Strategies for Distributed CBR

Santi Ontañón

IIIA-CSIC

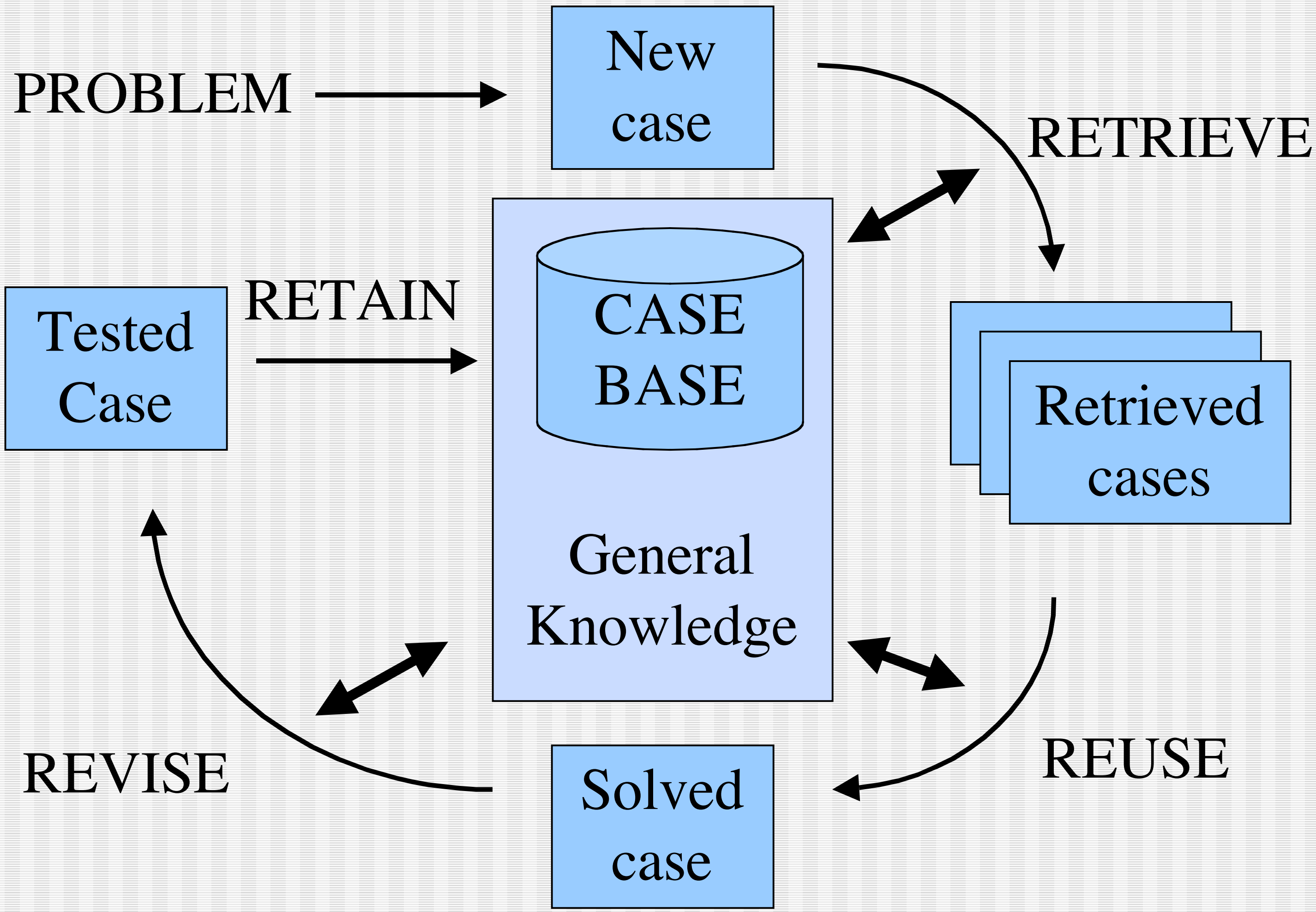
OUTLINE

- CBR
  - Introduction
  - Applications
- Distributed CBR
  - Autonomous CBR agents
  - Collaboration policies
  - Learning to collaborate
  - Case Bartering
- Conclusions

What is CBR?

- **<u>Case Based Reasoning</u>**: To solve a new problem by noticing its similarity to one or several **previously solved problems** and by adapting their solutions instead of working out a solution from scratch.

CBR: Cycle



PROBLEM
New case
RETRIEVE
CASE BASE
General Knowledge
RETAIN
Tested Case
Retrieved cases
REVISE
Solved case
REUSE

CBR: Retrieve

- Is the key problem in CBR
- Indexing:
  - Relevant features: predictive, discriminatory and explanatory.
  - Good case representation
  - Similarity metrics

CBR: Reuse

- Copy solution (for classification)
- Modify old solution:
  - Reinstantiate (variables)
  - Local search
- Transform
  - Domain knowledge (causal model)

CBR: Revise

- Apply solution to problem:
  - In real world
  - In simulated world
- Evaluate
- Repair
  - User guided
  - Internal (Domain knowledge)

CBR: Retain

- What to retain?
  - Relevant problems
  - Solution (failed/successful)
  - Solution method
- Store case
  - Update/indentify indexes

CBR systems:

- CHEF, cooking planing (Hammon 86)
- HYPO, legal reasoning (Ashley/Rissland 87)
- PROTOS, medical diagnosis (Bereiss/Porter 88)
- CASEY, medial diagnosis (Koton 89)
- SAXEX, expresive music (Arcos 96)

Distributed CBR

- Autonomous CBR agents
- Collaboration policies
- Learning to collaborate
- Case Bartering

Autonomous CBR agents

- Multiagent CBR system
  - Each agent has an individual Case-Base
- Agent has autonomy in
  - Problem acquisition
  - Problem solving
  - Learning
- Collaboration policies
  - Explicit strategies to improve individual performance

Multiagent CBR system

- MAC (Multiagent CBR) System

$$\{(A_i, C_i)\}_{i=1\ldots n}$$

- An agent is a pair (A_i, C_i)
- A Case base $\{(P_j^i, S_k)\}_{j=1\ldots N_i}$
- Classification task in $\{S_1, \ldots, S_K\}$

