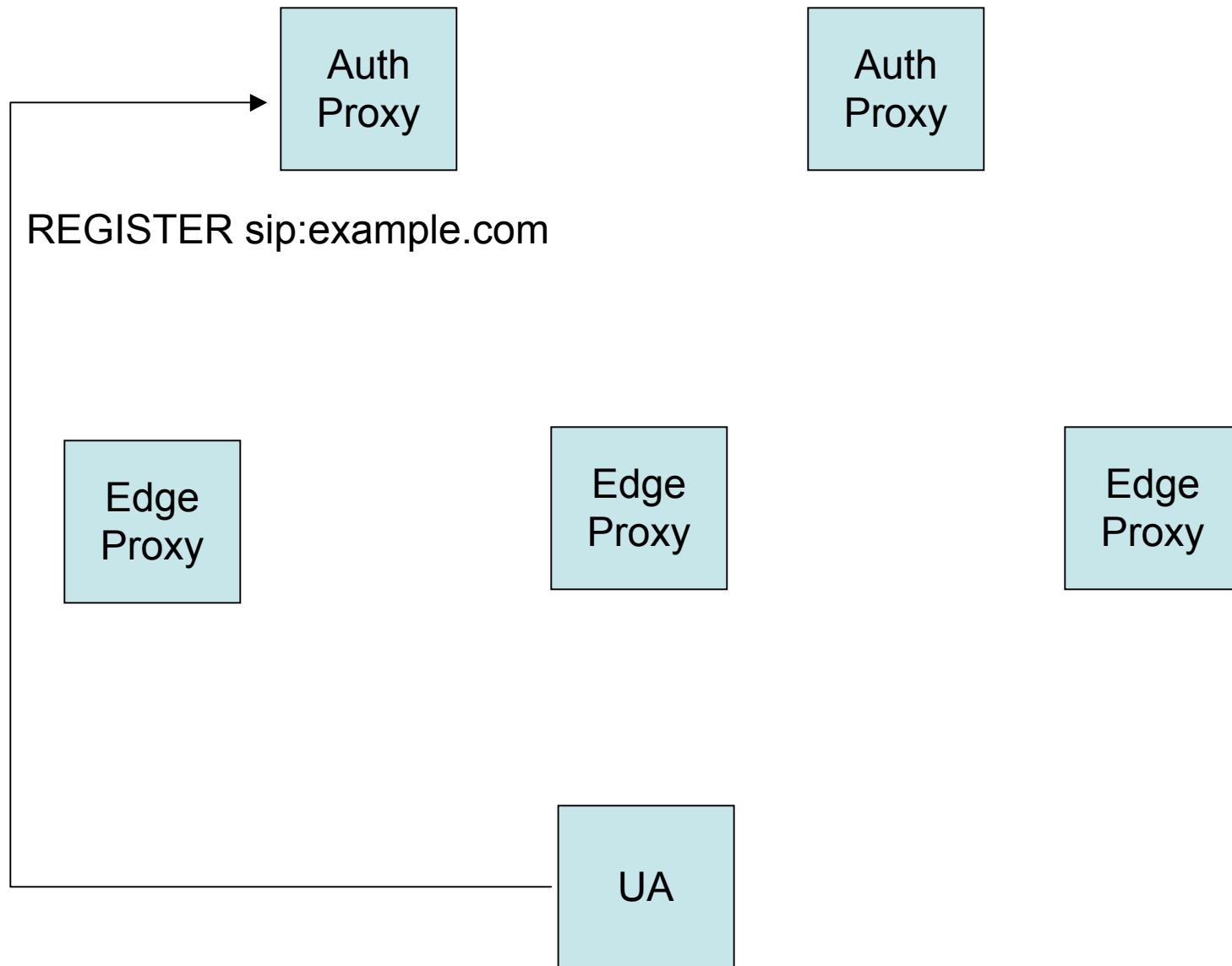
1. Home Proxy needs to reach alternate edge proxy
2. UA needs to reach alternate edge proxy
3. Alternate edge proxy needs to route to UA



