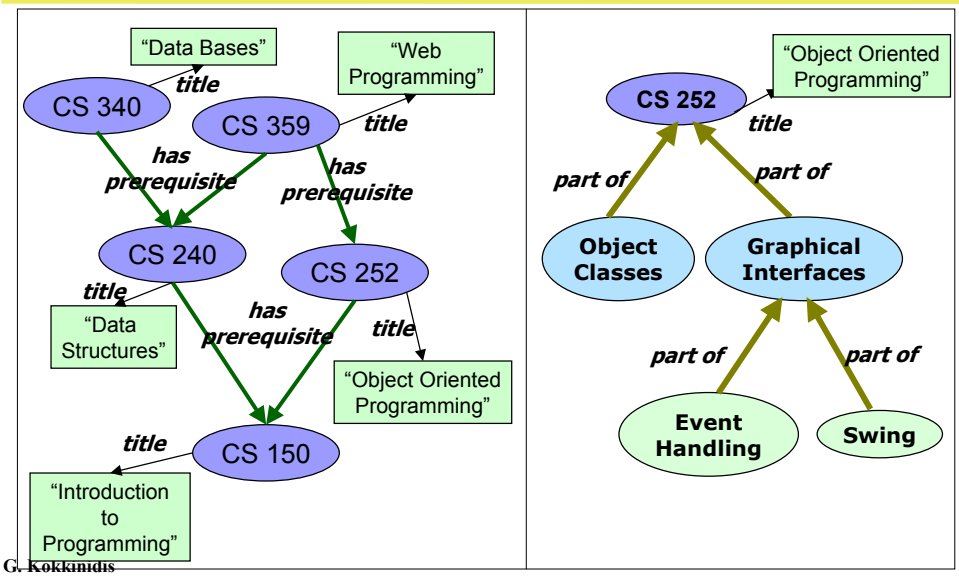
Semantic Query Routing and Processing in P2P Database Systems: The ICS-FORTH SQPeer Middleware