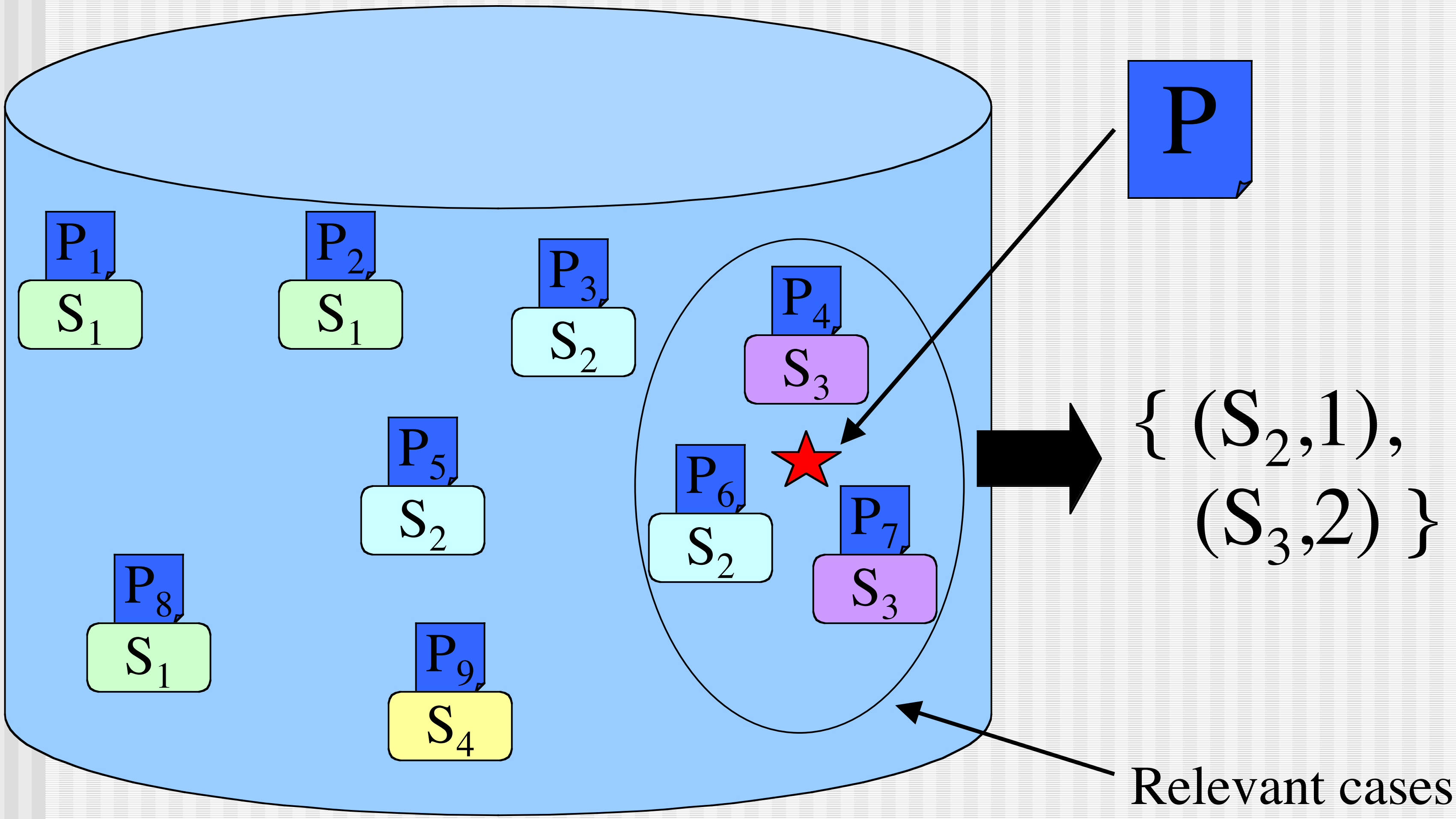
Collaboration policies

- Committee
- Peer Counsel
- Bounded Counsel

Collaboration

- An agent A_1 with problem P
  - Sends P to some agents $A=\{ A_2,\ldots,A_n \}$
  - Each A_j answers with a **Solution Endorsing Record (SER)**
    - $< \{ (S_1\, E^j_1) \ldots (S_m\, E^j_m) \} , P , A_j >$
  - Endorsing pairs: (S_k, E^j_k)

Solution Endorsing Record (SER)



Voting(1)

- A^t is the set of agents submitting SER at time t (including the initiating agent).
- Each agent has 1 vote
  - The vote can be fractionally assigned to various solution classes:

$$Vote(S_k, A_i) = \frac{E_k^i}{c + \sum_{r=1...K} E_k^i}$$

Voting(2)

- The agent that receives de SERs, aggregates the votes:

$$Ballot^t(S_k, \boldsymbol{A}^t) = \sum_{A_i \in \boldsymbol{A}^t} Vote(S_k, A_i)$$

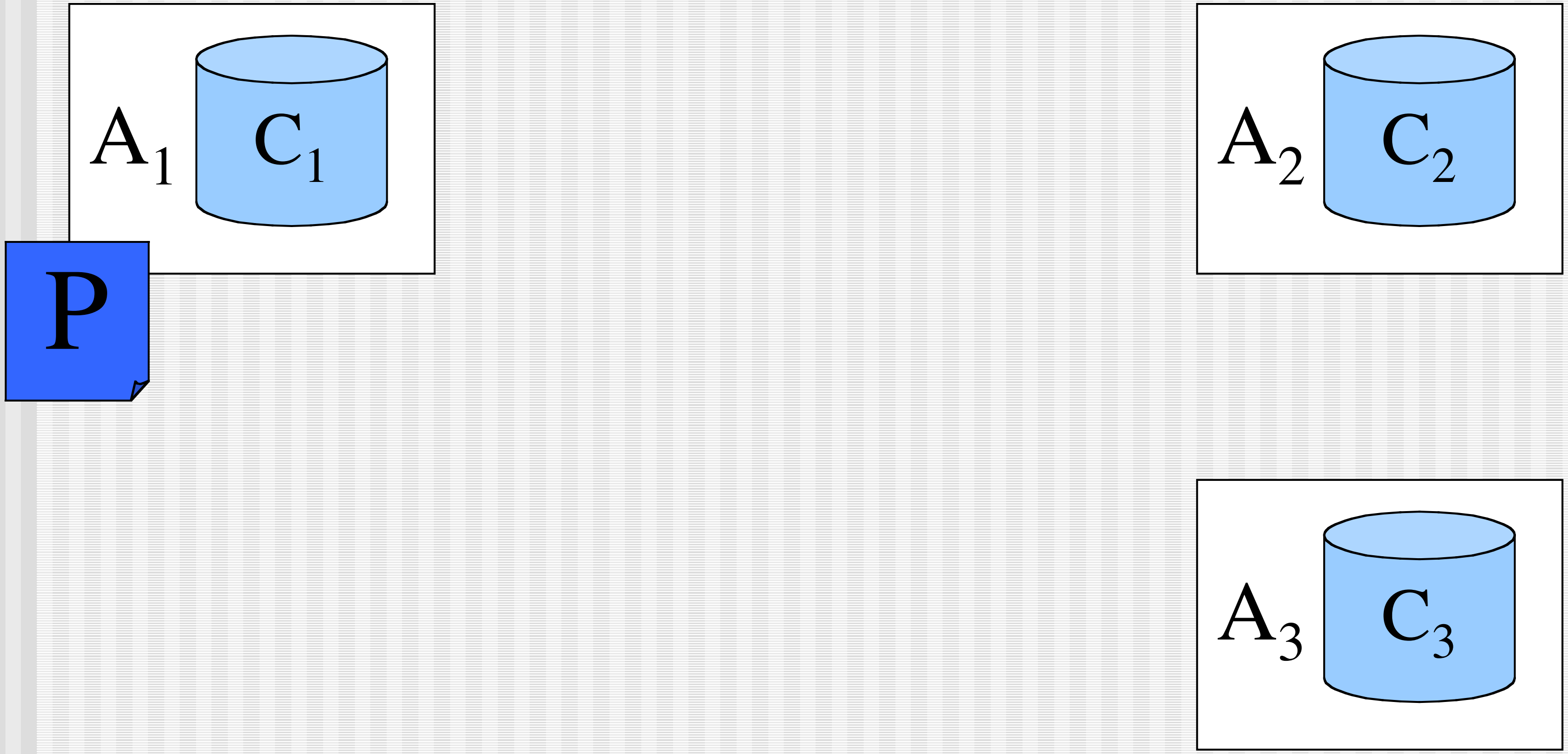
- The proposed solution is that with maximum Ballot:

$$Sol^t(P, \boldsymbol{A}^t) = \arg\max_{k=1\ldots K}\left(Ballot^t(S_k, \boldsymbol{A}^t)\right)$$

