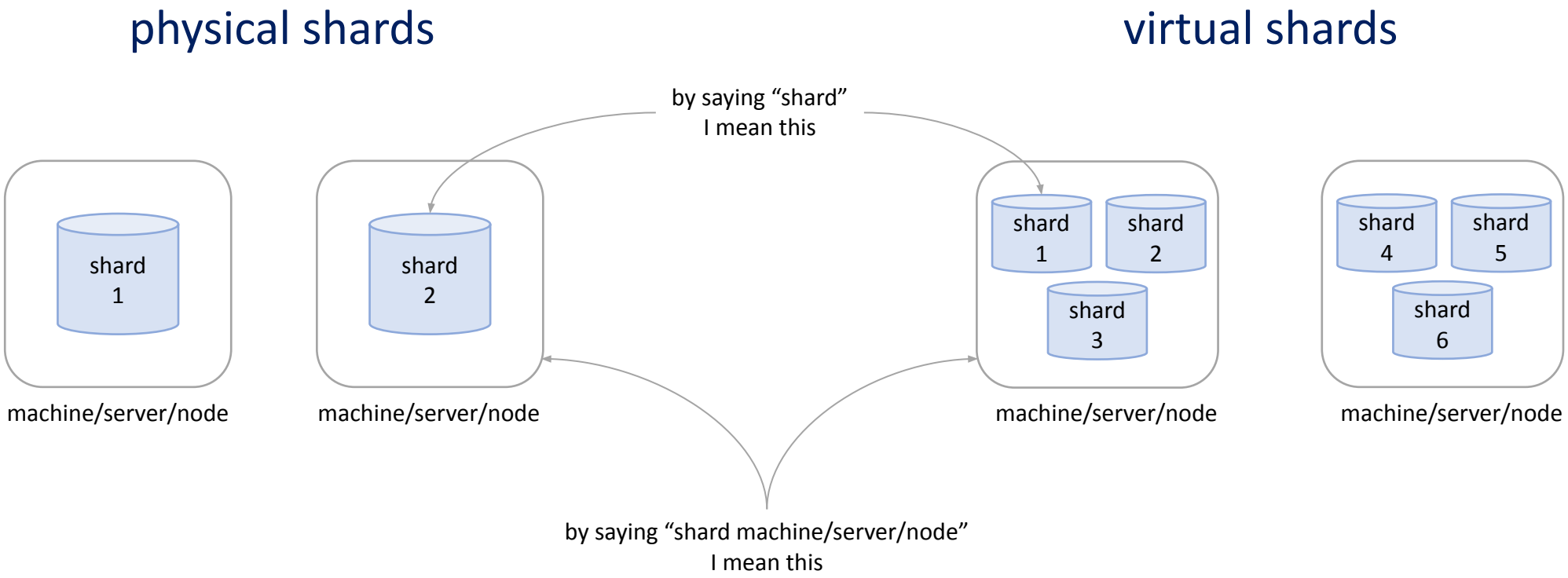
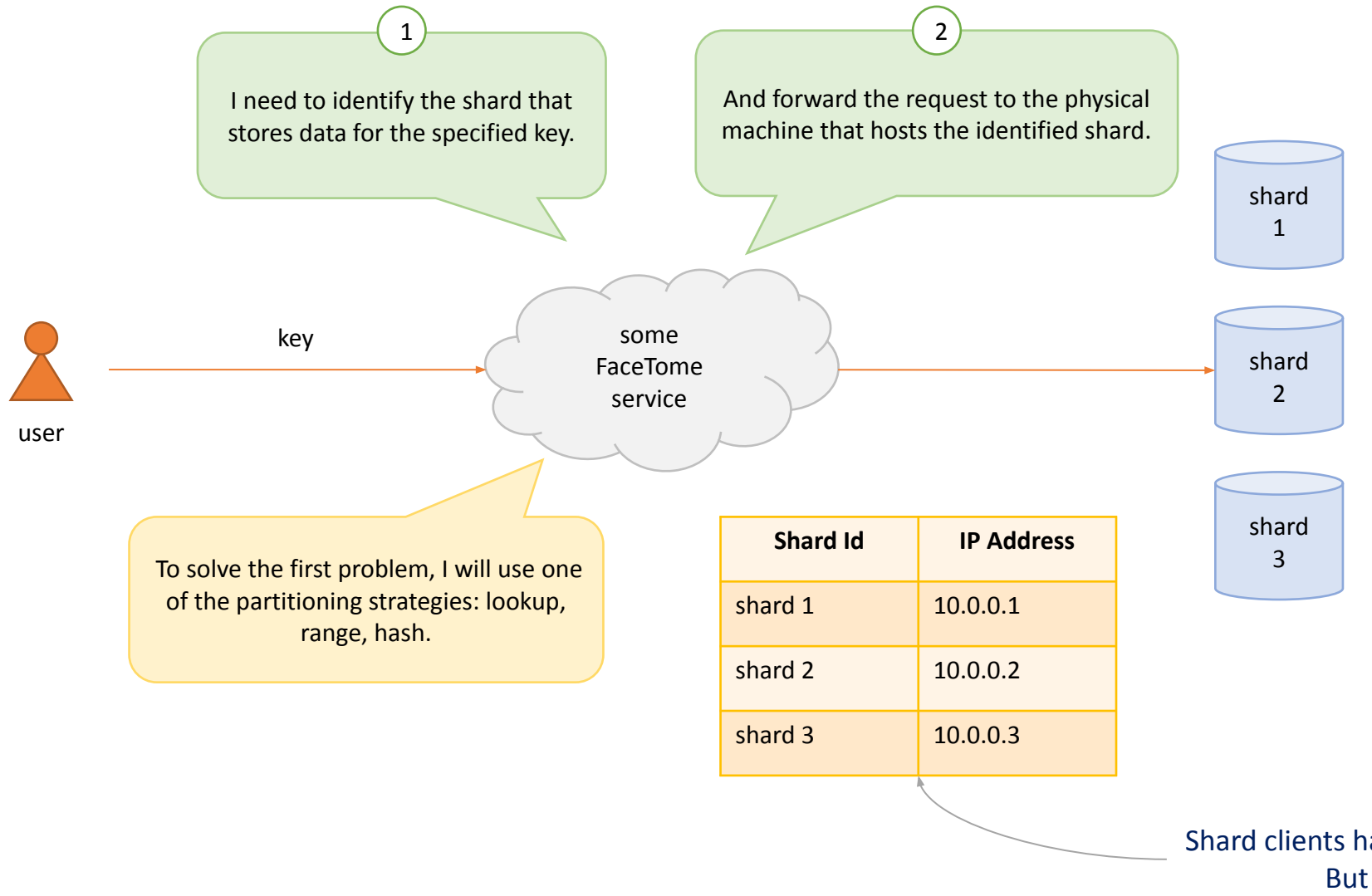
Request routing