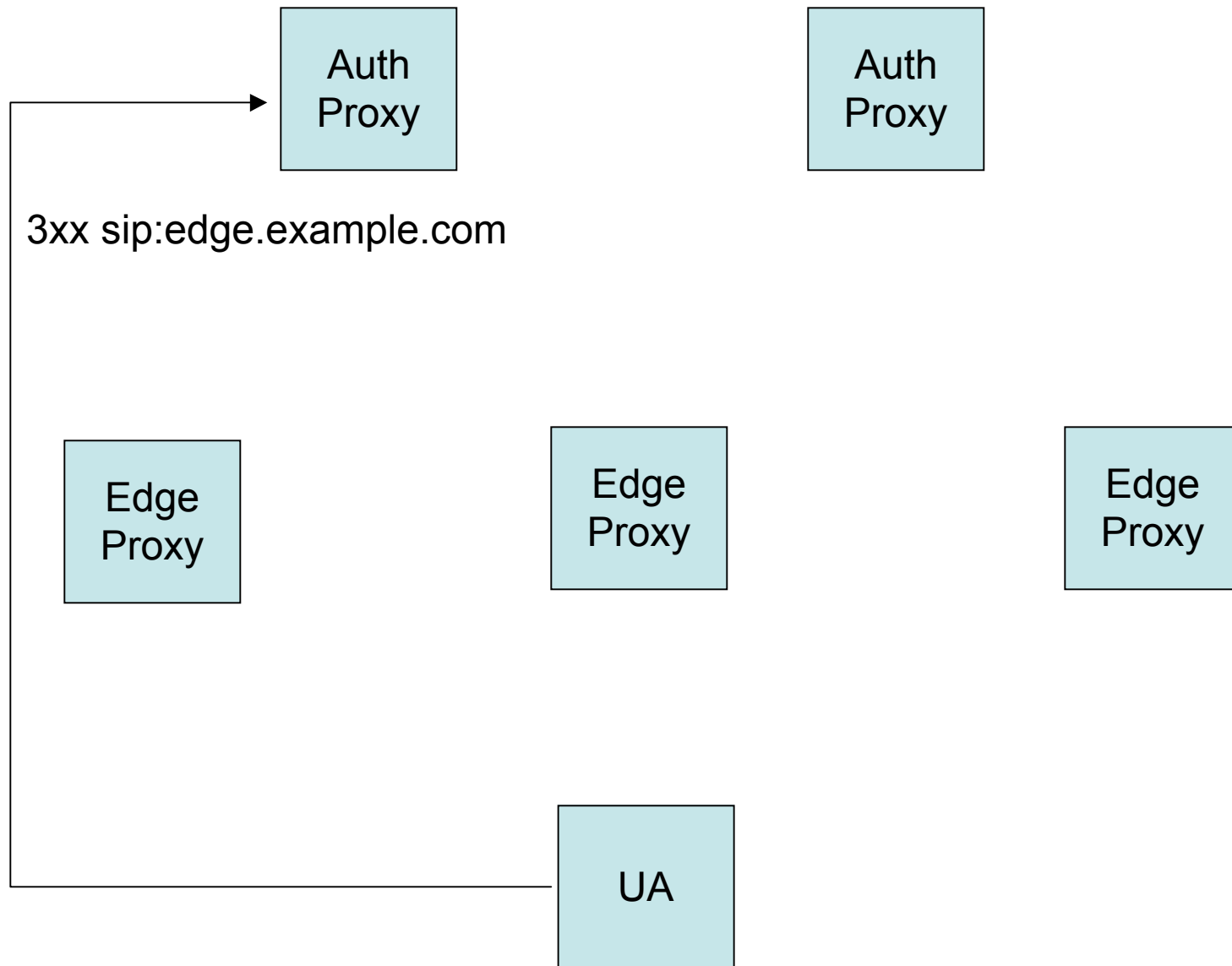
Basic Approach

- Path URI is constructed using
 - Identifier for primary and backup in user part
 - Identifier for instance ID in user part
 - Identifier for user resulting from authentication
- Primary and Backup use the same Path URI
- Connection stickiness will “glue” that Path URI to the connection from home proxy to each backup
- Record-Route is the same as that Path URI, but includes dialog gunk in a URI param
- Connection stickiness will allow mid-dialogs to go to the same proxy normally, but it’s a next-best match for backup connection

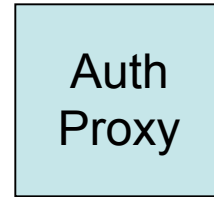
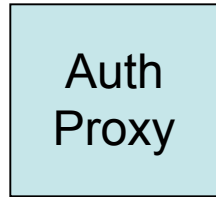
Discovery: Home Case



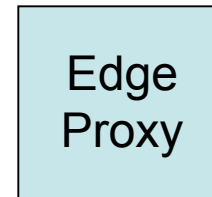
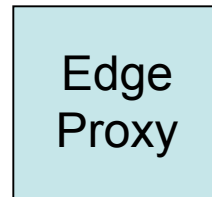
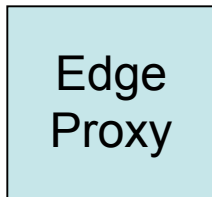
Discovery: Home Case



