Evaluating Centralized, Hierarchical, and Networked Architectures for Rule Systems

Benjamin Craig

University of New Brunswick
Faculty of Computer Science
Fredericton, NB, Canada

Senior Technical Report Presentation
November 20, 2008

Outline

- Defining the Terminology
  - Rules, Distributed Systems, Topologies, OO jDREW, Rule Responder
- Topologies for Distributed Architectures
  - Star Topology Advantages and Disadvantages
  - P2P Topology Advantages and Disadvantages
- Knowledge Maintenance for Rule Systems
  - Knowledge Organization
  - Knowledge Maintenance
- Conclusion

What is a rule?

- Fact (POSL format):

```
spending(Peter Miller, min 5000 euro, last year).
```

- Rule (POSL format):

```
premium(?Customer) :-
   spending(?Customer, min 5000 euro, last year).
```

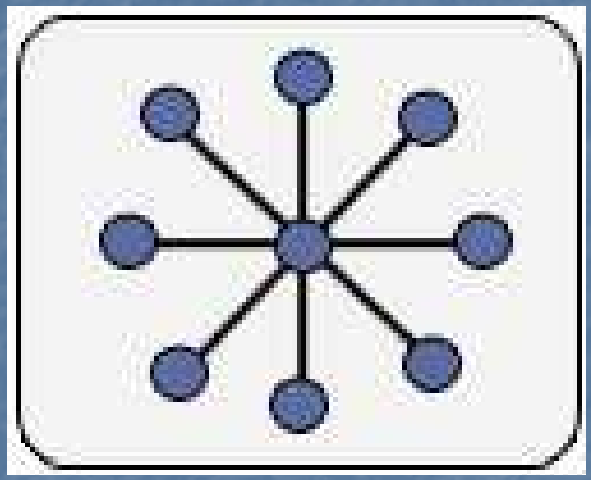
- A deductive rule engine can deduce that Peter Miller is a premium customer from his spending

Distributed Systems

- A distributed system is a set of computer processes that appear to the user as a single system
- The distributed system must coordinate all of these processes
- Distributed systems are implemented using middleware that creates a communication topology

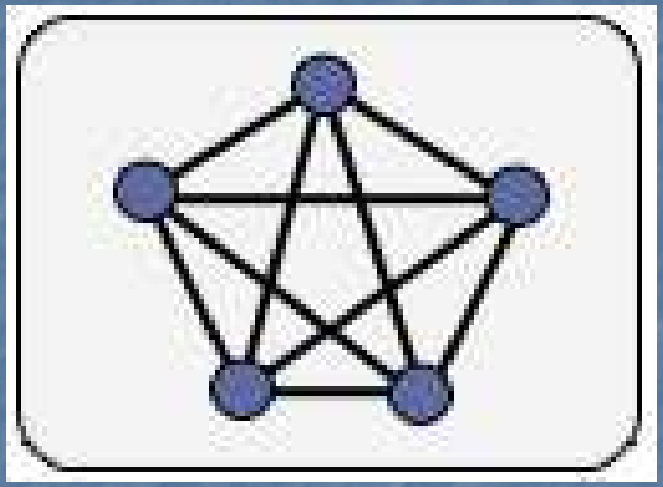
Hierarchical - Star Topology

- Single level hierarchy
- Star: connects all spokes with a centralized hub
- All information must be sent through the hub to the spokes

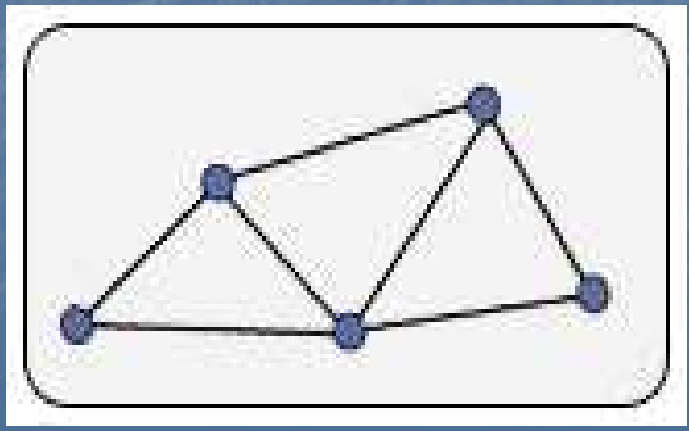


Networked - P2P Topology

- Fully connected network
  - Full mesh topology: connects all nodes together with a direct connection



- Partially connected network
  - Partial mesh topology: Only a subset of nodes are connected together