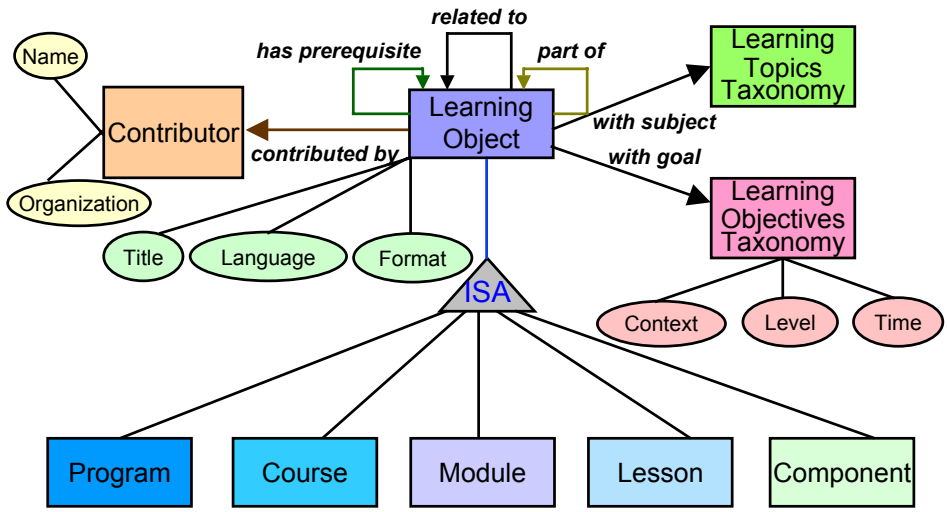
HDMS 2004

George Kokkinidis and Vassilis Christophides
ICS-FORTH

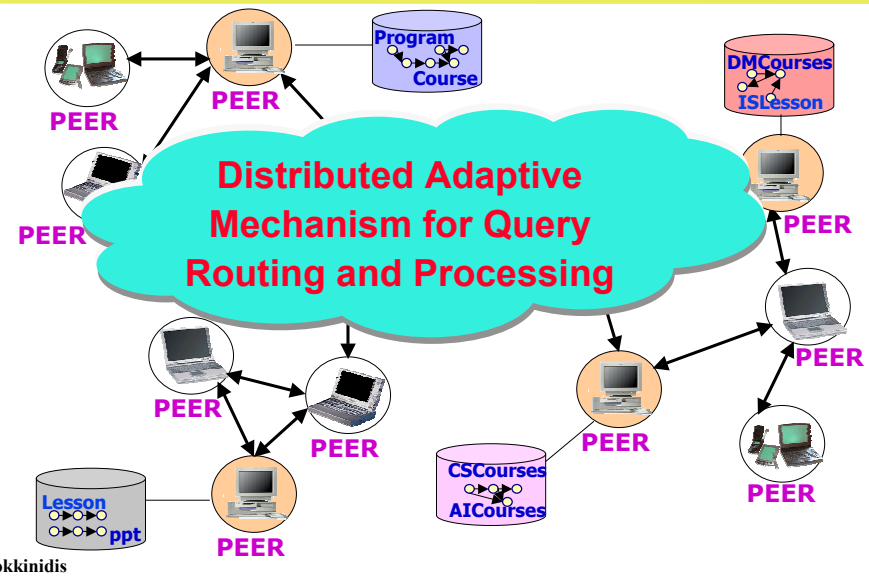
Motivation: E-Learning



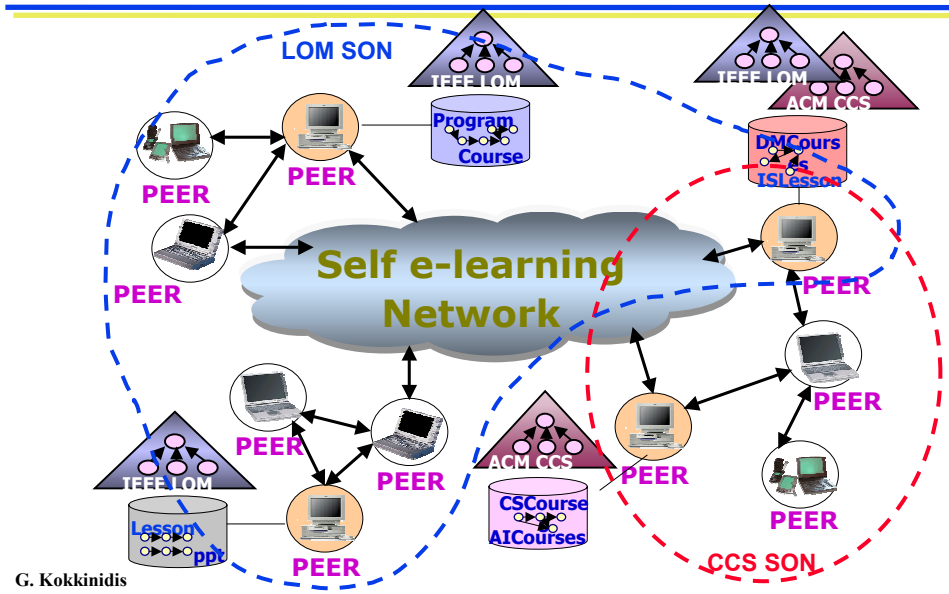
Motivation: E-Learning



Motivation: Peer-to-Peer E-Learning



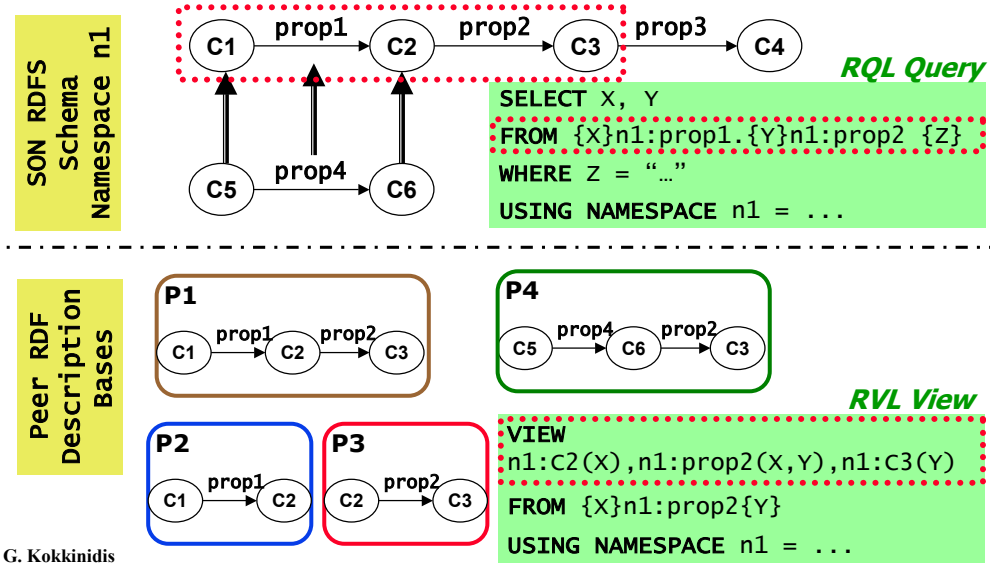
Semantic Overlay Networks based on RDF/S