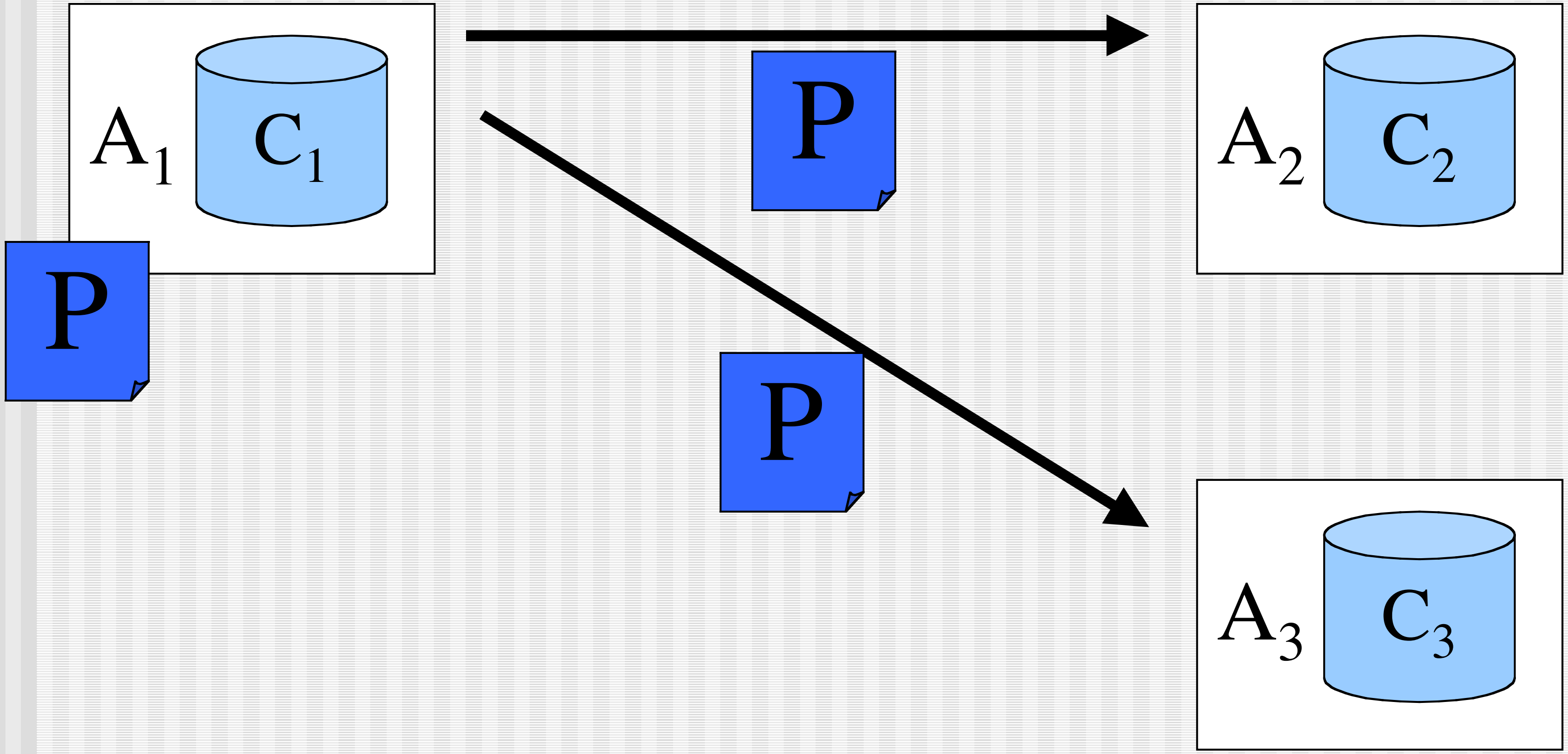
Committee policy (1)

- An agent A_1 with problem P
  - Sends P to all the other agents $A=\{ A_2,\ldots,A_n \}$
  - Each A_j answers with a **Solution Endorsing Record (SER)**
    - $< \{ (S_1\, E^j{}_1)\ldots(S_m\, E^j{}_m) \} , P , A_j >$
  - The majority vote selects the solution class