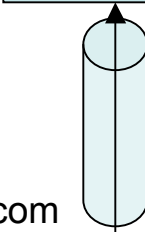
Discovery: Home Case



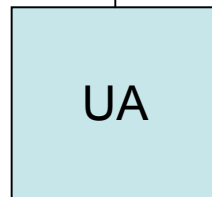
3xx sip:edge.example.com



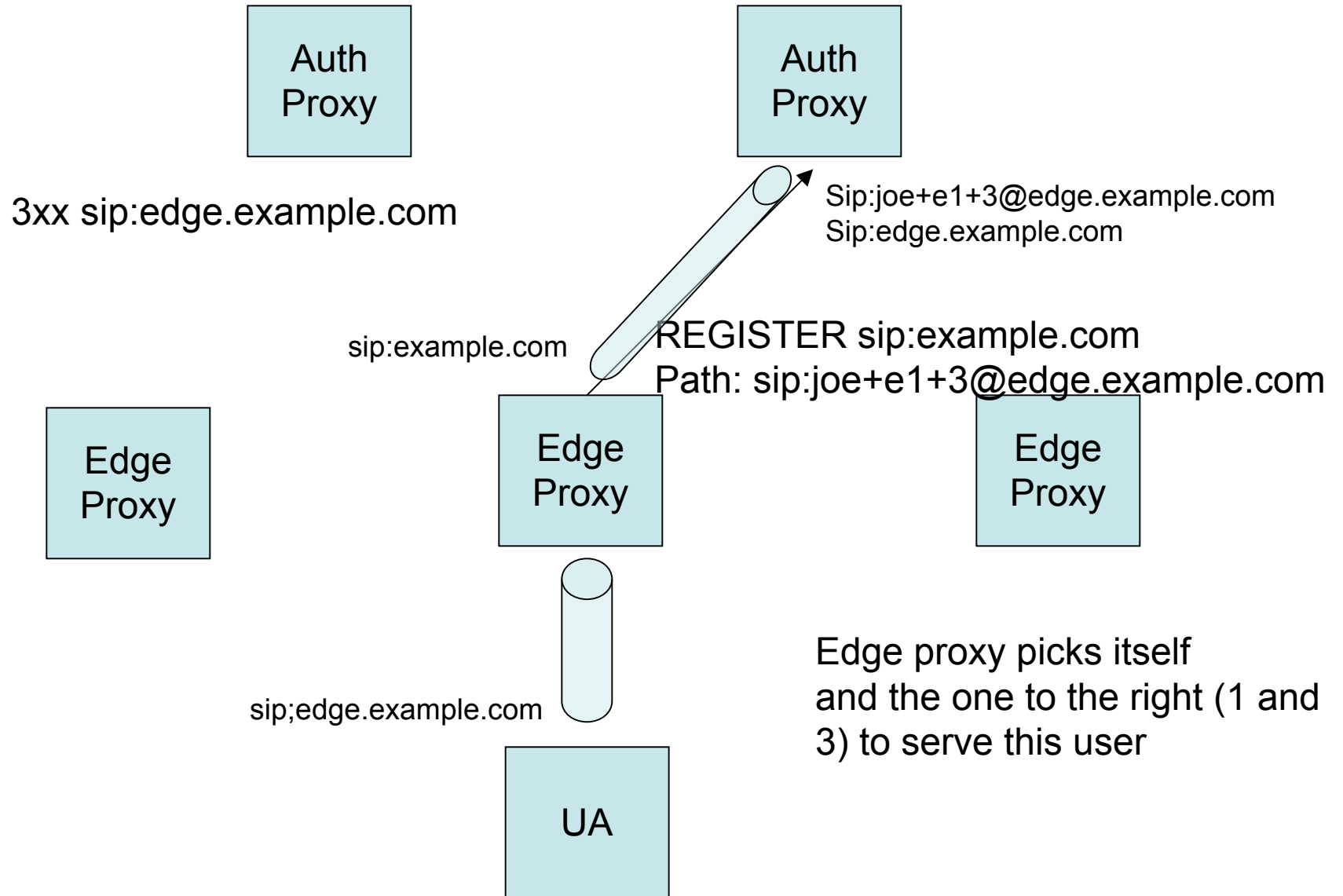
sip;edge.example.com



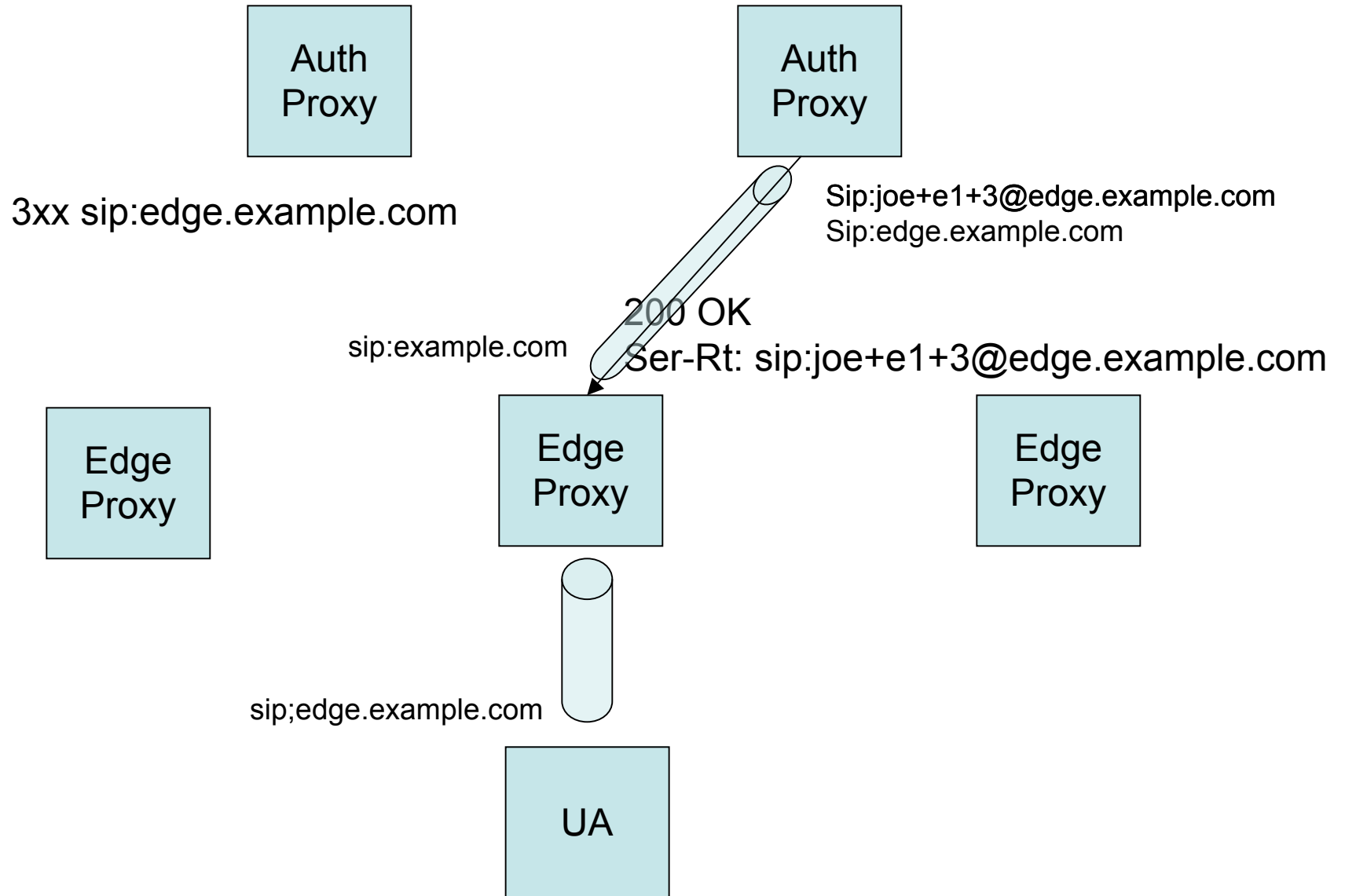
REGISTER sip:example.com
Route: sip:edge.example.com