Challenges and ICS-FORTH Contributions

- How peer nodes formulate queries?
  - Declarative RDF/S query languages: **RQL**
- How peer nodes advertise their bases?
  - Declarative RDF/S view specification languages: **RVL**
- How peer nodes route a query?
  - RQL query/RVL view subsumption techniques: **SWIM**
- How peer nodes process a query?
  - Semantic Query Plans and Peer Channels: **SQPeer**
- How distributed query plans are optimized?
  - Data/Query Shipping and run-time adaptability: **SQPeer**

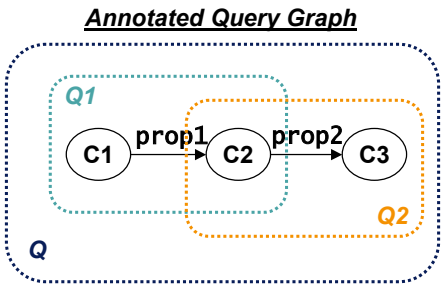
An RDF/S-based SON Example



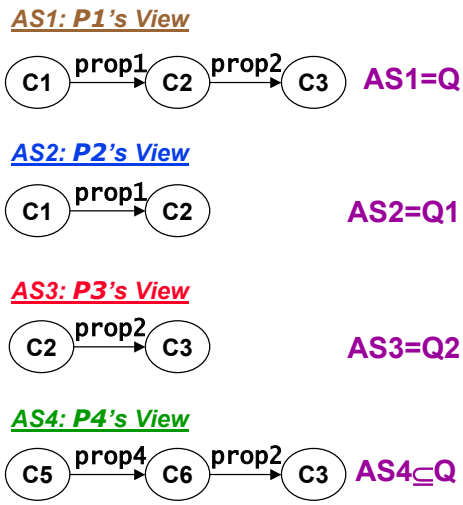
Why Query Patterns and Views?

- In SQPeer we capture intensionally the meaning of peer queries and what is actually stored in a peer database
  - the cost of maintaining peer views (i.e. schema information) is smaller compared to the cost of maintaining (XML or RDF) extensional indices
  - a peer receives only relevant to its content queries, thus decreasing the processing load of a P2P system
- RQL query/RVL view subsumption (sound and complete) techniques from *the ICS-FORTH SWIM middleware* are employed to
  - determine which part of a query can be effectively answered by a peer
  - rewrite the query sent to a peer according to
    - the available data on its base captured by its view (i.e. same SON)
    - the local schema using mapping rules to other peer schemas (i.e. different SONs)

