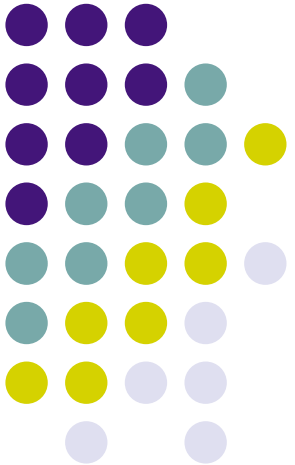
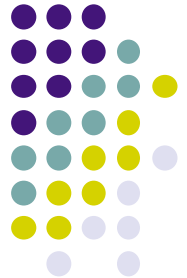
Contract Nets

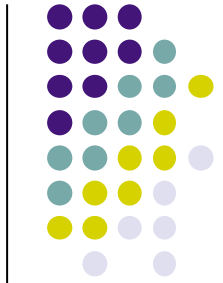


What is a contract net



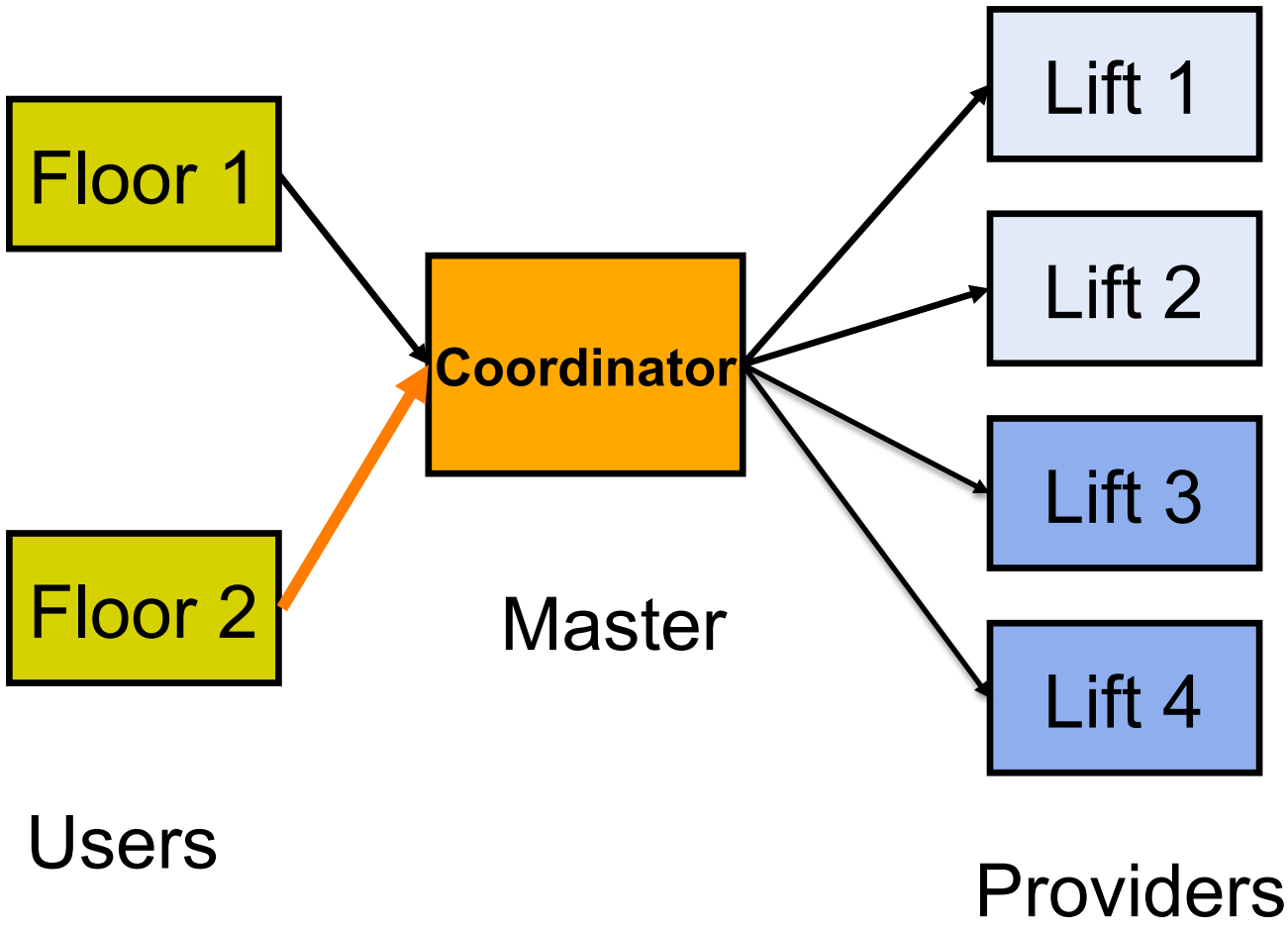
- Given a user and a set of providers
- The user sends a query to all providers
- Each provider responds with information
  - cloud a price, location, etc.
- The user select the provider that is most suitable
- The user informs all providers of its decision

