ProMo – A Scalable and Efficient Middleware for Real-Time Online Data Delivery

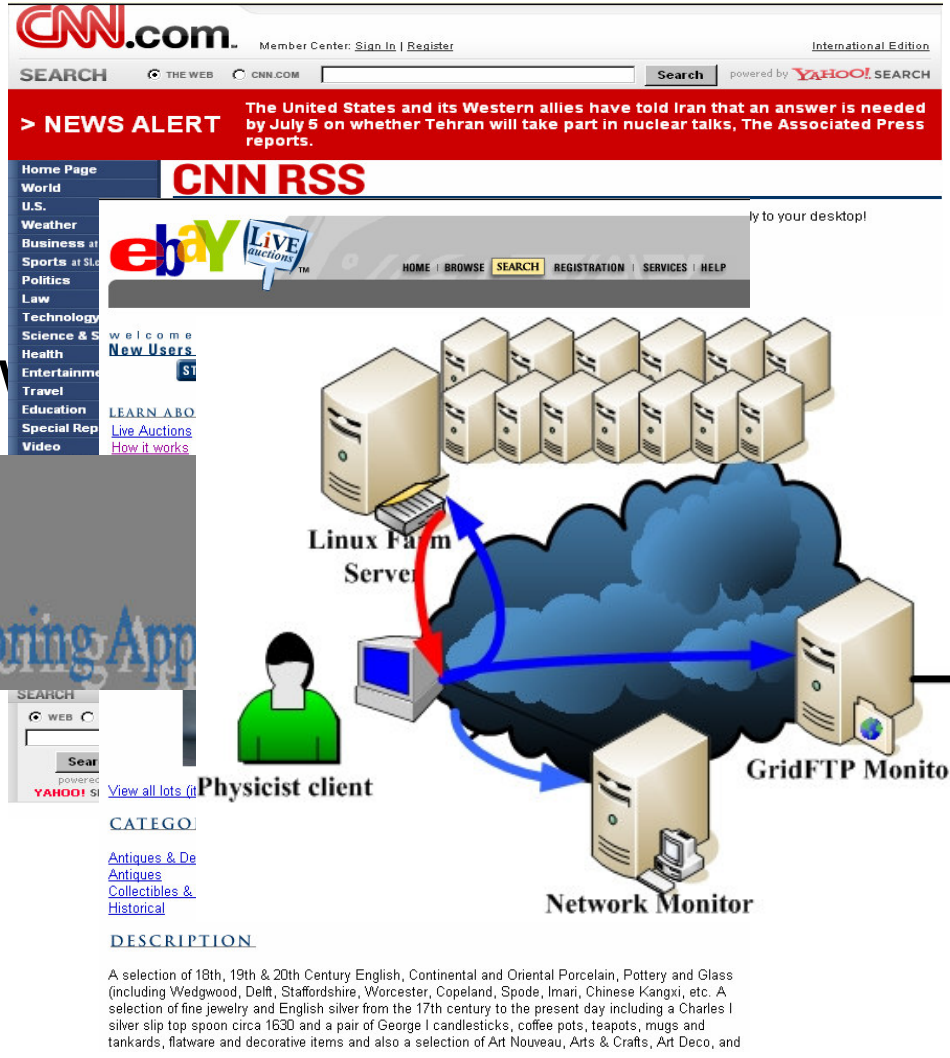
Haggai Roitman, **Avigdor Gal**

Louiqa Raschid

ProMo – A Scalable and Efficient Middleware for Real-Time Online Data Delivery

