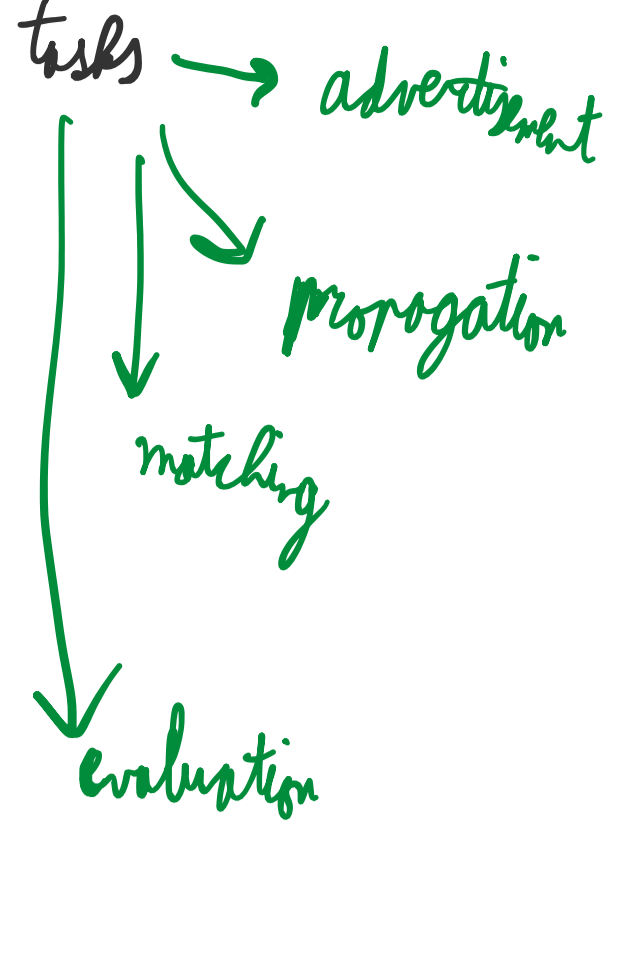


Service discovery: make two parties who do not know each other meet



discovery techniques: client knows access point, or contacts mediator to obtain actual service's AP
 responder, which listens to many access points, then hands over request to actual server
 (down AP)
 which may be stateful or demand

cluster
 first tier: logical switch ← the first tier may become a bottleneck
 second tier: application / compute servers
 third tier: distributed file / data system

goals: improve performance + reliability + availability

TCP handoff: switch hands off TCP connection to the server

application-aware distribution

front-end sends request, the selects best server

transport-layer switching

front-end passes TCP request to one of the servers, taking some performance metrics into account

routers can also find each other through multi-cast broadcasting

a name serves as a reference to an entity
 names are organized into a namespace

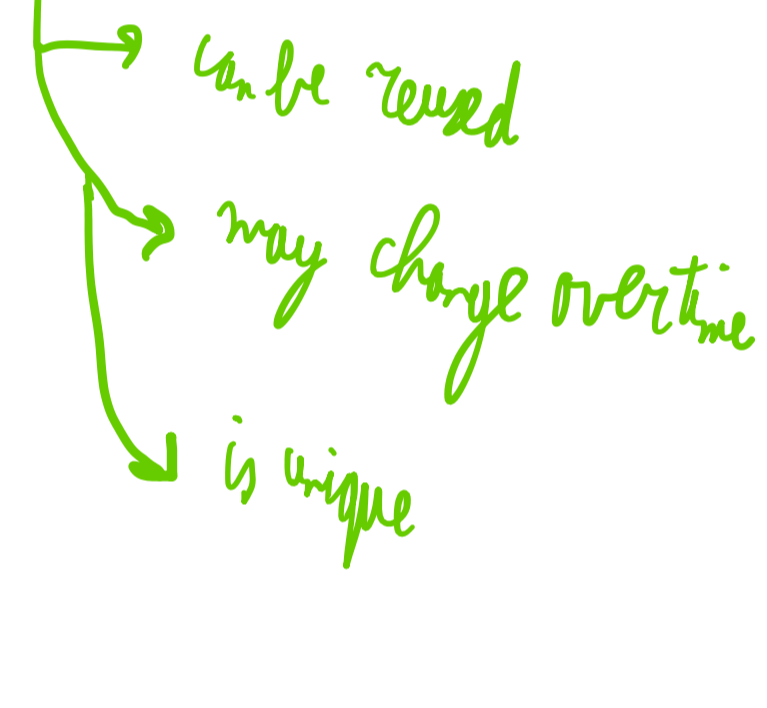
entity is a thing that needs to be identified
 Identifiers are the namespace for entities

true identifier properties

1. refers to one entity
2. almost one identifier per entity
3. never reused

access point: represents place where a entity can be reached

addresses form a namespace for access points



a namespace maintains a name-to-address binding
 a namespace is the set of valid names

hierarchical namespaces are unbounded in size and better manageable
 however, determining whether name is invalid may take longer

binding: relationship between reference + referenced object

resolution: given name, determine referenced object

- structured resolution
- flat resolution

lookup mechanism: knowing where and how to do resolution

naming types: flat naming

- structured-based
- attribute-based
- named-data networking

distributed flat resolution

distributed hashtable (DHT) of Chord

distributed storage
 flat name space: n-bit identifier
 set of nodes N
 set of entity E

each node p maintains a finger table FT_p with at most n entries
 $FT_p[E] = succ(p + 2^{i-1})$

i^{th} entry points to the first node succeeding p by at least 2^{i-1}

follow new wire stubs: mobile object replaced by stubs that forward invocations
 final stub responds with new stubs
 can store new / updated reference upon return (sticky fusion)

hierarchical naming systems are based on a labeled, directed acyclic graph
 edge-labeled

leaf node: access point

directory node: access point to directory structure