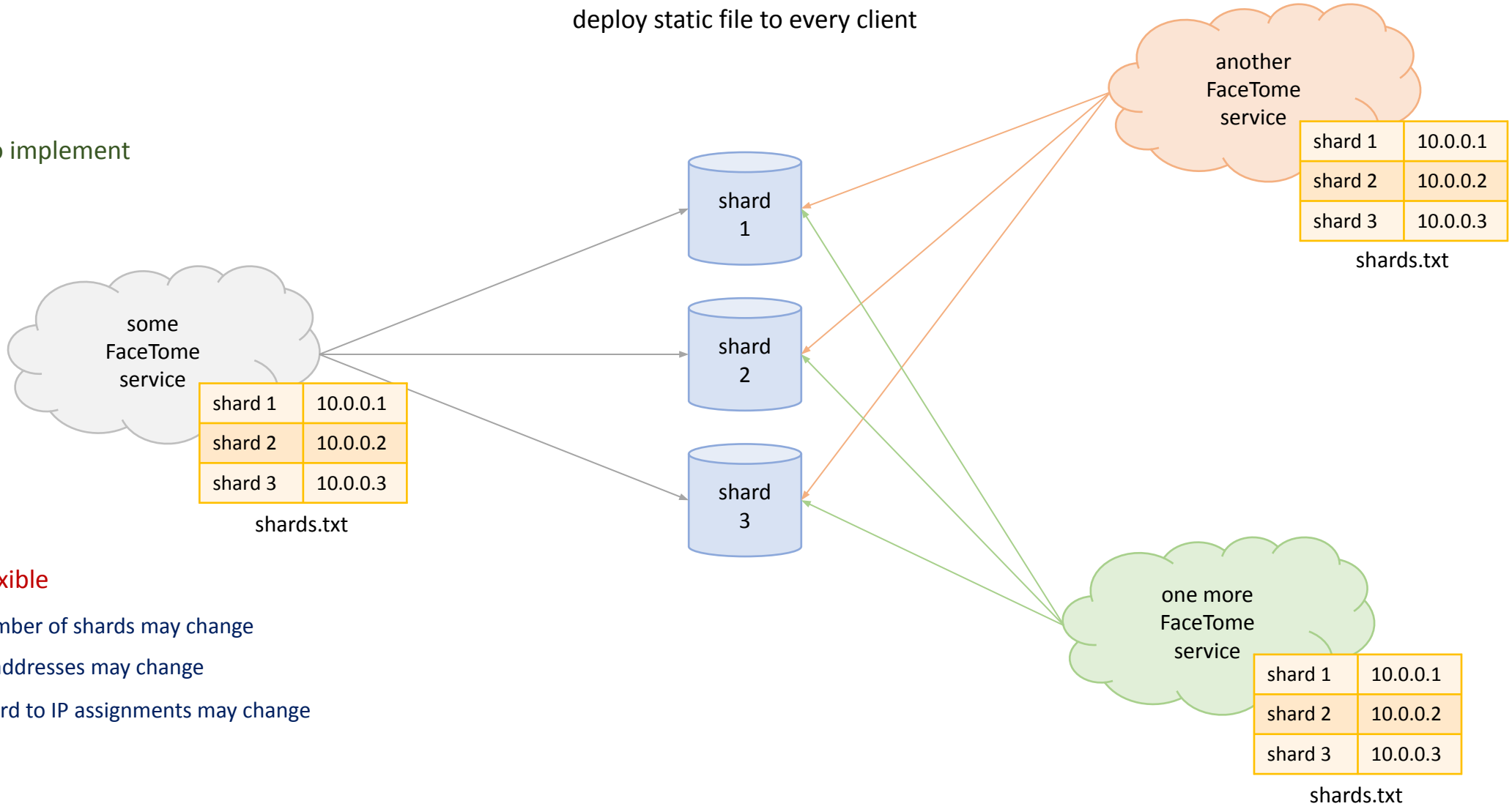
Request routing



Request routing

option 1 (a)
every client knows about shards
deploy static file to every client

- easy to implement



- not flexible
  - number of shards may change
  - IP addresses may change
  - shard to IP assignments may change