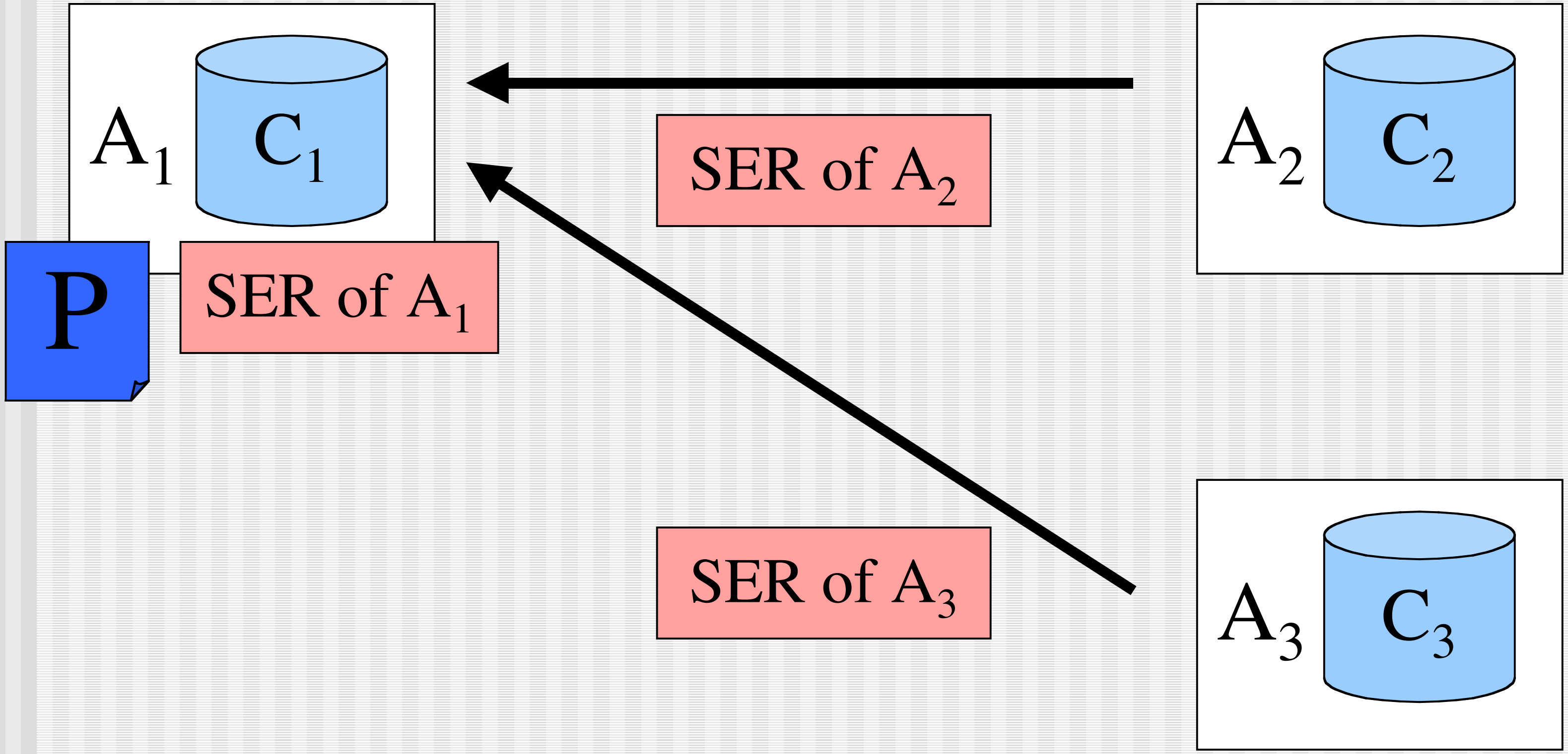
Agent interaction: Committee



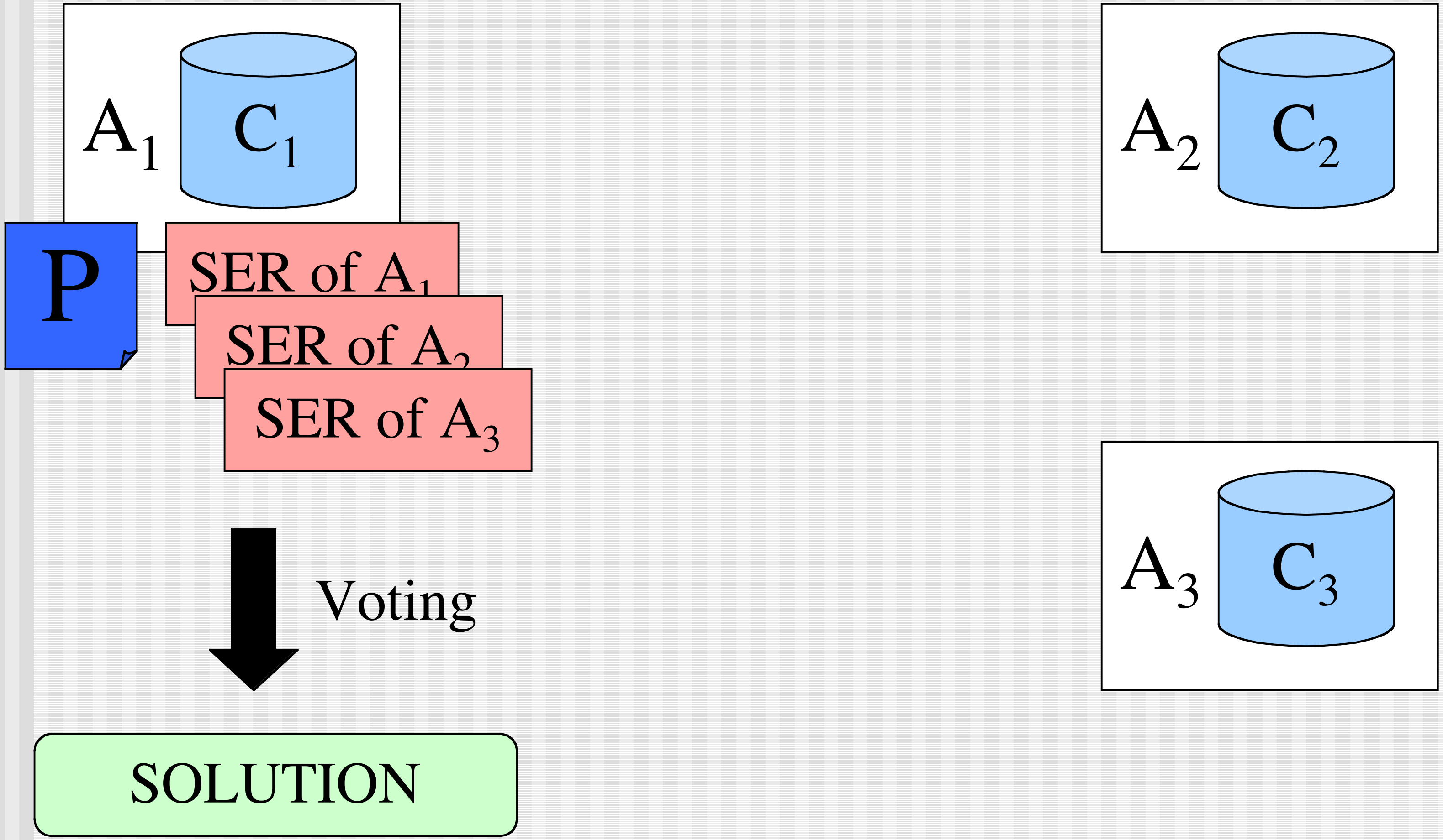
Agent interaction: Committee



Agent interaction: Committee



Agent interaction: Committee



Committee policy (2)

- Robust
  - Good accuracy even with big number of agents with small case-bases
- Cost
