Distributed Systems

ID2201

peer-to-peer

Johan Montelius

Idéa

- use resources in edge of network
  - computing
  - storage
  - communication

Computing

seti@home

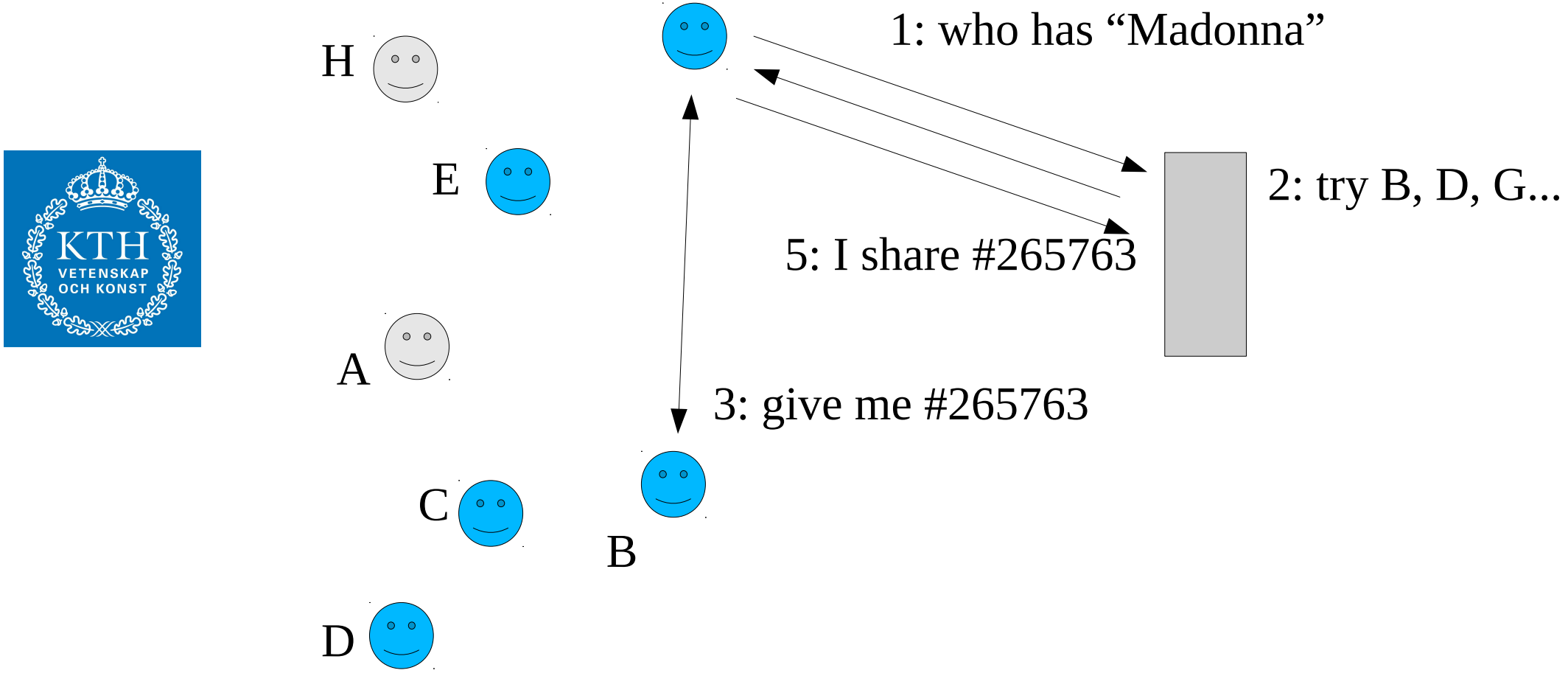
- central server
  - millions of clients
  - hundred of thousands active
- super computer
  - hundreds of TeraFLOPS
  - one of the largest computations performed
- continued in the BOINC project

File sharing

Napster

- First large scale peer-to-peer file sharing system - 1999
- Used a central server to store index of all files.
- Clients copied files peer-to-peer.