Choosing an Agent

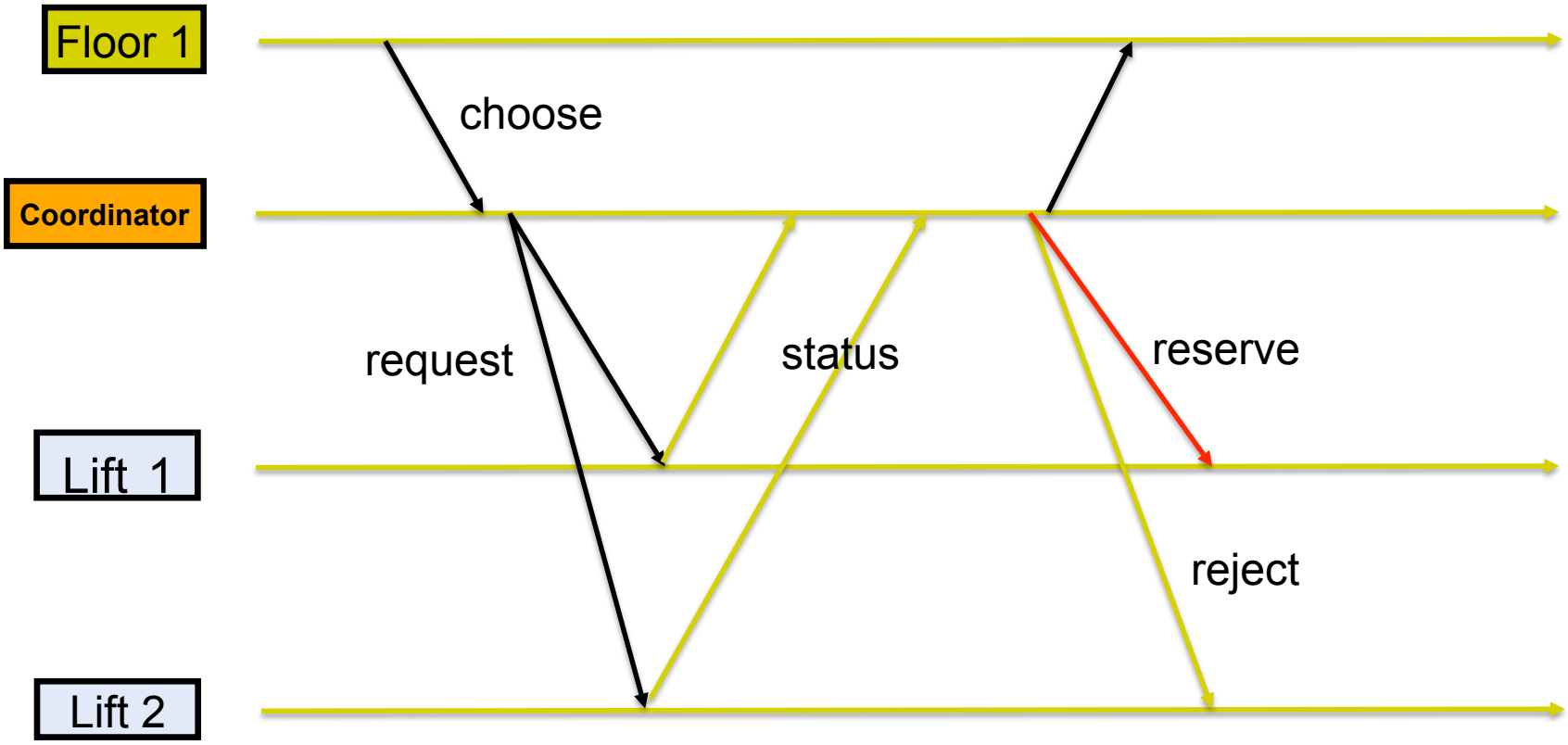


- Example: choosing the best lift
- More general: seeking agreement
- Lift control scenario
- General idea:
  - Floor agent
  - Coordinator
    - send message to all slaves requesting their status
  - Slaves: Lift agents
    - each answer by sending their information: relative position
    - coordinator choses the nearest
    - Each lift waits for a selection decision

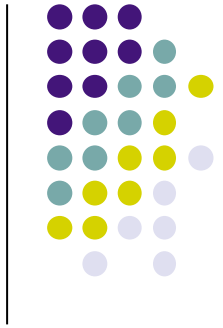
Lift Scenario



Lift Scenario: process-time diagram



Choosing an Agent



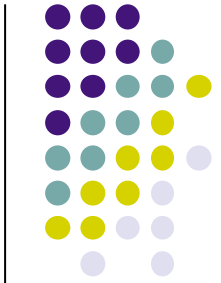
- Coordinator
  - Broadcasts a **request** enquiry to all slaves
  - Completes the whole protocol before responding to another client (Floor)

- Slave (Lifts):
  - receives **request** message
  - sends a **status** reply
  - waits until a decision is made (**reserve** or **reject**)

Master to Multiple Slaves

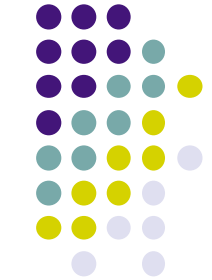
```
choose(pos:FN id:Agent) then
      Reply
      Rs = {Map S.agents
      fun {$ Slave} A in
         {Slave request(floor:FN answer:A)}