  - Linear with number of agents
  - Always asks all agents

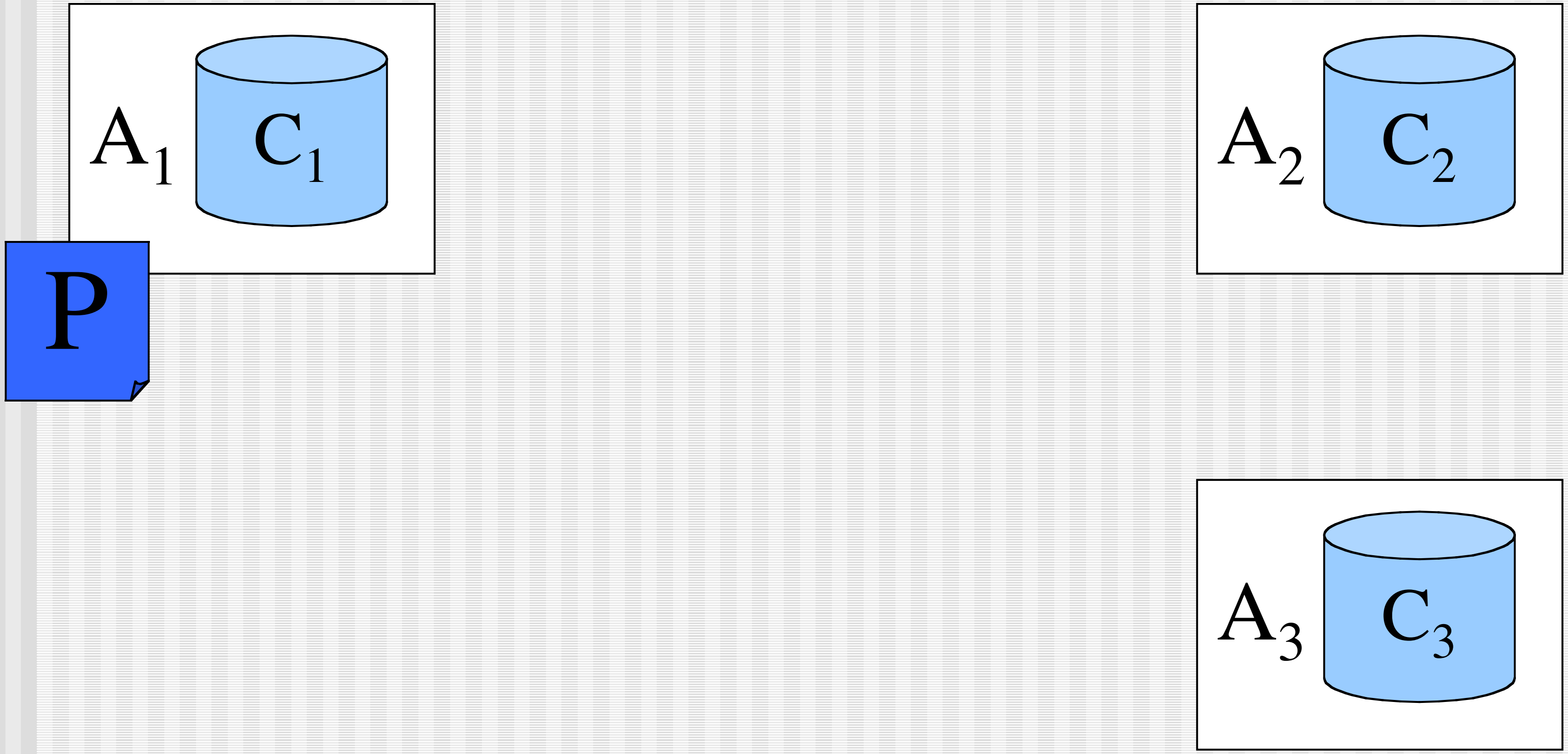
Peer Counsel

- The agent try to solve the problem by itself
  - Assess competence of current solution:
    - Votes for the max class >> rest of votes:

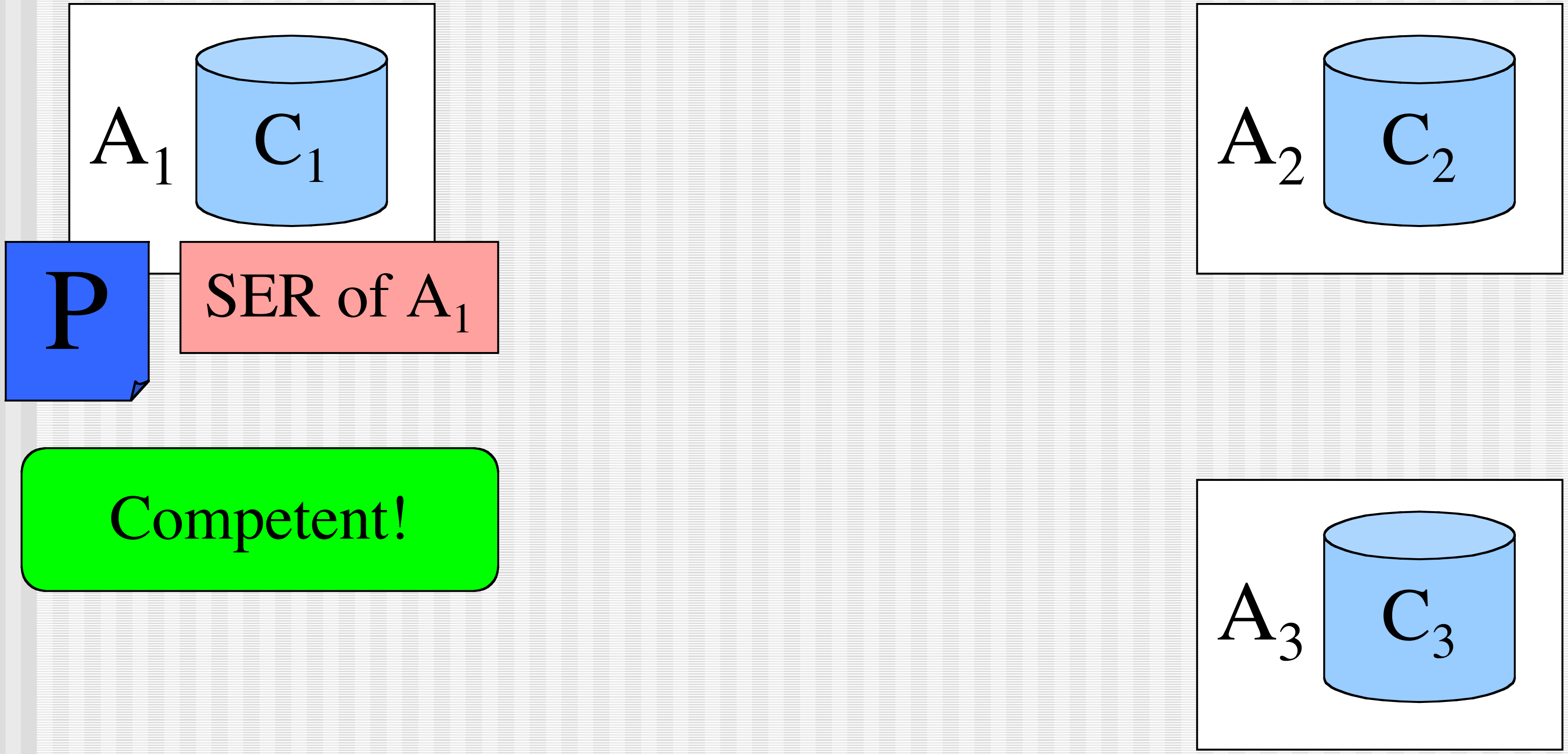
$$\frac{V_{\max}^{t}}{V_{rest}^{t}} > \eta$$