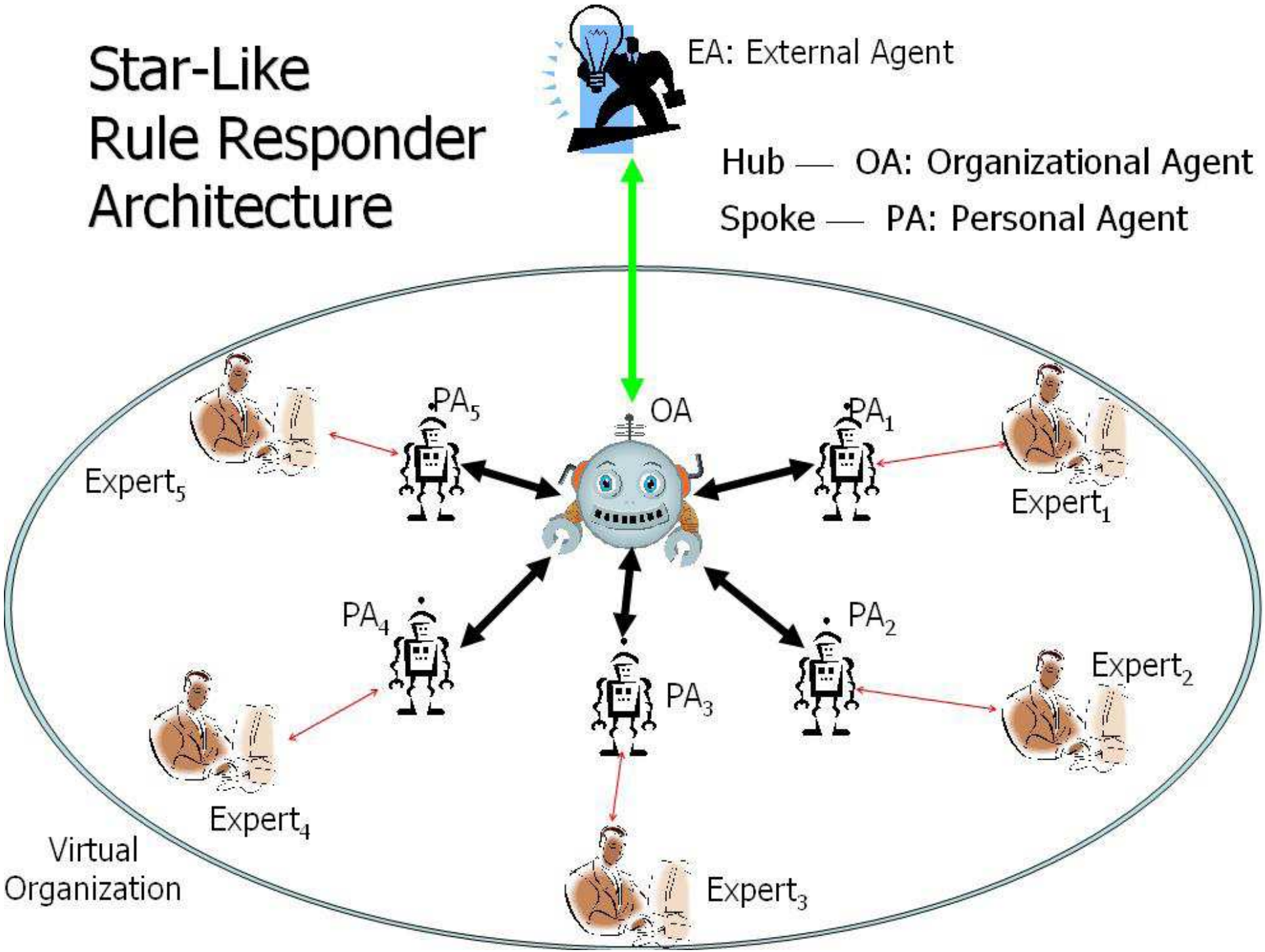


OO jDREW

- Centralized Rule System
- Object Oriented Java Deductive Reasoning Engine for the Web – extensions of jDREW
- Also implements agents of Rule Responder

Rule Responder

- Distributed rule system
- Is currently implemented as a hierarchical rule system
- Rule Responder is a prototypical multi-agent system for virtual communities
- Supports rule-based collaboration between the distributed members of community
- Members are assisted by semi-automated **rule-based** agents, which use rules to describe the decision and behavioral logic


Star-Like
Rule Responder
Architecture
EA: External Agent
Hub — OA: Organizational Agent
Spoke — PA: Personal Agent
OA
PA5
PA1
PA4
PA2
PA3
Expert5
Expert1
Expert2
Expert3
Expert4
Virtual
Organization