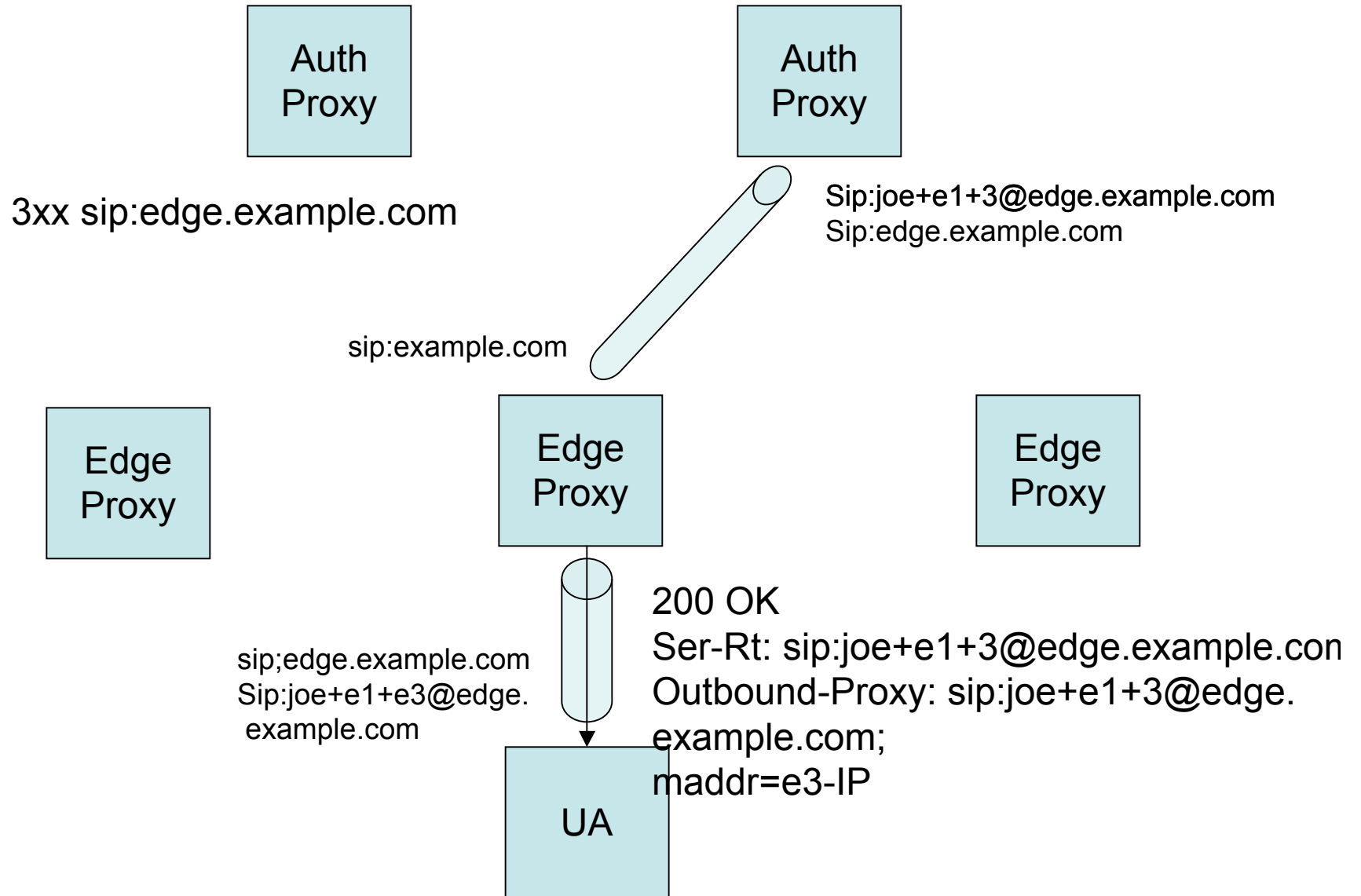
Discovery: Home Case



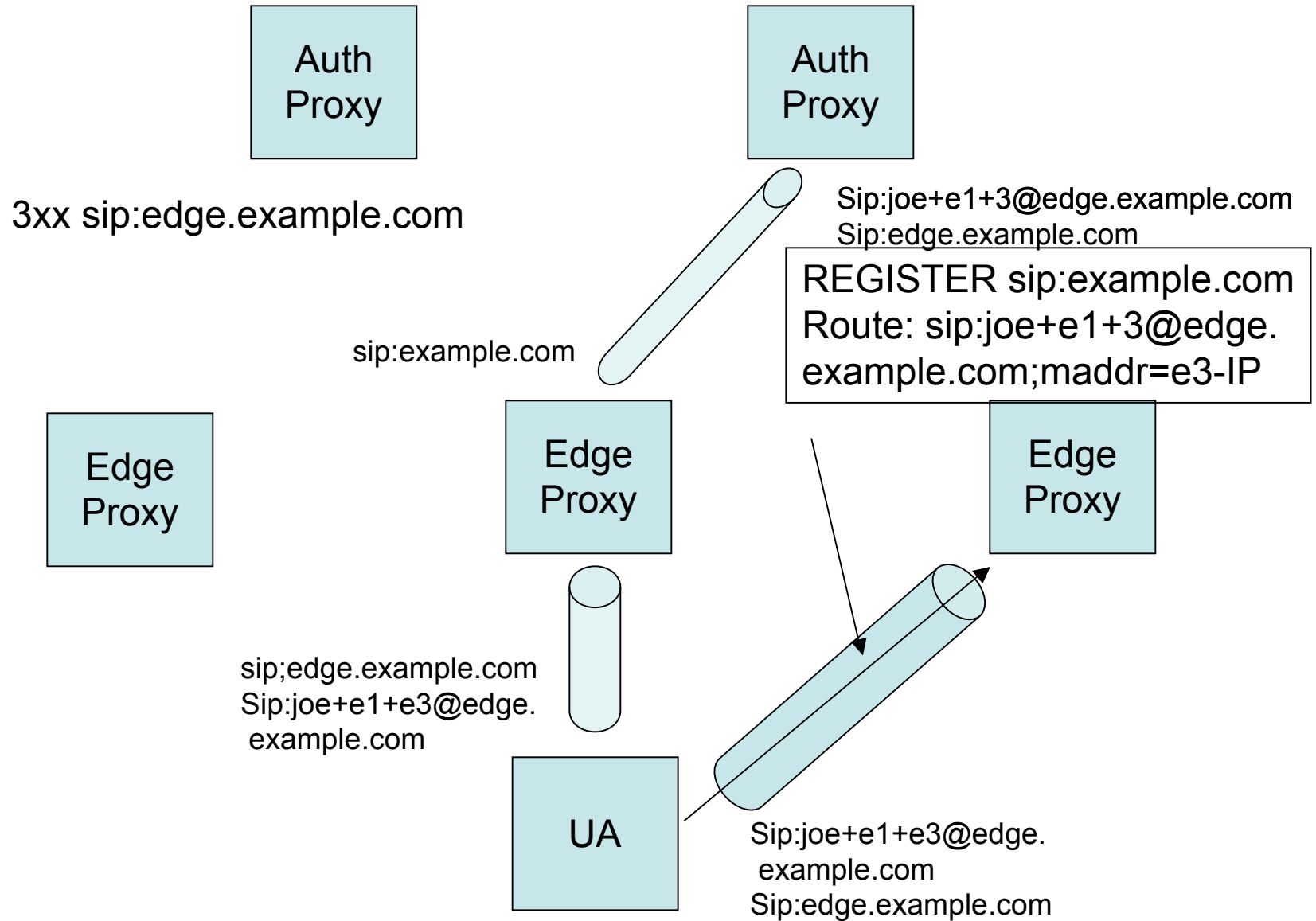
Discovery: Home Case



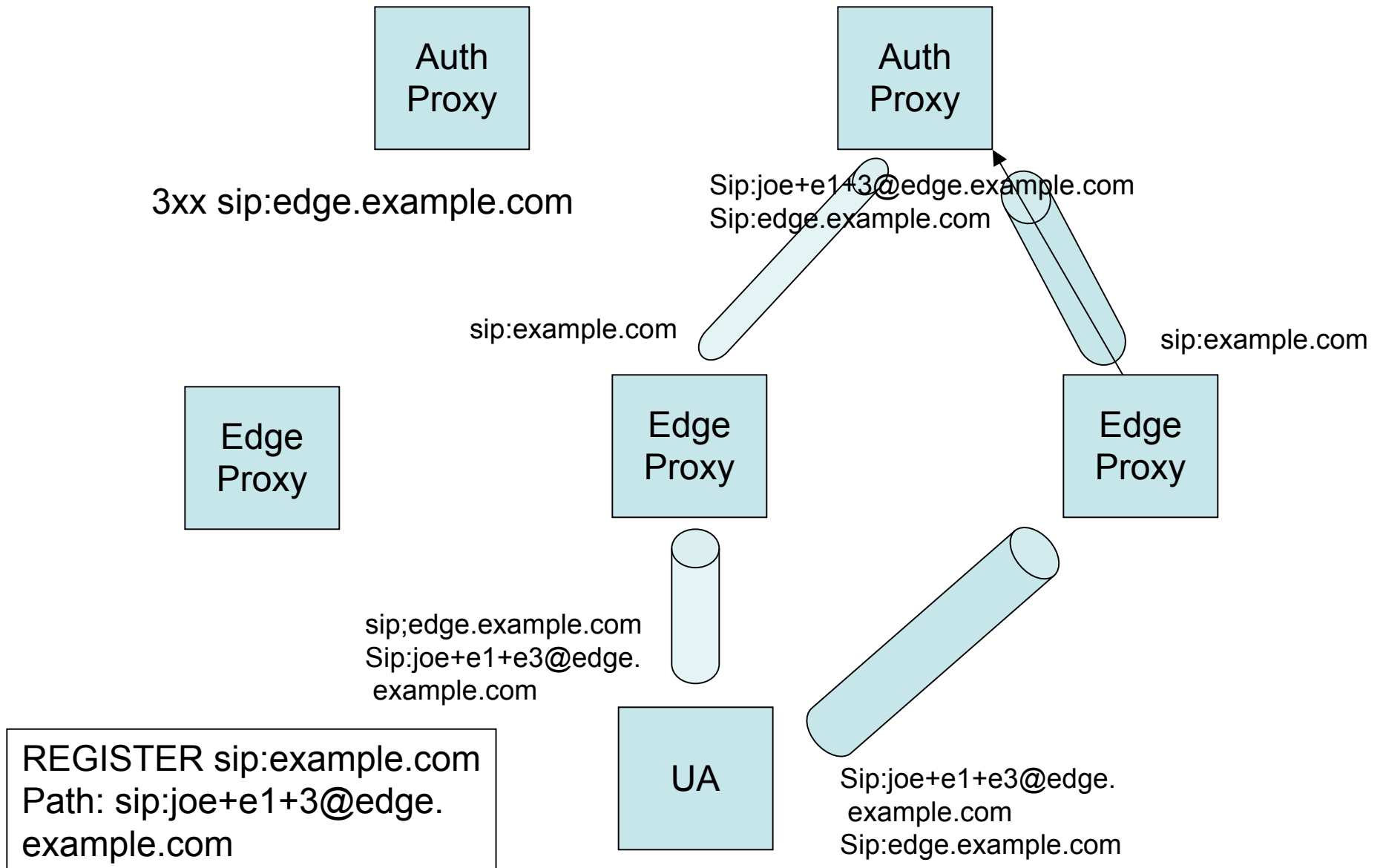