Napster

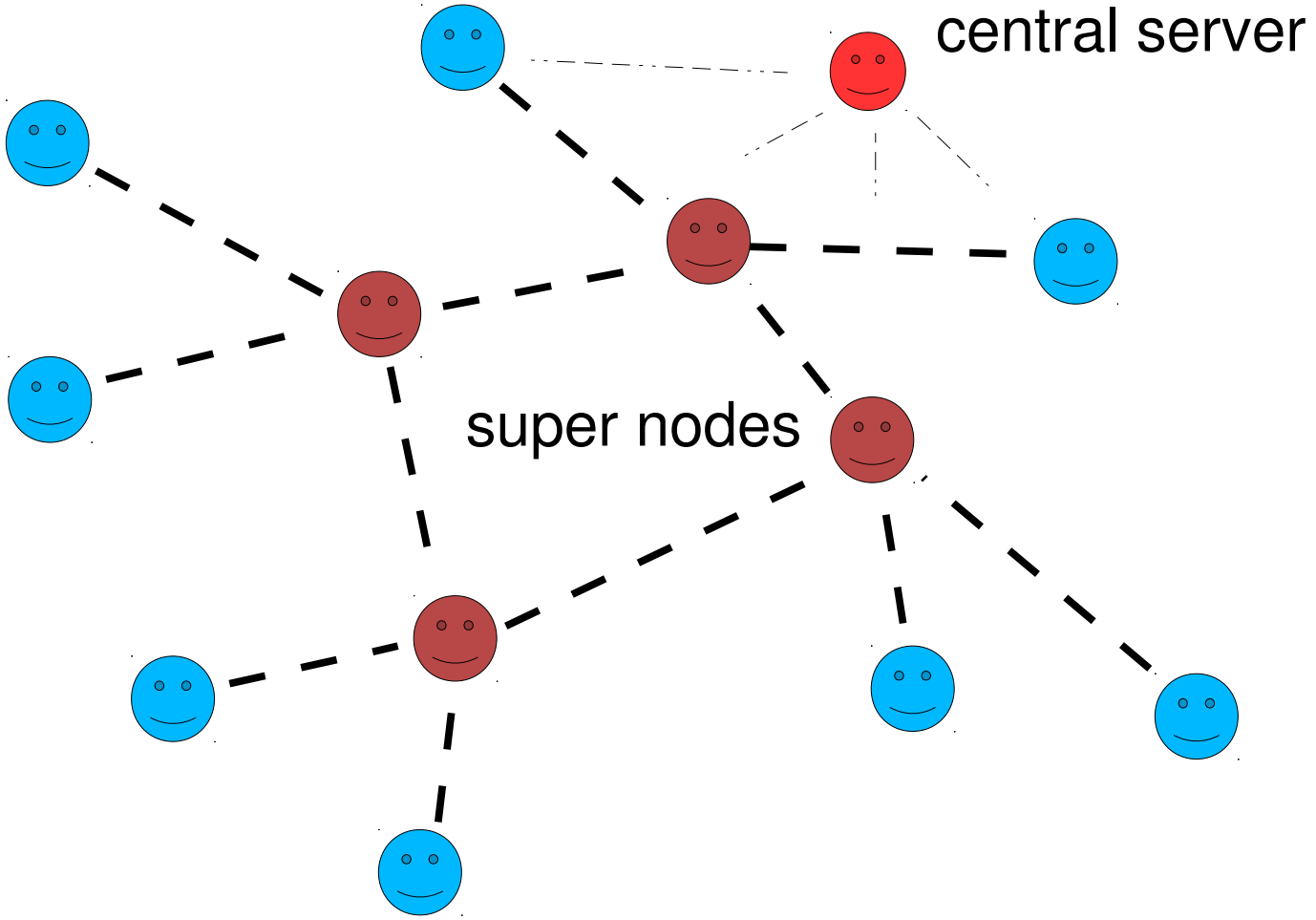

H
E
A
C
B
D
1: who has “Madonna”
2: try B, D, G...
5: I share #265763
3: give me #265763

Napster

- Central server
  - knows everything
  - needs to be alive
  - can easily be replicated
- File transfer
  - limited by client upload capacity
- Problems
  - copyright issues
  - why share
  - is it the correct file