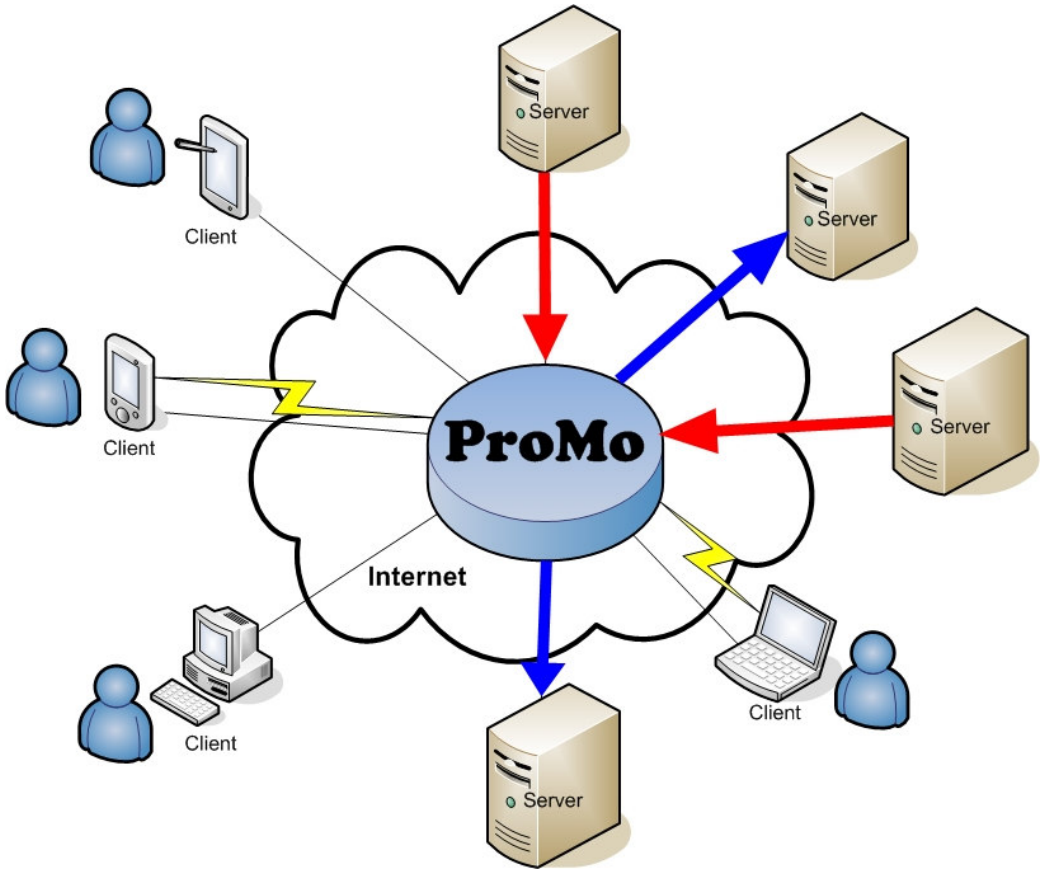
Online Data Delivery

- News syndication
- Auction progress
- Stock market follo
- Gr



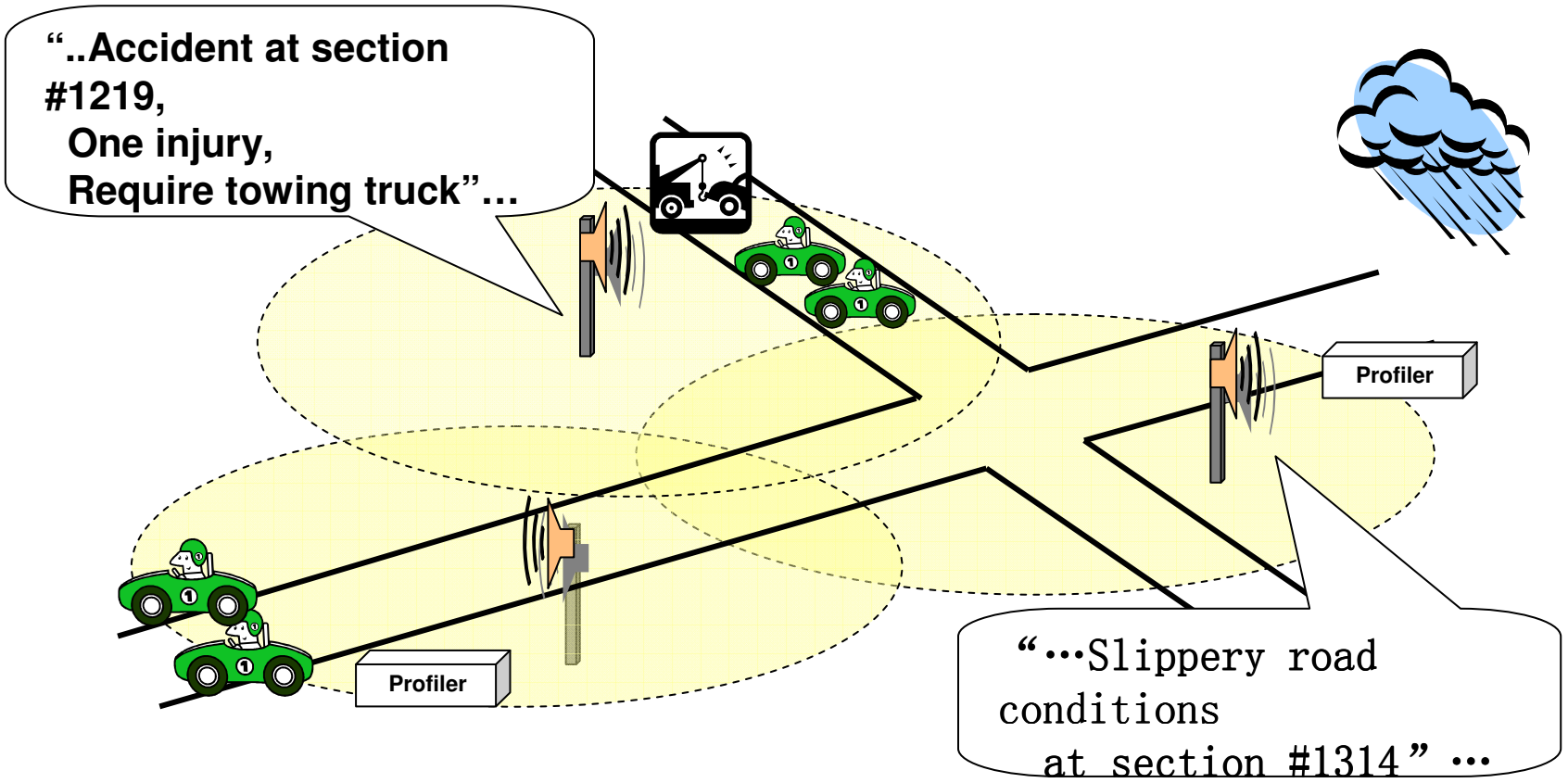
The Role of a Middleware in Online Data Delivery