Semantic Query Routing Example



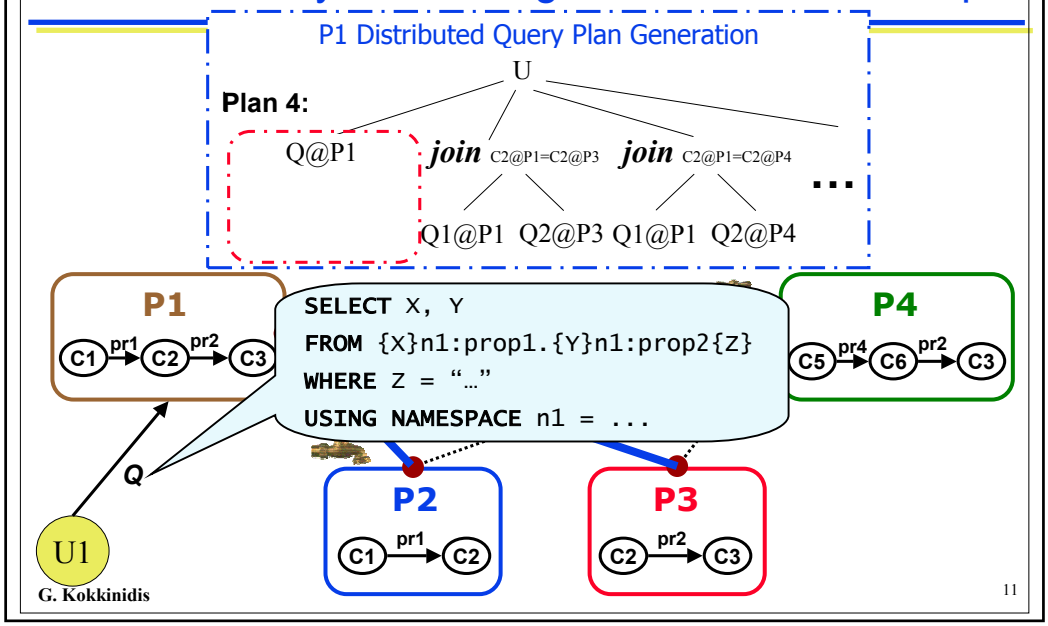
Q1	P1, P2, P4
Q2	P1, P3, P4



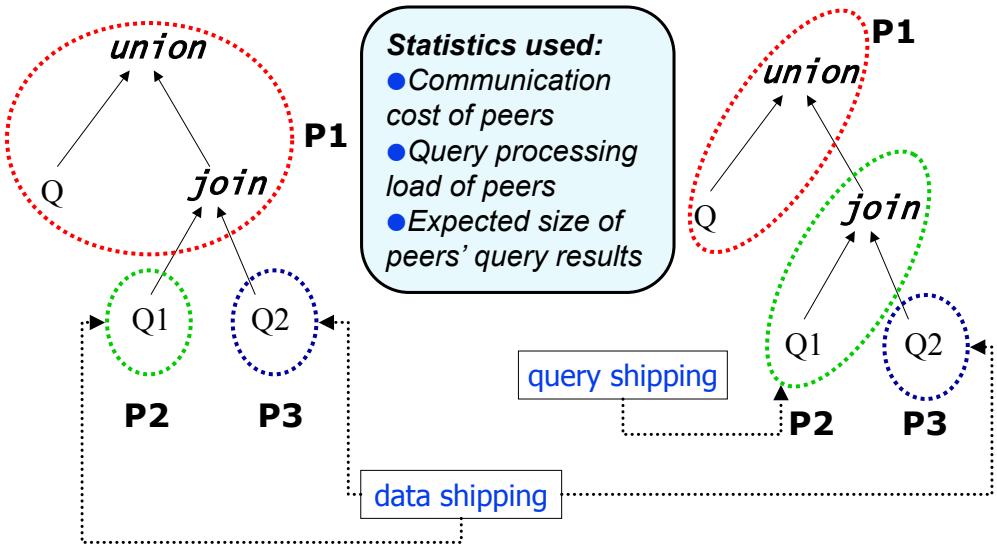
Query Plans and Channels

- The query-processing algorithm receives as input an annotated query graph and outputs its corresponding query plan
  - A *query plan* specifies how the query processing load should be distributed to the involved peers
  - *Communication channels* enable the distributed execution of these query plans
    - Establish physical connections between remote peer nodes
    - Enable the exchange of query results and possible statistics information
    - Handle the adaptability of the query plans
  - Data distribution (vertical, horizontal or mixed) in the P2P system is taken into account

Semantic Query Processing and Execution Example



Query Optimization: Data vs Query Shipping



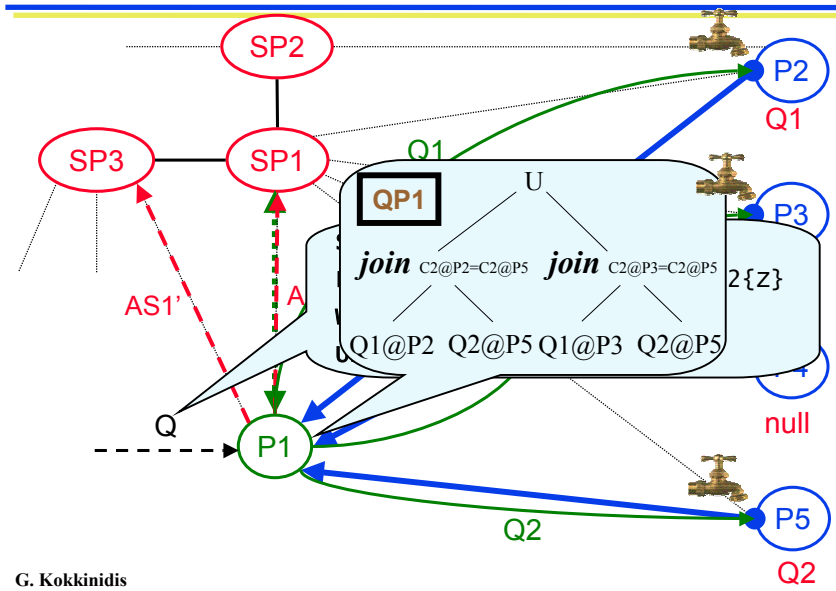
Adaptability

- Adapting is essential in a P2P system, where peers join or leave the system at free will, and due to network load and failures
- Changing plans at execution time usually requires the processing of already installed channels and of both query plans of the root and destination node of the channel. Changes include:
  - Data/Query shipping on the fly
  - Selecting alternative peers
- The root node should
  - Identify the problem
  - Inform all the necessary nodes that are affected by the change
  - Create a new query plan by re-executing the query-routing and query-processing algorithm
  - Execute the new query plan