Next step

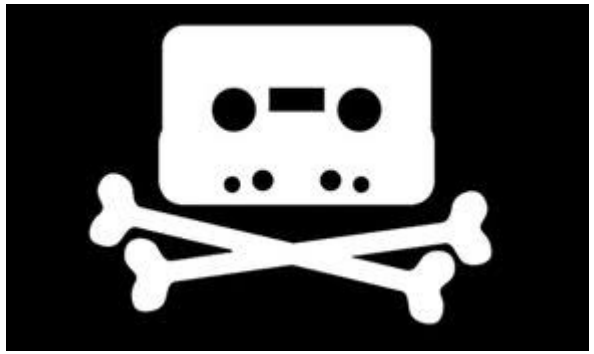
Kazaa



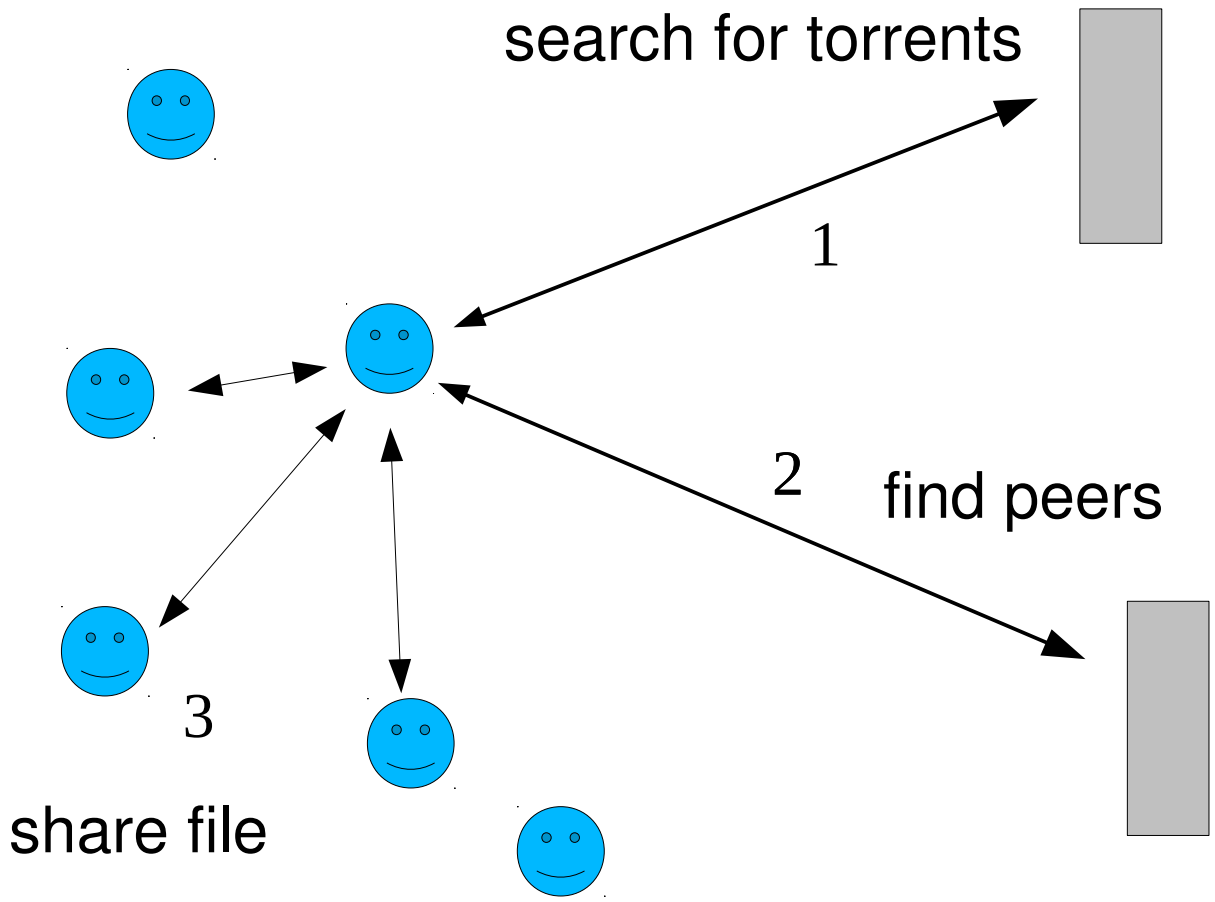
Kazaa

- FastTrack (closed protocol)
  - super nodes, responsible for indexing
  - central server, blacklist of super nodes
  - regular nodes, connects to local super node
- Integrity is checked by hash function.
  - not very strong
- Money made on
  - advertising
  - ...

Time of the pirates



BitTorrent


search for torrents
1
torrents and description of content
2
find peers
3
share file
tracker, who wants to participate

BitTorrent

- torrent
  - trackers to use
  - name of content
  - size and number of segments
  - hash codes of segments
- tracker
  - provides list of peers
  - could be helpful in suggesting network close peers