Topology Performance

- When building a distributed system a communication topology is required
- Distributed topologies all have communication overhead that centralized systems do not have
- A key design goal for distributed systems is to minimize this communication overhead

Star Advantages

- Isolation of spokes from other spokes
  - If one spoke fails then it does not affect others
- Adding and removing nodes in the hub is trivial (just add/remove a spoke)
- Hub provides single point of inspection of all traffic through the topology
  - Improved Security
- Trouble shooting is easy
- Easy to understand and implement

Star Disadvantages

- Scalability, reliability and performance of the star topology rely on the hub
- If the hub fails then the entire system fails
- The hub can become overloaded and the system will experience slowdown

  To prevent the bottleneck of the star topology a P2P topology can be used

P2P Advantages

- Removes bottleneck performance issues of the star topology
- Whenever a node is added the internal bandwidth capacity is increased
- When a node fails the system will be able to recover
  - A peer can act in place of another peer

P2P Disadvantages

- Adding and removing nodes in the P2P topology is non-trivial (need to add many connections)
- P2P networks are more complicated than star topologies
  - Difficult trouble shooting
  - Difficult to implement

Knowledge Maintenance for Rule Systems

- A centralized rule system has all of the knowledge stored in a single location
  - Either a file or a database
- Both distributed topologies are used to implement rule systems
- A distributed rule system partitions the knowledge bases across the system
  - Each knowledge base acts as a module
  - Many files and databases

Knowledge Organization - I

- When deciding how to group modules one of two ways can be used
  - Predicate Centric
    - All clauses of a predicates are stored in one module
  - Person Centric
    - All clauses about one person or thing is stored in one module
    - Rule Responder uses person centric organization
- *example on next slide*

Knowledge Organization - II

- Predicate Centric:
  - phoneOf(ben, 1-506-270-3403)
  - phoneOf(jim, 1-506-275-9712)
  - emailOf(ben, ben.craig@unb.ca)
  - emailOf(jim, jim.lorde@unb.ca)
- Person Centric:
  - phoneOf(ben, 1-506-270-3403)
  - emailOf(ben, ben.craig@unb.ca)
  - phoneOf(jim, 1-506-275-9712)
  - emailOf(jim, jim.lorde@unb.ca)

Module Boundaries

- When querying modules, sometimes information from multiple modules is required
- Example Queries
  - "What are the phone numbers of everyone in the organization?"
    - This query must backtrack across multiple modules when using person centric storage
  - "What properties does Ben have?"
    - This query does not require backtracking across multiple modules when using person centric storage

Centralized Maintenance

- All knowledge is stored in a single location
  - Updating knowledge is simple
  - Can better avoid/repair knowledge inconsistencies
- All knowledge is stored in a single format
  - No translation steps when using a rule engine to execute the rules and facts

Distributed Maintenance

- Knowledge is stored in a many locations
  - Each agent can scalable update their own knowledge
  - Knowledge bases could become incomplete or inconsistent
  - Integrity rules can be used to test if the knowledge is complete and consistent
- Knowledge is stored in many formats
  - Translation steps are required when sending a query from one rule engine to another
  - An interchange language is required

Deployed Benchmarking Use Case

- RuleML-20xy Symposia
  - An organizational agent acts as the single point of entry to **assist** with symposium planning:
    - Currently, query answering about the symposium
    - Ultimately, preparing and running the symposium
  - Personal agents have supported symposium chairs since 2007 (deployed as Q&A in 2008)
    - General Chair, Program Chair, Panel Chair, Publicity Chair, etc.

Queries Used

- 1) Sponsoring the symposium: 5 rules
- 2) Check panel participants: 1 rule, 3 joins
- 3) View symposium sponsors: 1 rule, 3 joins
- 4) View organization partners: 1 rule, 2 joins
- 5) Check panel time: 1 rule, 5 joins

OO jDREW (centralized) Benchmarking

Query:	Computation Time (ms):
1)	141
2)	31
3)	22
4)	18
5)	16

- Results show that our centralized use case does not take much computation time
- Queries do not require heavy computation