         A
      end}

% A = status(id:Agent, pos:AState, answer:X)
```

Slave behavior



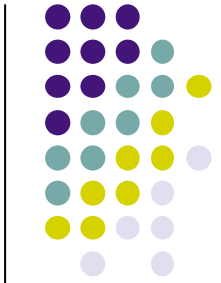
```
request(floor:F answer:R) then Ans in
    R = status(id:S.id pos:{Abs S.pos-F}
               answer:Ans)
   {Wait Ans}        …
```

Master to Multiple Slaves



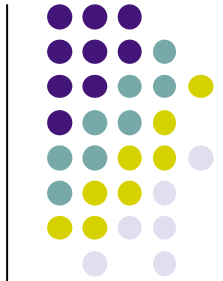
```
choose(pos:FN id:Agent)then Rs Reply in

  Rs = {Map S.agents  … }
  Reply =  {Minimize Rs}
  Agent = Reply.id
  for R in Rs do
    R.answer = if Agent == R.id then reserve
               else reject end
  end
```

Summary



- Protocols for coordinating agents
- We have seen
  - Broadcast protocol
  - Contract nets
- By using coordinator
  - We could make the protocol transactional
  - Finish the whole protocol before starting the next
  - Dataflow variables used as private channels within one protocol session

Summary



- Distributed Systems (and algorithms)
  - Where agents can also fail and message can be lost
- Protocols gets more complicated
  - Reliable broadcast
  - Agreement protocols (consensus)