BitTorrent

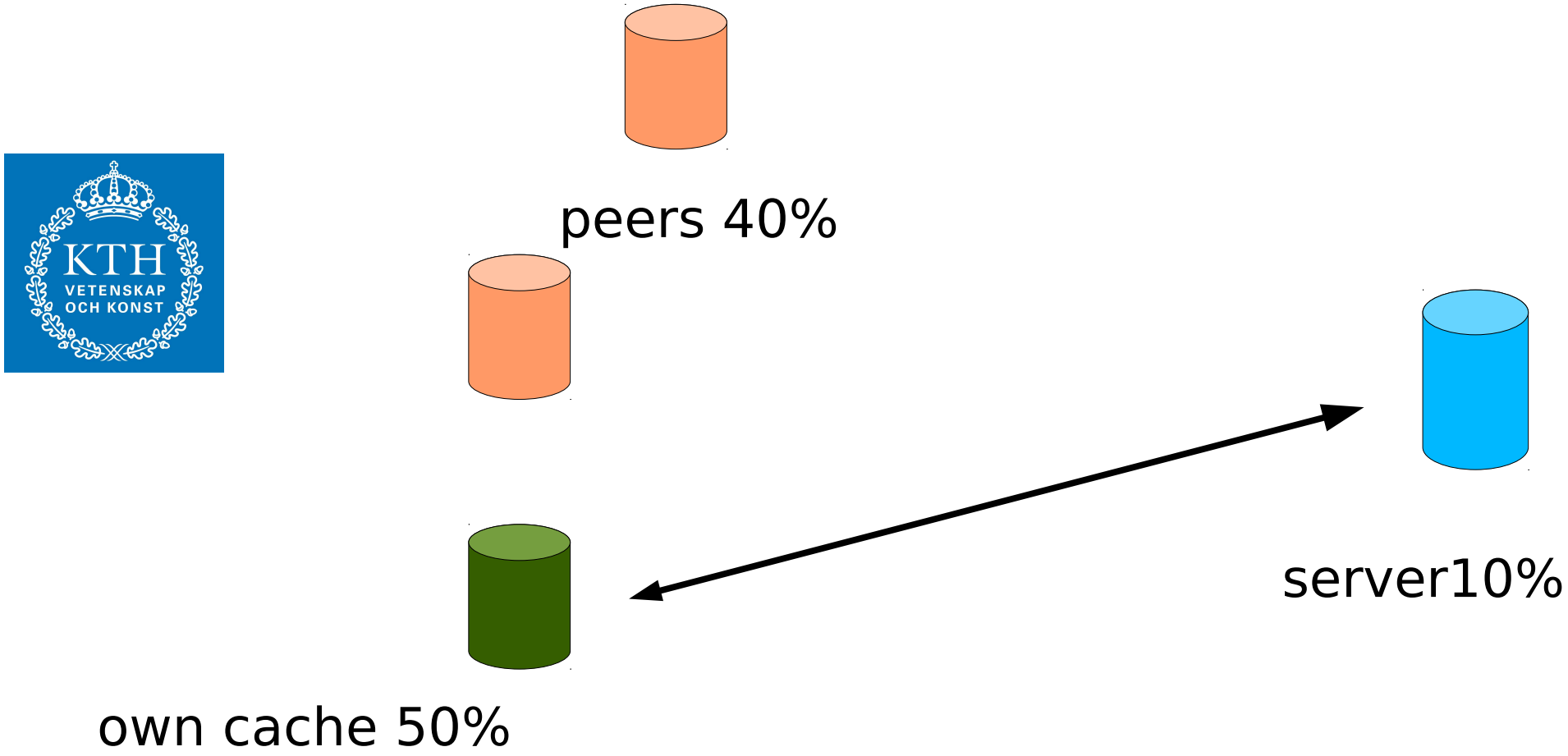
- Query peers to find who has what.
- Tit-for-tat
  - per file, not on total
  - if you don't get something, why share
- Rarest first
  - rare segments are valuable
- Multiple peers
  - change if connection is slow
  - choke if you don't get anything back

Magnet links and DHT

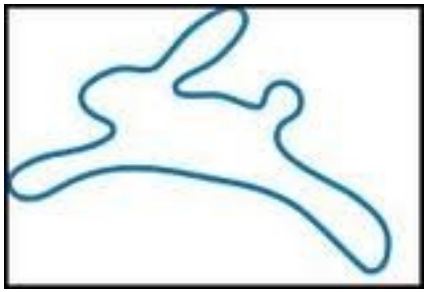

```
magnet:?
xt=urn:btih:d2438d70a205566b2baf8eb45
4c1270237b1dbcf&
dn=Rick+Astley+Never+Gonna+Give+You+U
p.mp3&
tr=udp%3A%2F%2Ftracker.ccc.de%3A80
```

All the music, all the time.