Rule Responder (Hierarchical) Benchmarking

- Same 5 queries used as in the OO jDREW benchmarking

Query:	Computation Time (ms):
1)	3430
2)	4861
3)	4057
4)	9048
5)	2780

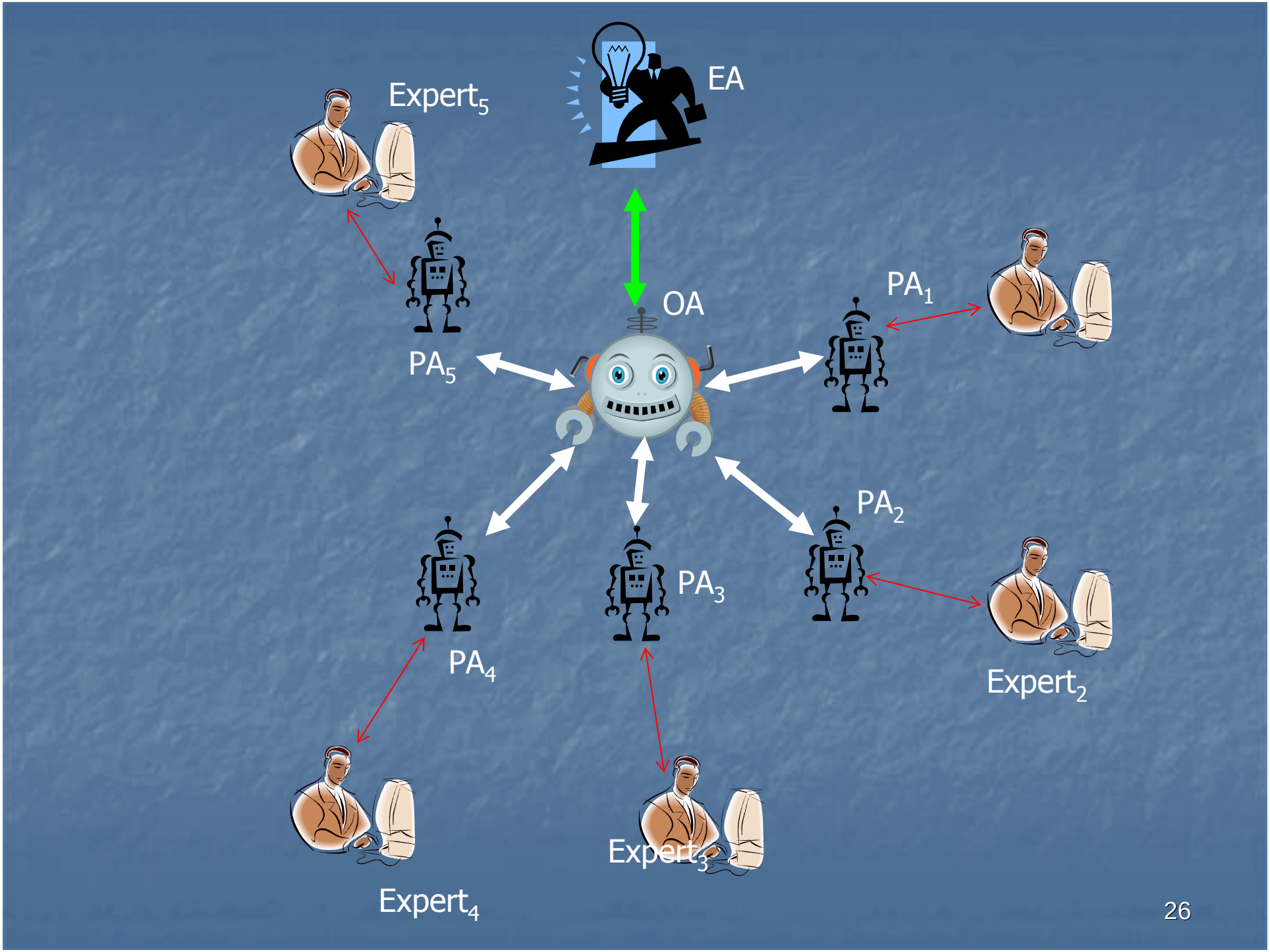
- Increase in computation time due to sequential delivery of answers to the queries
- Communication overhead of distributed system not compensated by workload distribution

Network Performance Considerations

- Speed-ups can be obtained using a P2P topology
- Instead of all communication going 'vertically' through the hub, direct 'horizontal' communication between spokes could be often used
  - Will reduce the amount of communication steps in the distributed system
- The bottleneck issue of a hierarchical system does not exist in a networked system

Conclusion

- A rule system can be either distributed or centralized
- When using a distributed system the communication topology must be decided
- The topology should reflect the modularization decisions about the distributed rule system
- The advantages and disadvantages of distributed knowledge maintenance must be weighted when building a rule system
- Our initial benchmarks, not requiring heavy computation, show increase in computation time for a distributed hierarchal system
- Only networked systems will *scale* to the Web


EA
Expert5
OA
PA5
PA1
PA2
PA3
PA4
Expert2
Expert3
Expert4