Discovery: Home Case



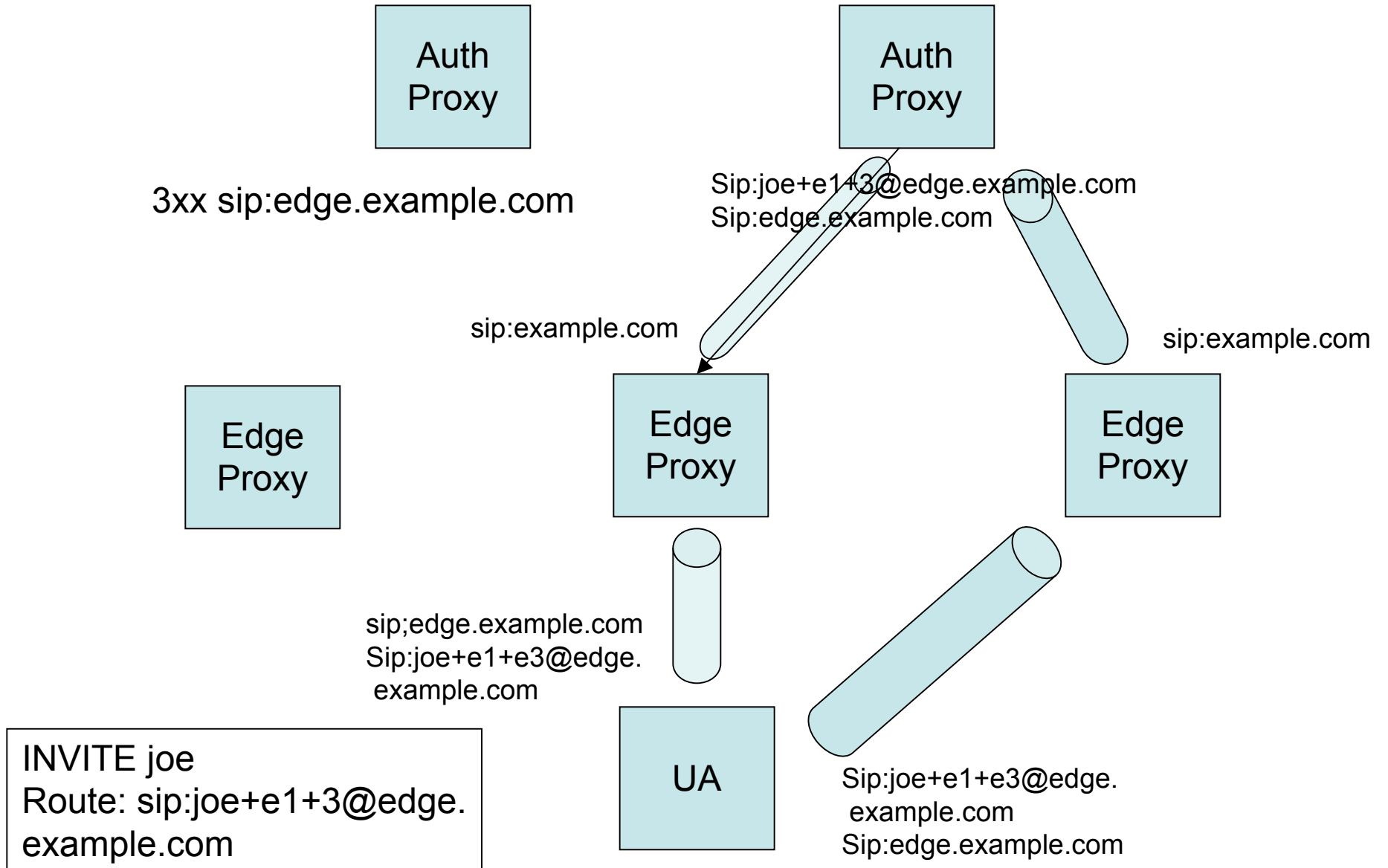
Discovery: Home Case



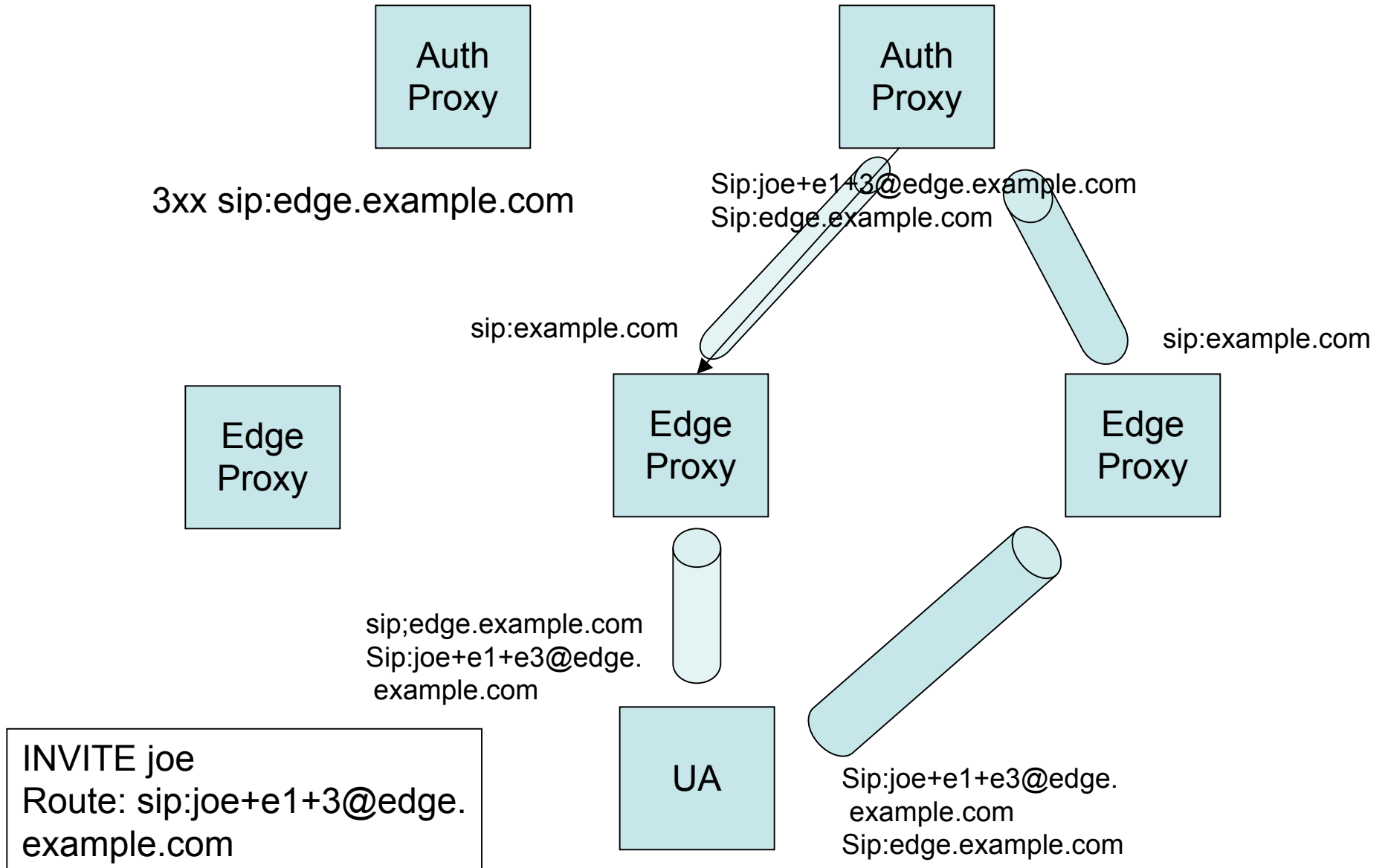
Discovery: Home Case