- If not competent, ask **all** other agents
  - Like committee

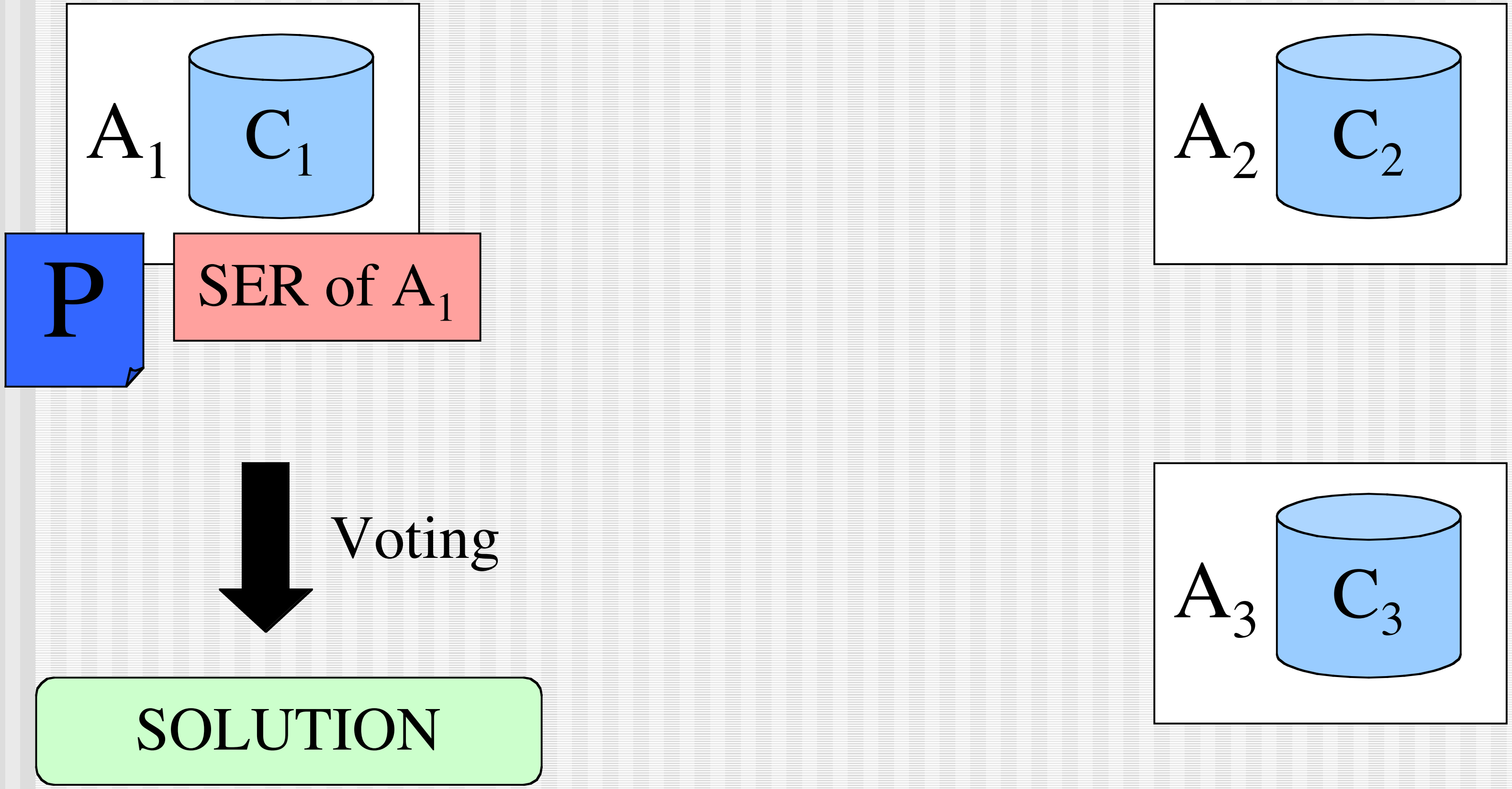
Agent interaction: Peer Counsel



Agent interaction: Peer Counsel



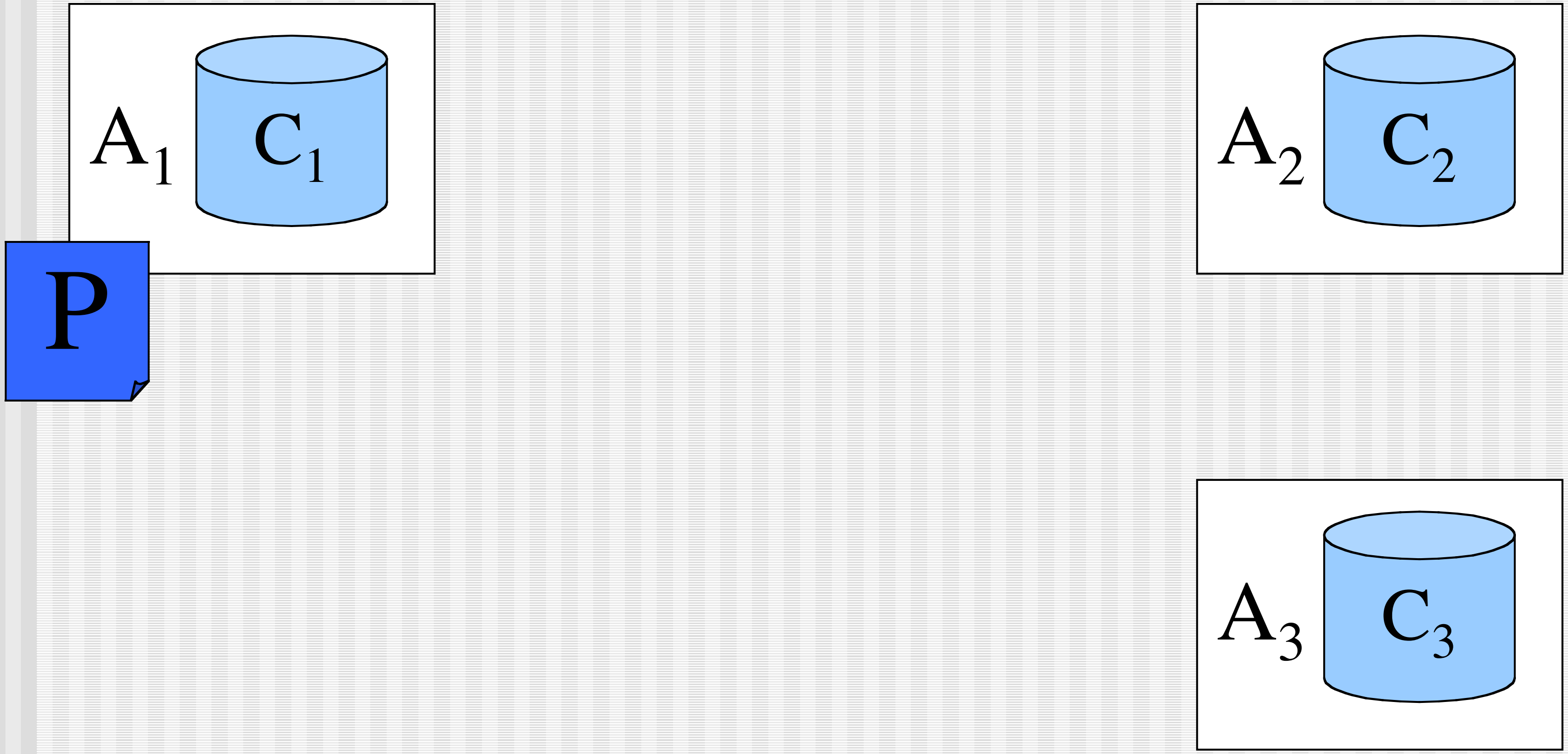
Agent interaction: Peer Counsel



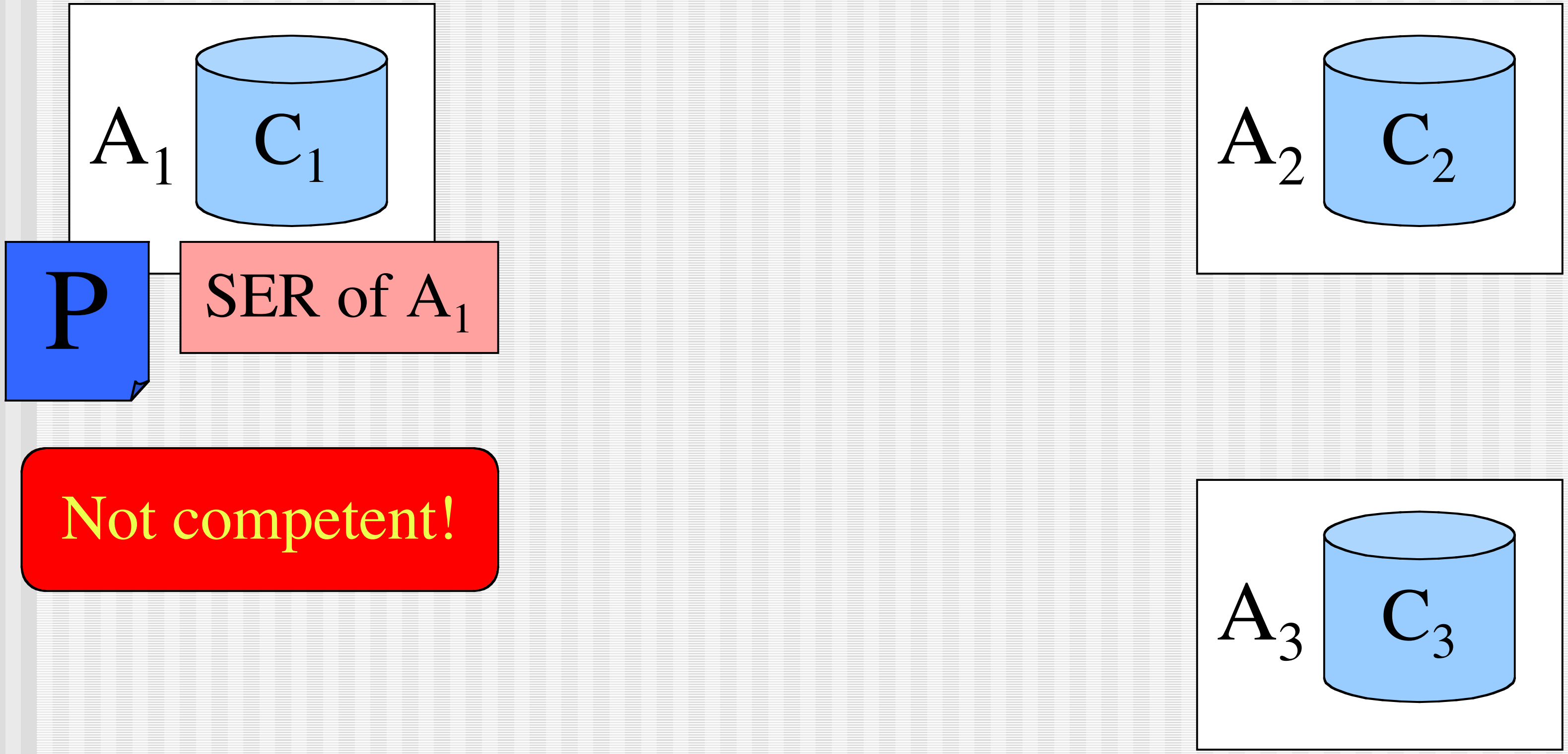
Bounded Counsel

- Use self-competence model
- If not competent, ask **one** agent
  - Agent returns a SER
  - Assess competence of current solution (**Termination Check**):
    - Votes for the max class >> rest of votes: $\frac{V_{\max}^{t}}{V_{rest}^{t}} > \eta$
  - If not competent, ask one agent more