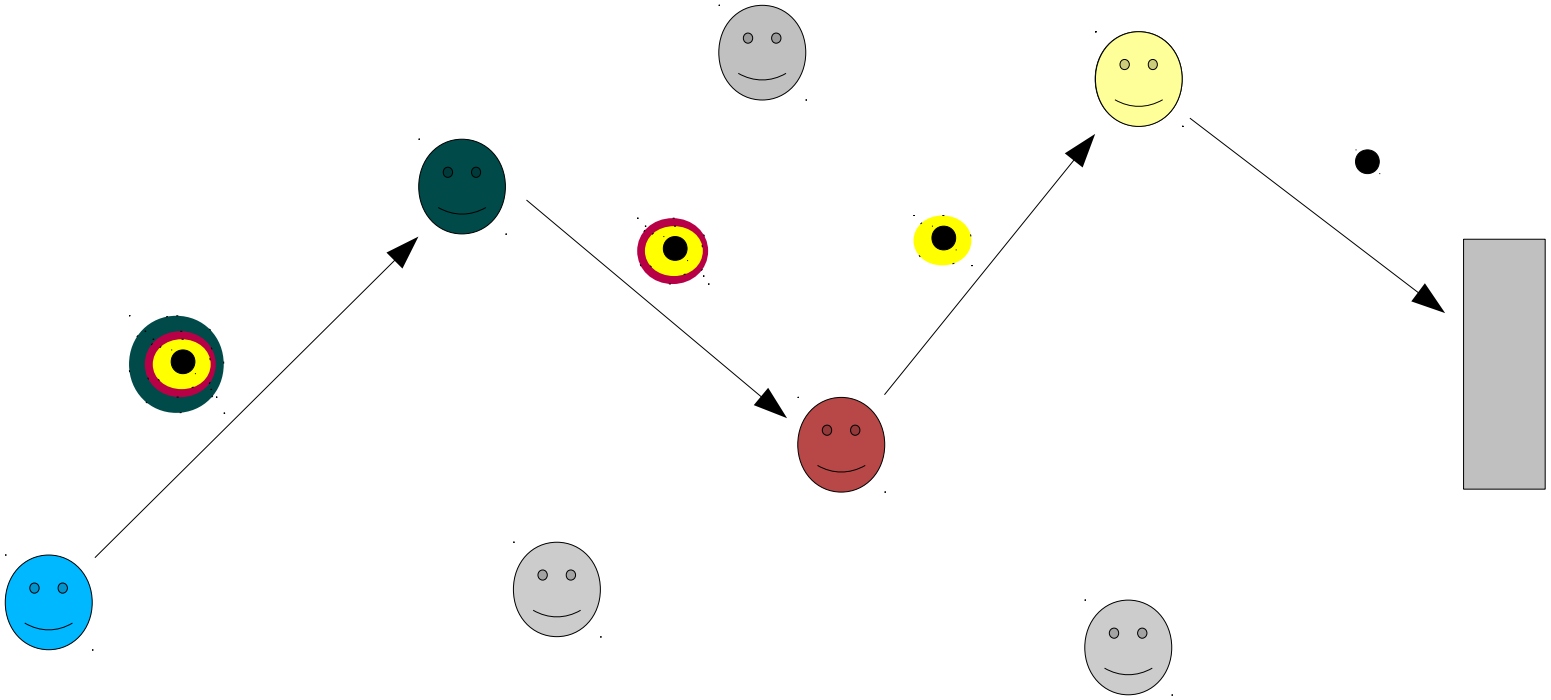
Spotify



Some privacy



Tor – anonymous routing



P2P middleware

- Function
  - add, remove, locate and communicate with resources in a network
- Requirements
  - global scale, millions of nodes
  - dynamic availability
  - integrity, privacy, anonymity, deniability