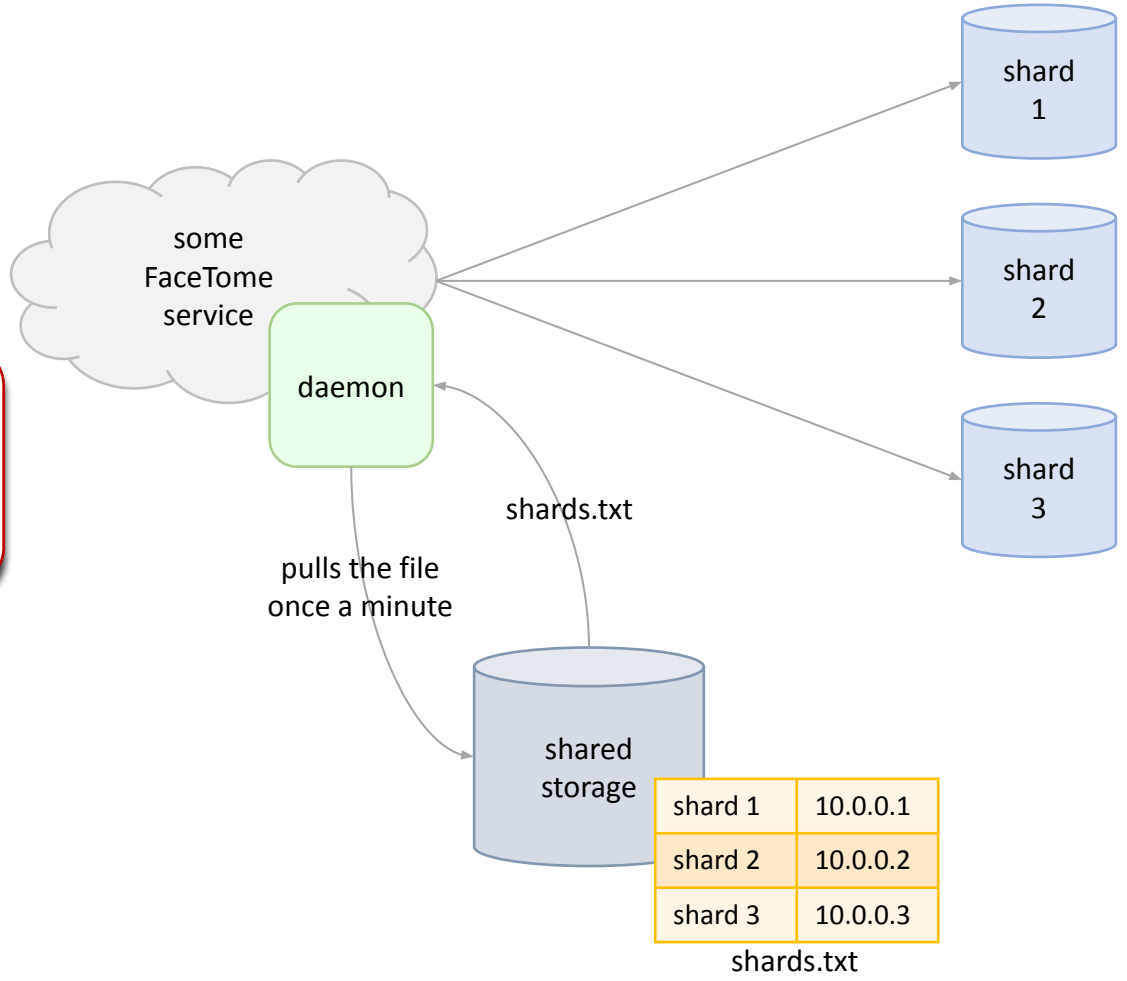
Request routing

option 1 (b)
every client knows about shards
put static file into a shared storage

- more flexible

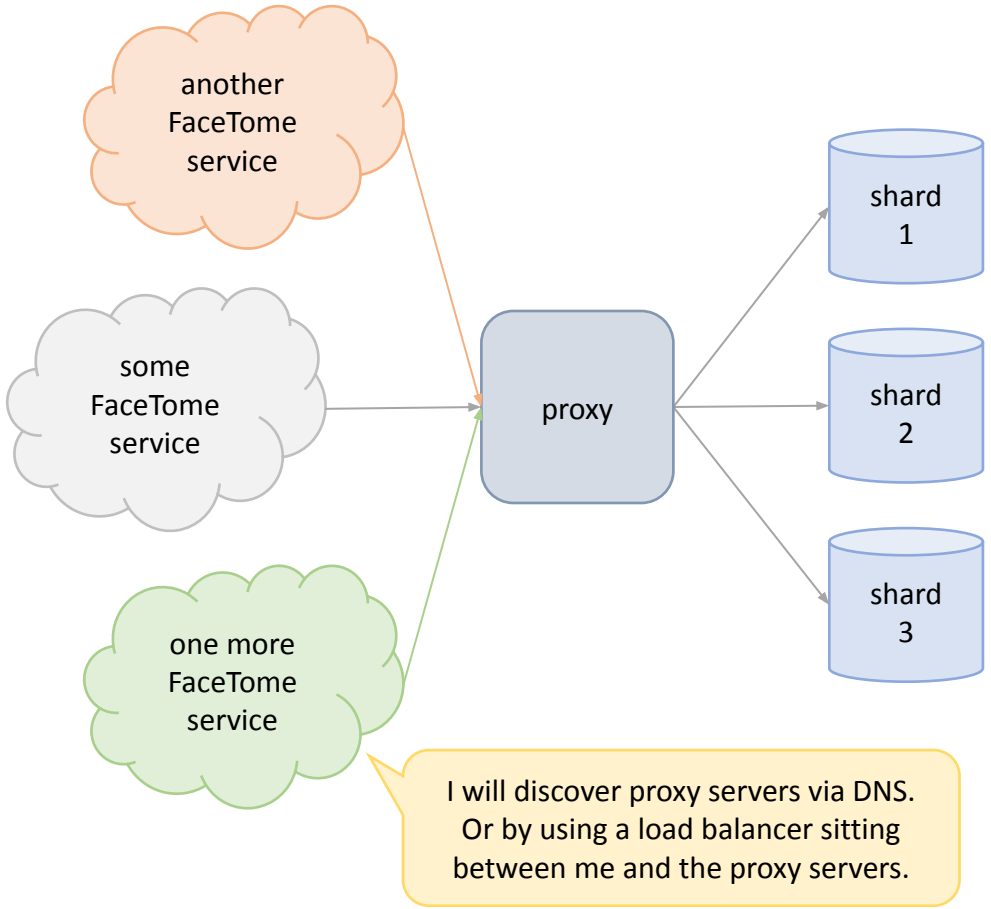
It is necessary to constantly monitor the health of the daemon process. Otherwise, the mapping may become outdated and requests will be forwarded to the wrong location.