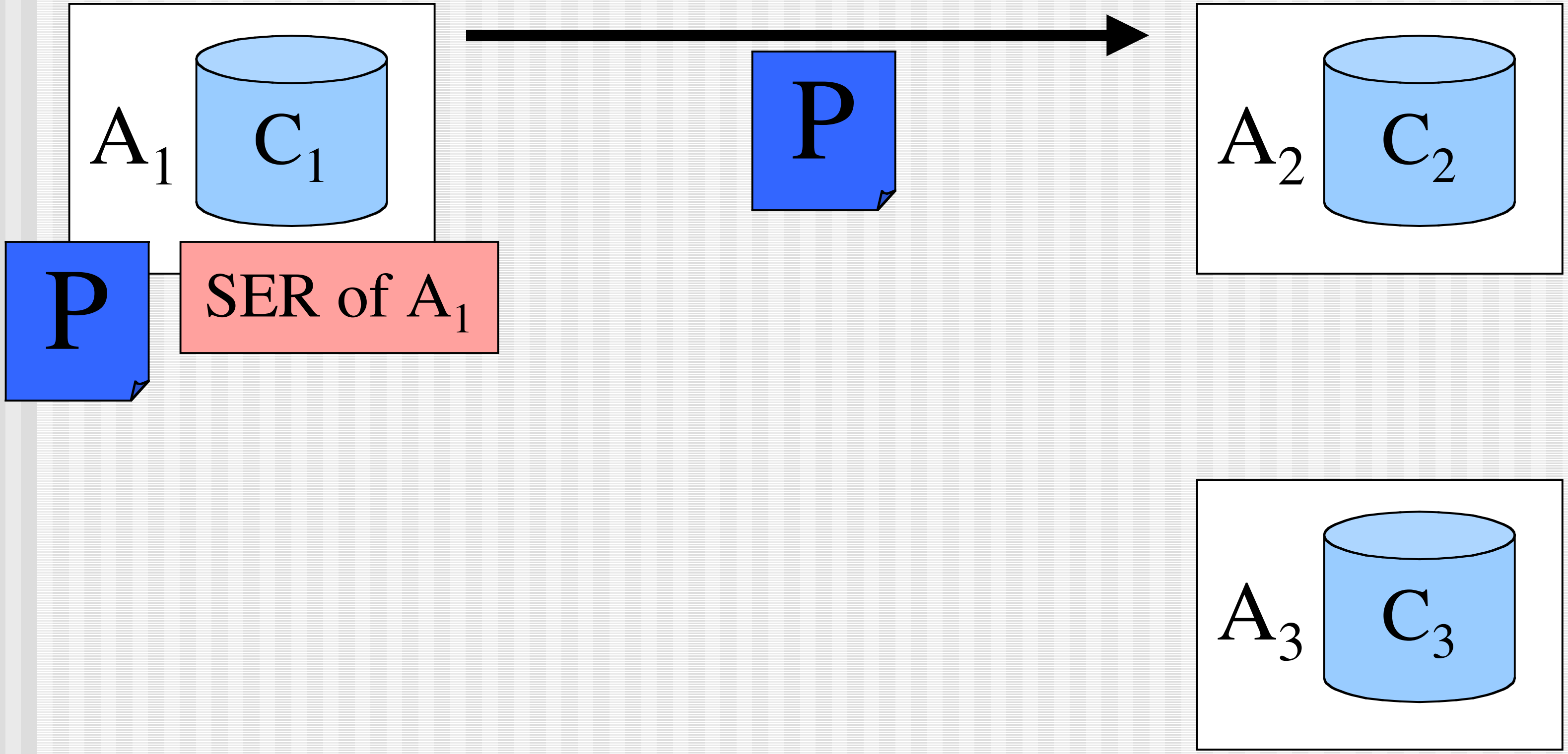
Agent interaction: Bounded Counsel



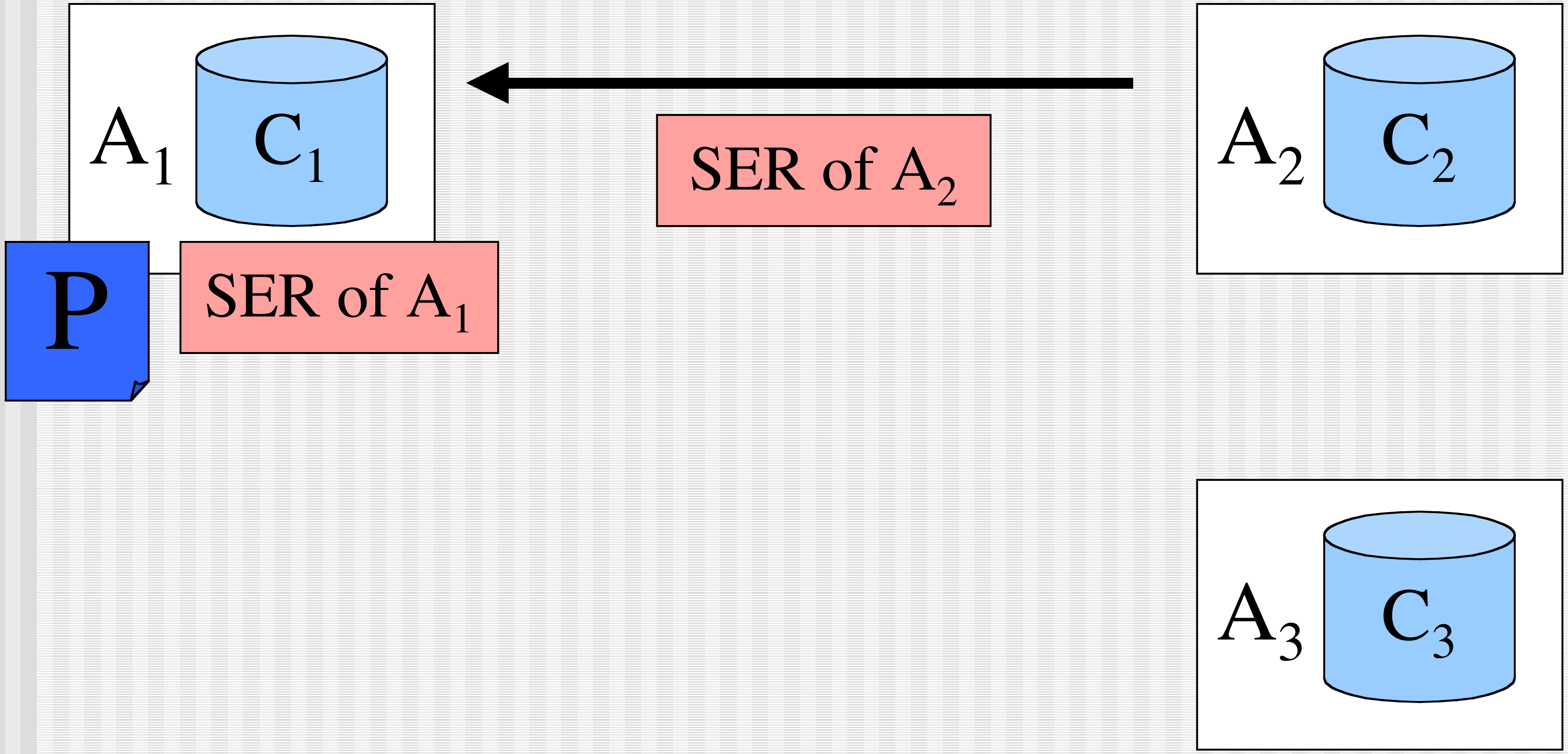
Agent interaction: Bounded Counsel



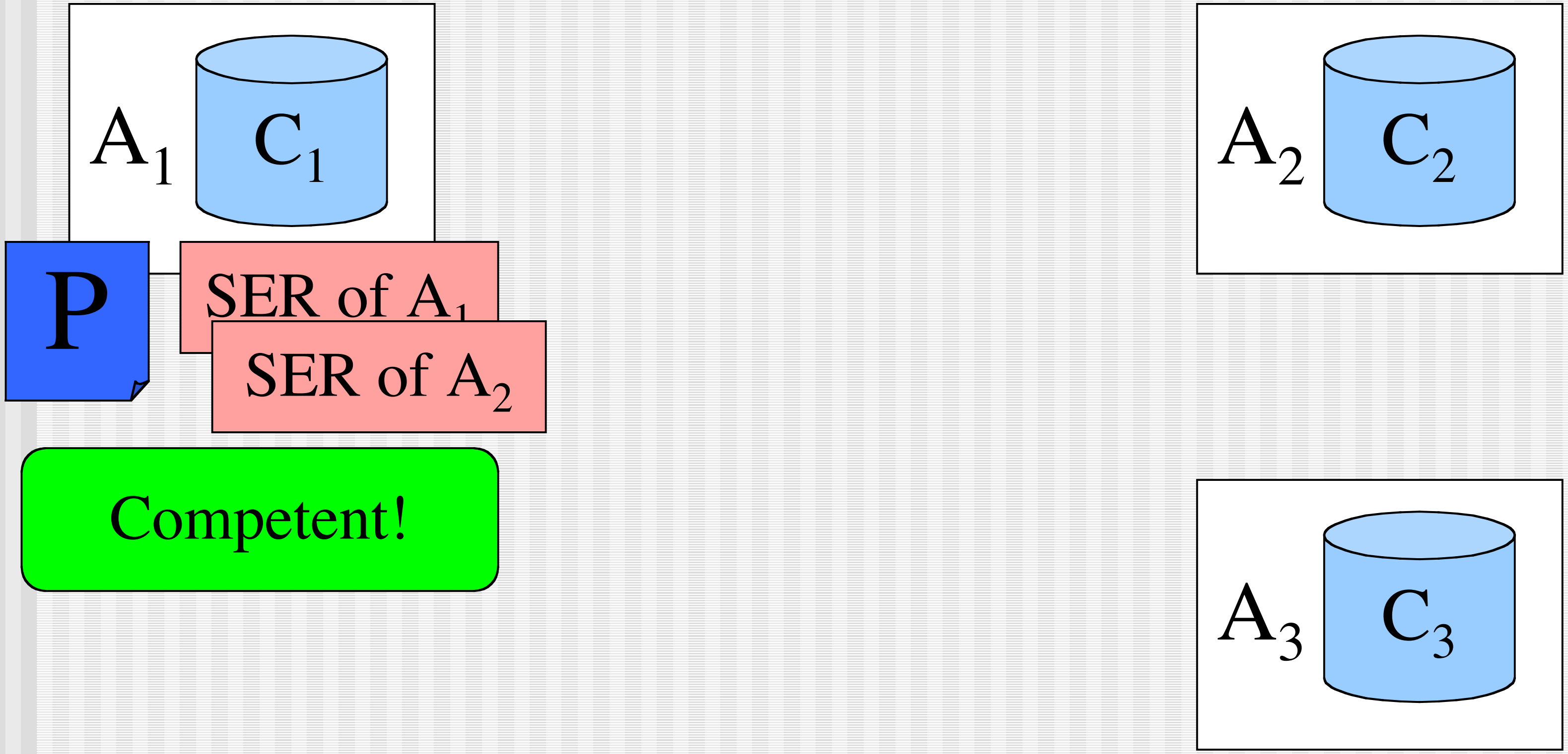
Agent interaction: Bounded Counsel



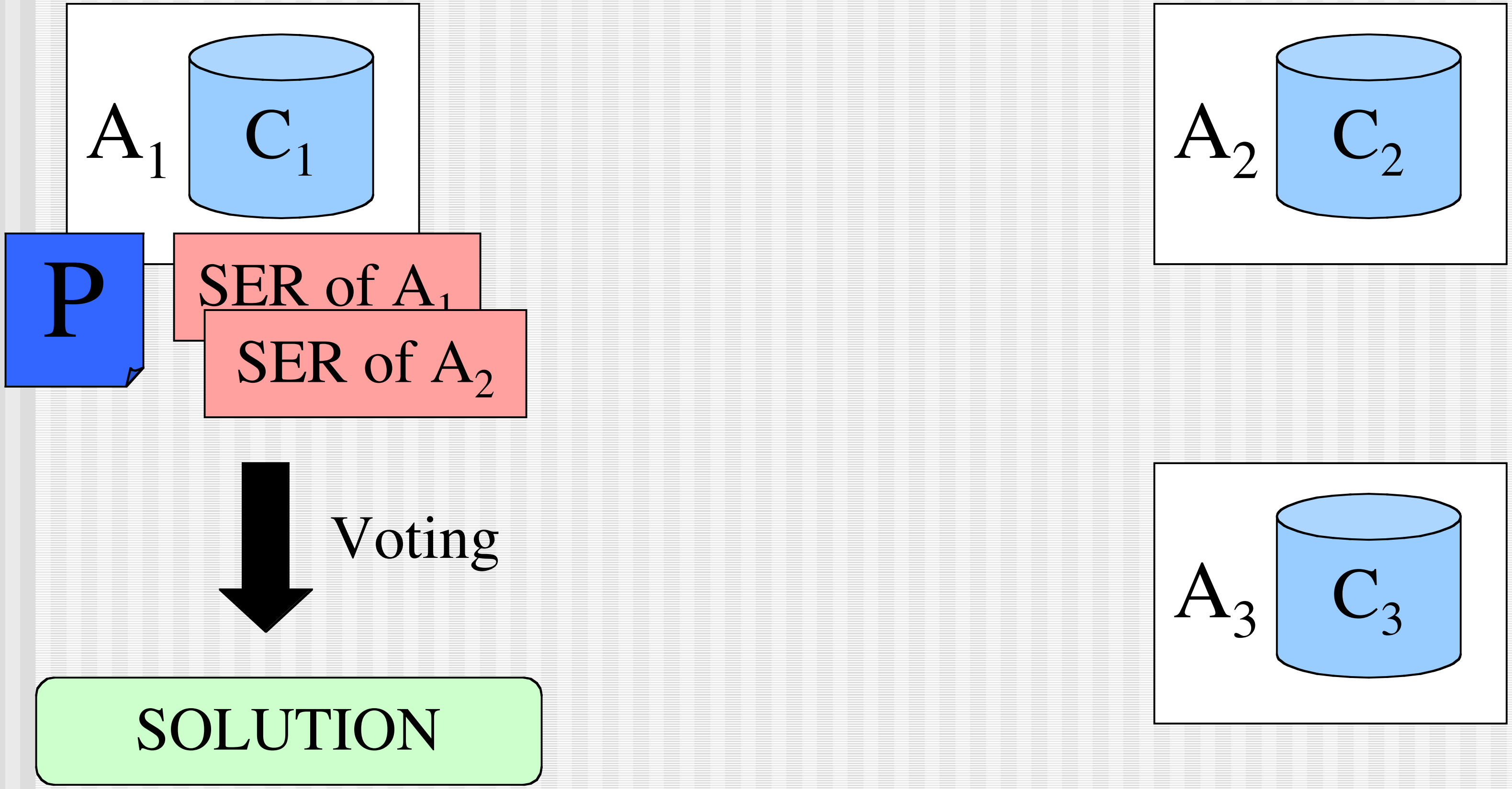
Agent interaction: Bounded Counsel



Agent interaction: Bounded Counsel



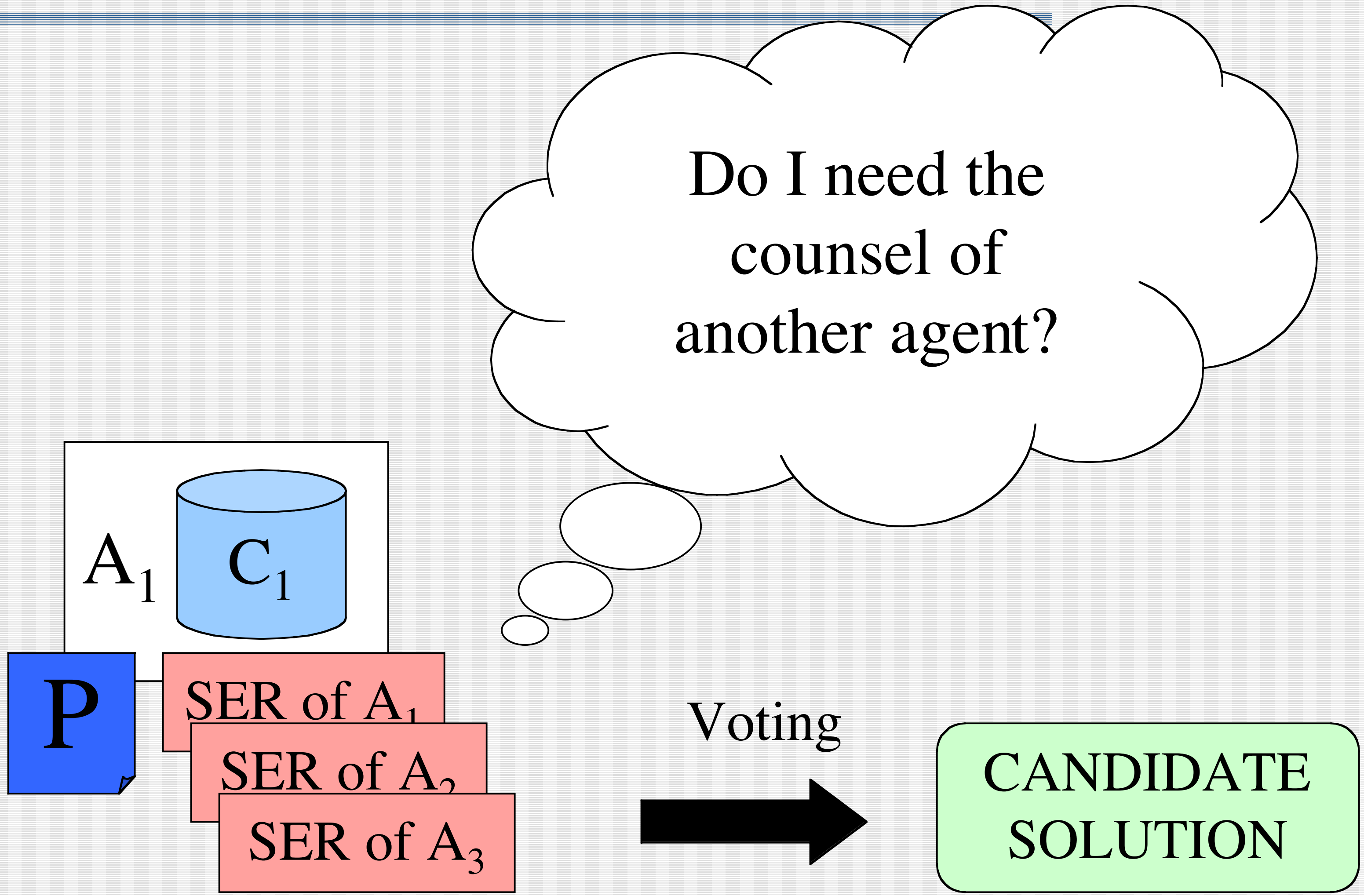
Agent interaction: Bounded Counsel



Learning to collaborate

- Bounded Counsel
  - fixed predicate to assess competence
- Proactive Learning
  - Improve collaboration
  - Actions performed in order to learn
  - learning when to ask counsel

Learning to collaborate(2)



Case Bartering

- Why?
  - Biased individual case-bases
- How?
  - Estimating underlying distribution
  - Interchanging cases between agents

Case Bartering(2)

- Estimation of underlying distribution:
  - Aggregating statistics of all the individual case-bases.

$$D = \{D^1, \ldots, D^K\}$$

$$\boxed{D^j = \frac{\sum_{i=1}^{n} d_i^j}{\sum_{i=1}^{n} \sum_{l=1}^{K} d_i^l}}$$

Case Bartering(3)

- Case-Base bias:

$$Bias(A_i) = \sum_{l=1}^{K}\left(D^l - \frac{d_i^l}{\sum_{j=1}^{K} d_i^j}\right)^2$$

- The agents should minimize its bias

Case Bartering(4)

- Bartering Protocol:
  - Estimate underlying distribution
  - Send offers
  - Confirm offers
  - Interchange cases
  - If changes go to 2, else **END**.

Conclusions

- Multi-agent systems offer new paradigms to organize AI.
- Agent interaction is still a stimulating challenge.
- CBR fits perfectly with these kind of applications.