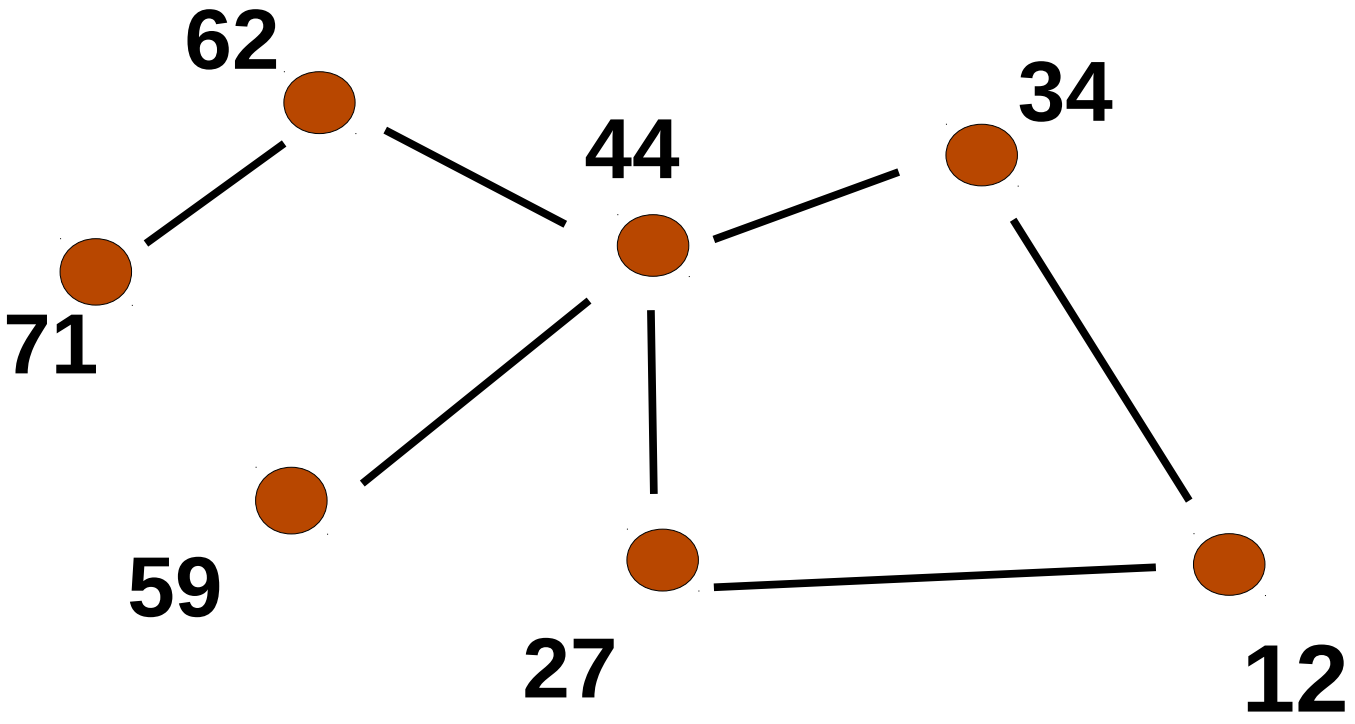
Objects and routing

- Why not use IP?
  - scale: 2^{32} nodes
  - structured network, costly to add new objects
  - slow updates
  - no mobility (well Ipv6)
  - trust, privacy