- client gets more complicated

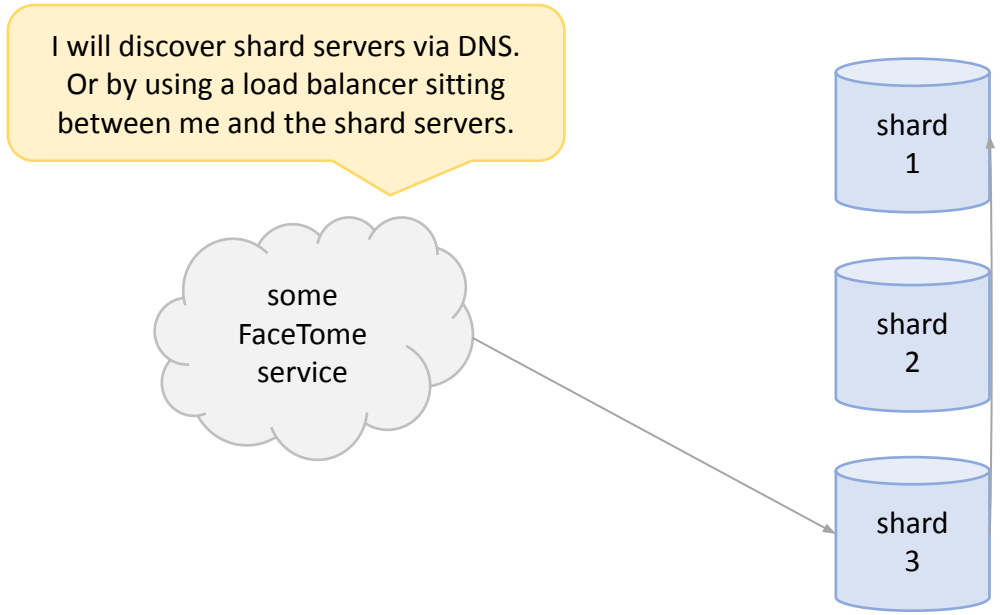


Request routing

option 2
proxy
reverse proxy knows about shards



option 3
peer-to-peer
shards know about shards



Request routing

How do proxy machines and shard servers
learn about each other?

static list of IP addresses

deploy the list directly to every proxy/shard machine
(e.g. using configuration management tools, such as Chef, Puppet)

use a daemon process to fetch the list from a shared storage
(e.g. S3)

share through DNS
(e.g. via TXT records)

service registry
(configuration service)

classic client-side discovery mechanism

gossip protocol

requires a list of seed nodes to bootstrap the gossip process