- Push-based:
  - Scalability?
  - Personalization?
  - Accuracy?
- Pull-based
  - Scalability?
  - Personalization?
  - Accuracy?
- Hybrid

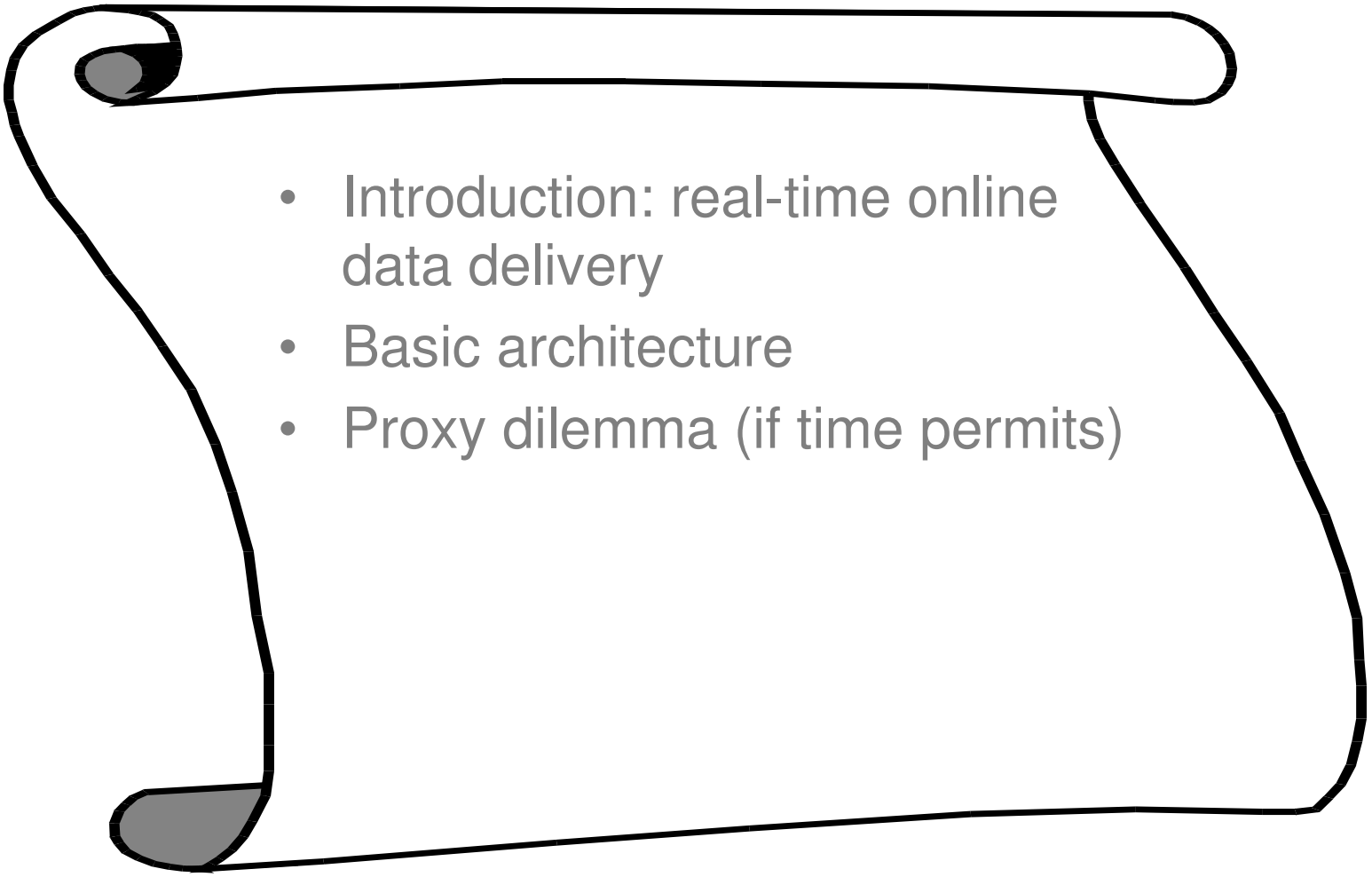


ProMo – A Scalable and Efficient Middleware for Real-Time Online Data Delivery

Real-Time Data Delivery



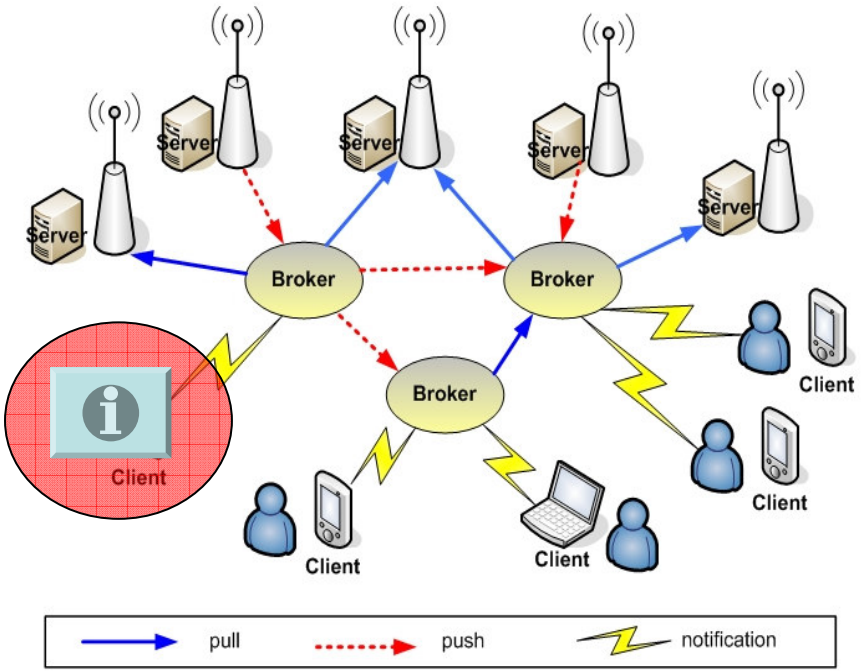
Outline