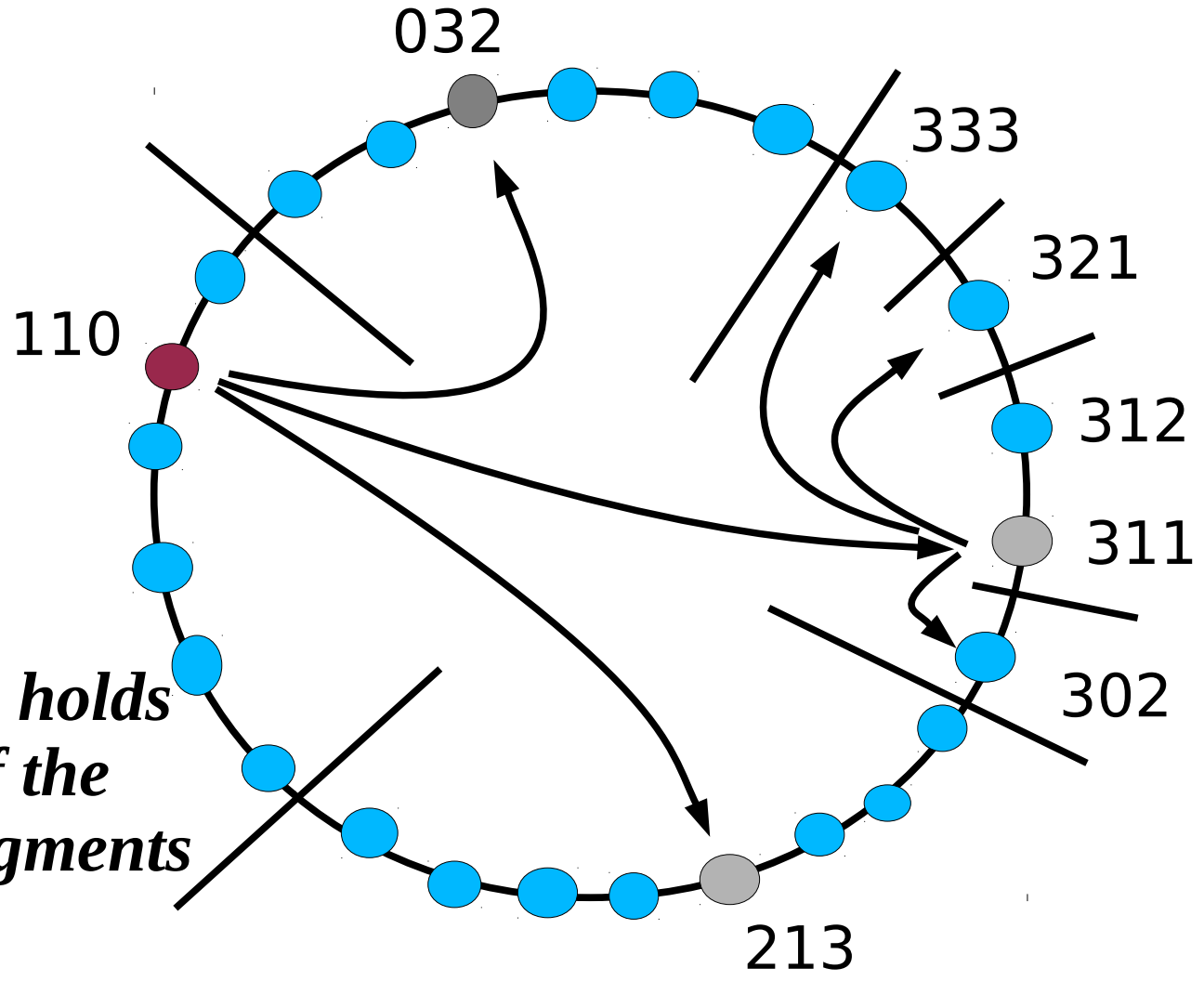
Overlay routing

- name space: global unique identifier (GUID)
- structured or unstructured
  - pay when you add nodes and objects
  - pay when you search for objects
- fault tolerance and consistency
  - replication

structured overlay



Pastry routing (example with k=4 not 16)