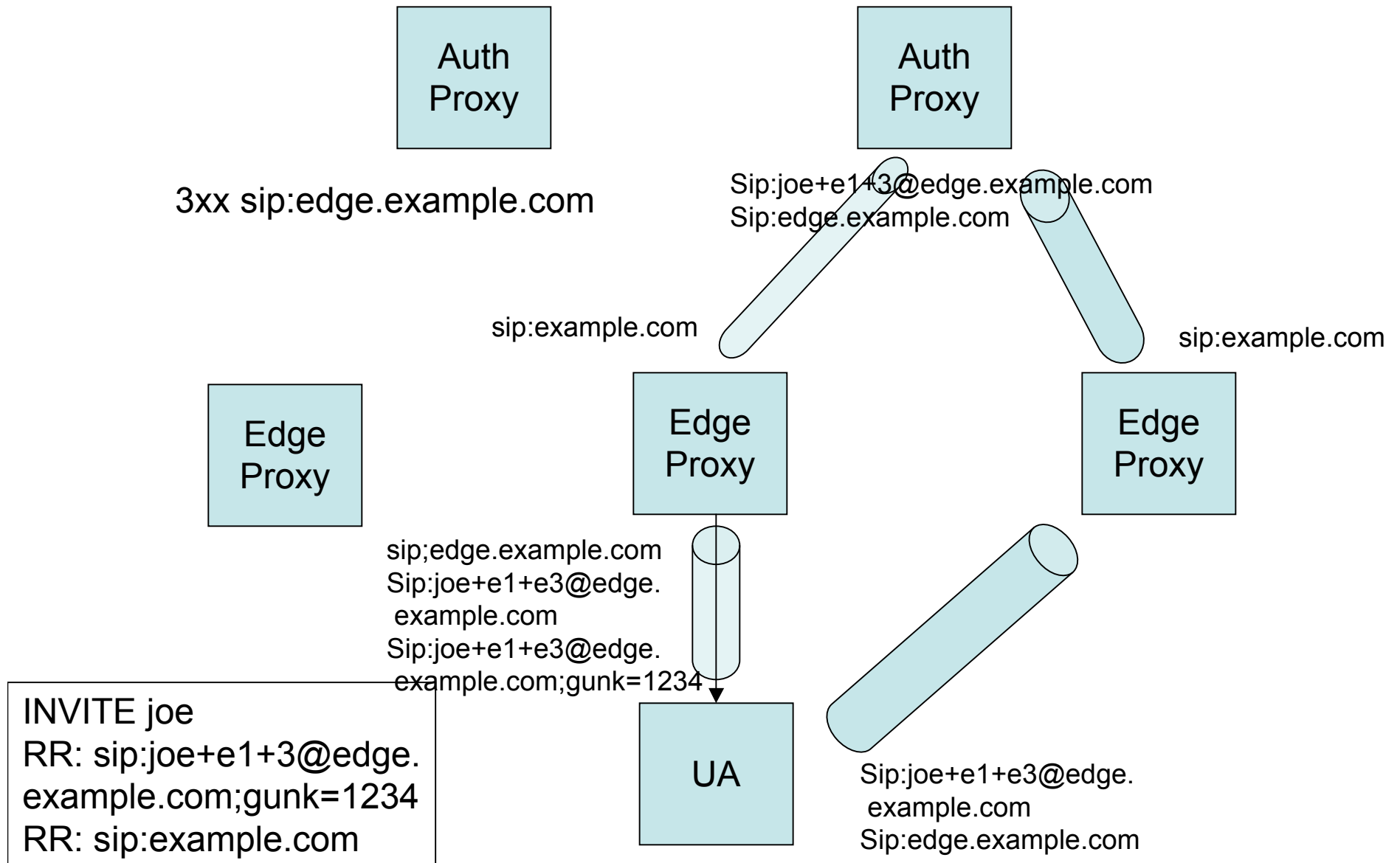
Discovery: Home Case



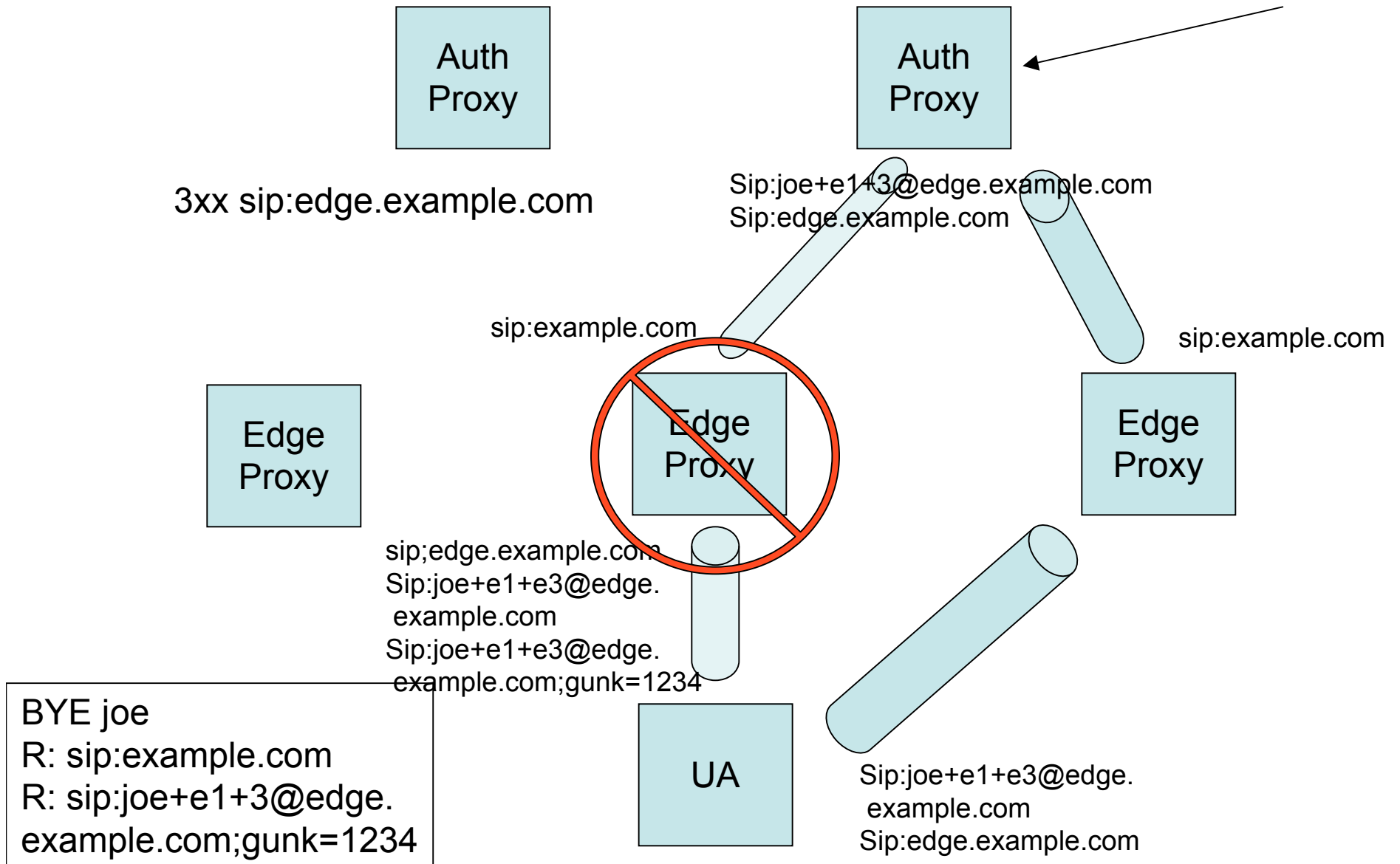
Discovery: Home Case



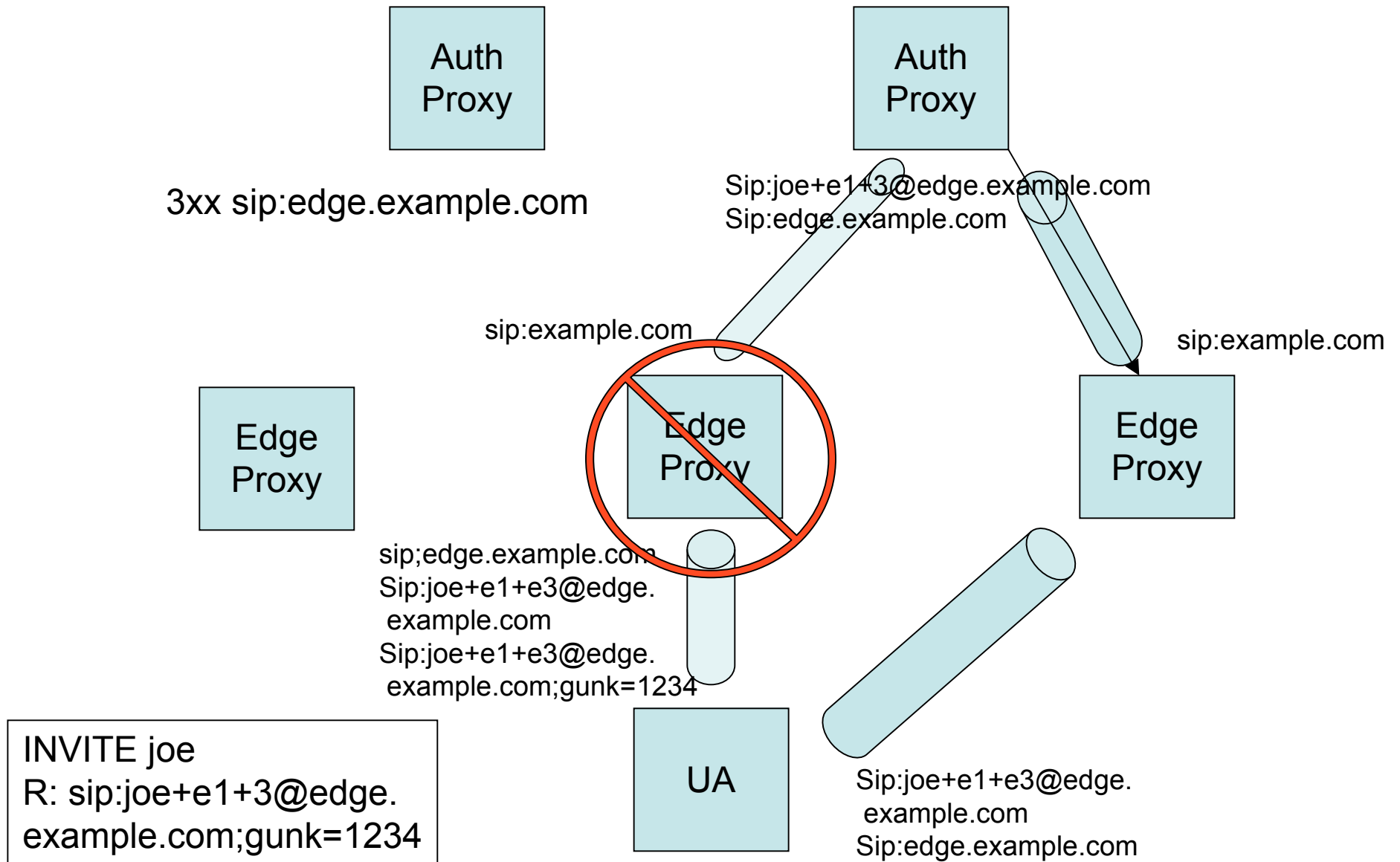