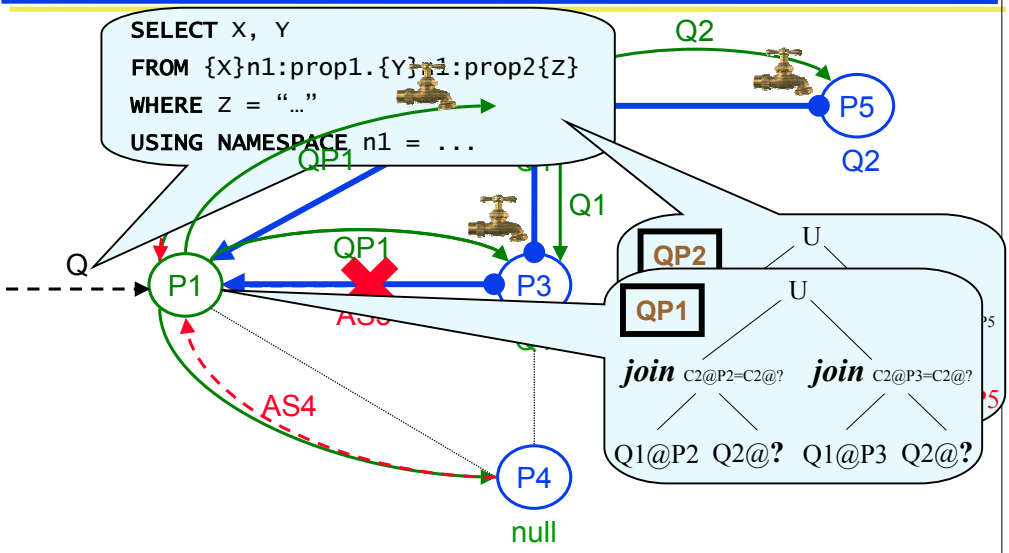
Architectural Alternatives

- SQPeer algorithms integrate smoothly with various P2P settings
- We distinguished two possible architectural alternatives according to the distribution of knowledge about peer nodes in a SON
  1. Hybrid P2P (super-peer and simple peers)
     - Each peer is aware of a super-peer, who stores all the active-schema information of the peers joined with it
     - Two separate phases: query routing (super peers), query processing (simple peers)
  2. Ad-hoc P2P (only simple peers)
     - A peer is not necessarily aware of active schemas of other peers
     - Two interleaved phases: query routing and processing (simple peers)

Super-Peer Example



Ad-Hoc Example (cont.)



Comparison with Related Work

- ***QueryFlow***, ***ubQL***, ***Mutant Query Plans*** and ***AmbientDB:*** address query processing issues in P2P database systems
  - Some of these systems require an a priori knowledge on where to find relevant peers
  - Few optimization opportunities are used for the execution of distributed query plans
- ***Edutella*** and ***RDFPeers:*** address mainly query routing issues in SONs
  - Index-based approaches
  - No run-time adaptability of the query plans
- **SQPeer** exploits the full power of RDF/S-based SONs for both hybrid and adhoc P2P systems
  - Subsumption of RDF classes/properties is not considered by any related project

Summary

- We propose the ICS-FORTH SQPeer middleware for routing and processing semantic RQL queries (peer requests) and RVL views (peer advertisements) in a distributed P2P system
- We proposed a query-routing and a query-processing algorithm for generating distributed query plans that consider data distribution
- Compile and run-time optimizations were discussed for handling the execution of these query plans
- Finally, we presented different architectural alternatives, on which the SQPeer can be implemented

Future Work

- Implementation ongoing
- Several issues remain open with respect to the optimization of distributed query plans and the statistics that need to be handled
  - Study the trade-off between result completeness and processing load
  - Possible use of restrictions regarding the number of peer nodes that each query is broadcasted and further processed
- Use of DHTs for RDF/S schemas with subsumption information
- Hierarchical organization of super-peers using RDF/S schemas articulations

SW Layer Cake and ICS-FORTH Vision