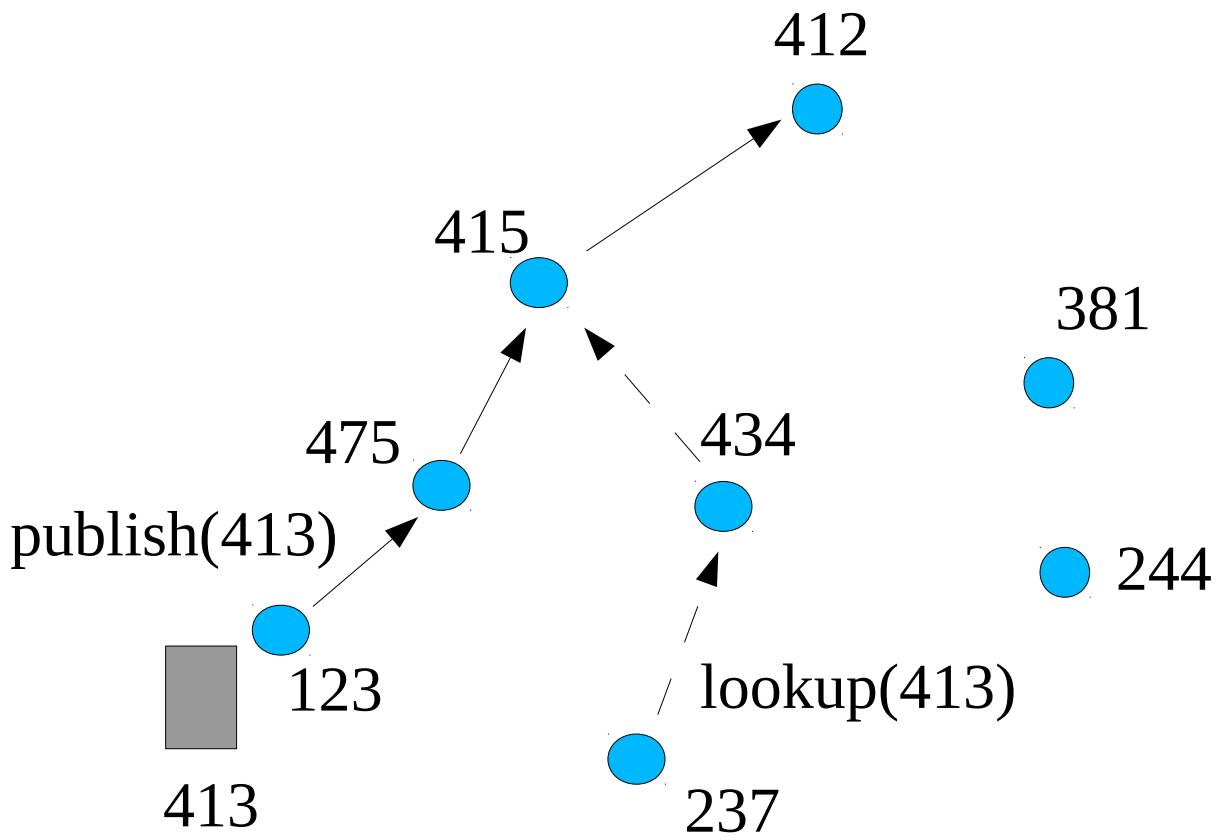


second row of 311 holds entries for each of the three other sub segments

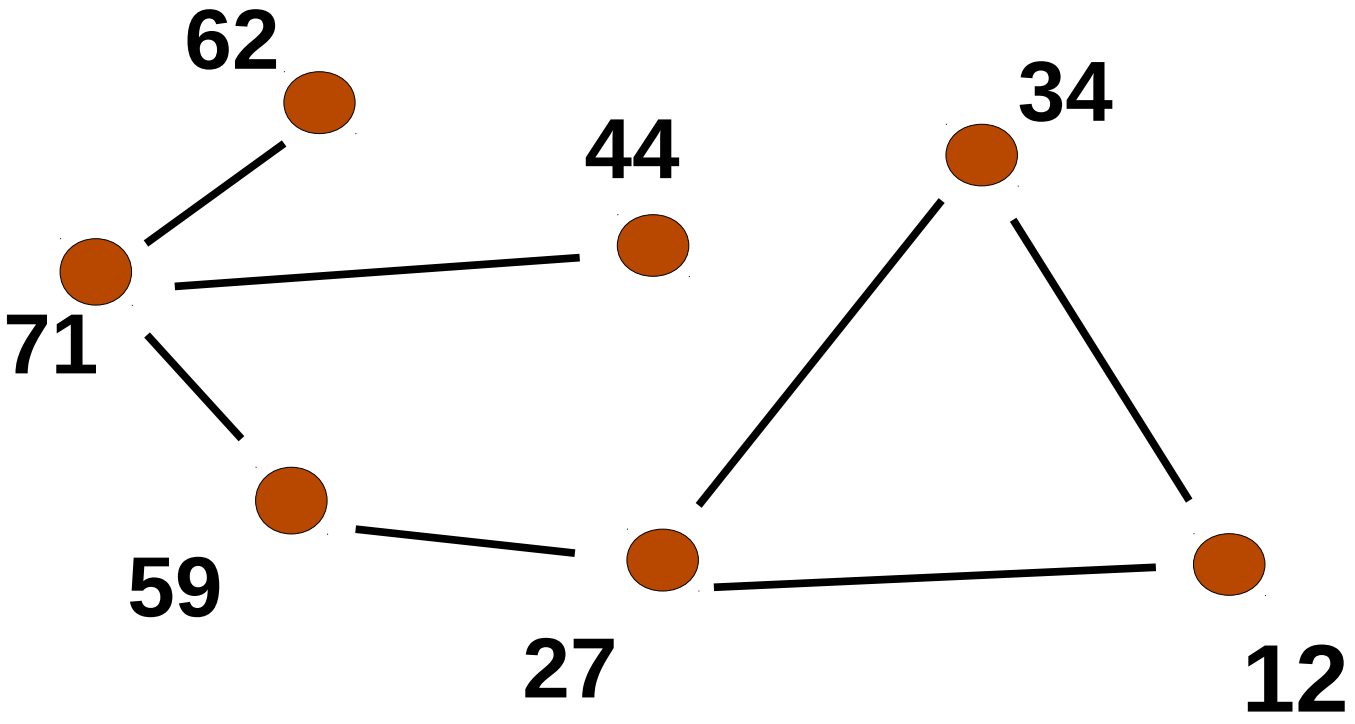
Distributed Object Location and Routing (DOLR) (Tapestry)

- The object is stored by the creator of the object.
- A hash key is computed and the object is published under the key.
- API:
  - publish(guid): object already created
  - unpublish(guid): remove connection
  - send(msg, guid, [n]): send a message to the object

Tapestry routing



unstructured overlay



Unstructured

- No network structure.
- You know some other nodes.
- No fixed location of objects
- Easy to join.
- Hard to search.
- No guarantees

Searching

- Flood the network
  - there should be a limit!
- Expanding ring
  - iterative flooding
- Random walk
  - several independent searchers
- Gossip
  - hopefully they will know

Summary

- Peer2Peer systems should scale with the number of clients by making the clients part of the service.
- Structured overlay
  - DHT, routing, how to join and leave, replication
- Unstructured overlay
  - group of peers, searching for content