Discovery: Home Case



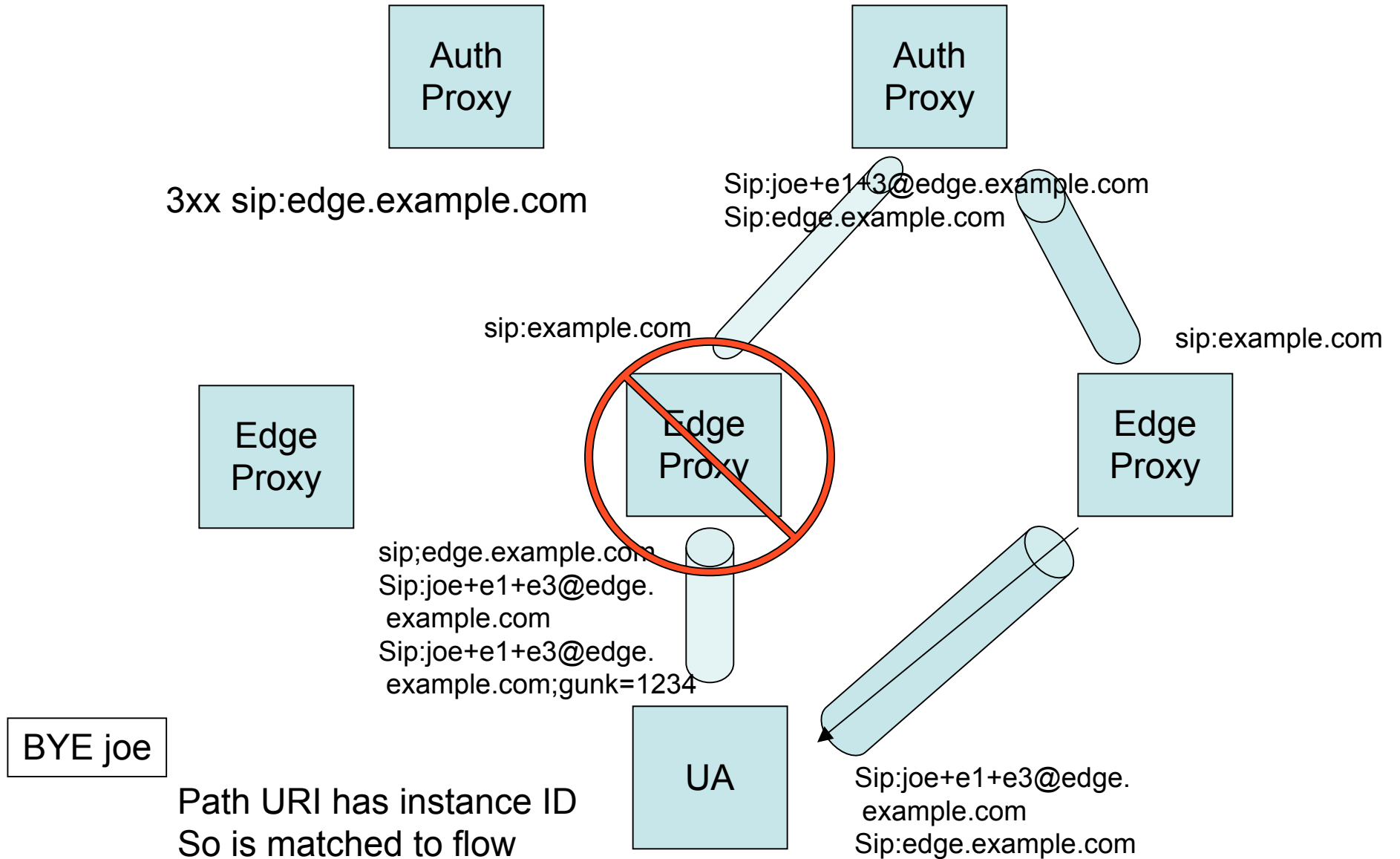
Discovery: Home Case



Discovery: Home Case



Discovery: Home Case



Path Forward

- Get some feedback
- Split draft into three in order to simplify and detangle
 - Connection stickiness draft
 - Outbound discovery draft
 - Mid-dialog HA draft