- Introduction: real-time online data delivery
- Basic architecture
- Proxy dilemma (if time permits)

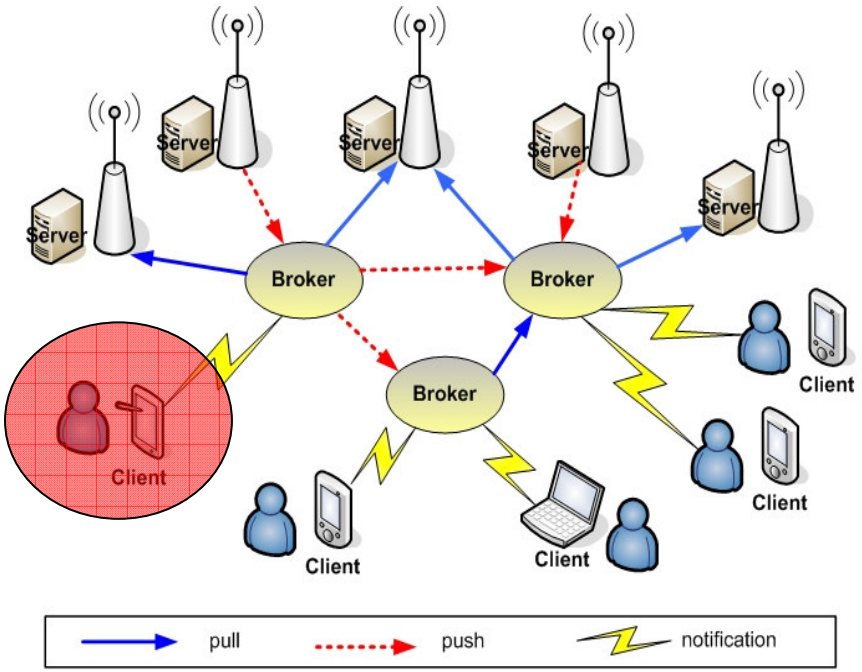
Basic Architecture

- Client:
  - Inform me whenever X new updates arrive.
  - Inform me as soon as there is an increase of Y% in value.
  - Profile and satisfiability
  - Constraints: No more than *C* monitoring tasks in a chronon



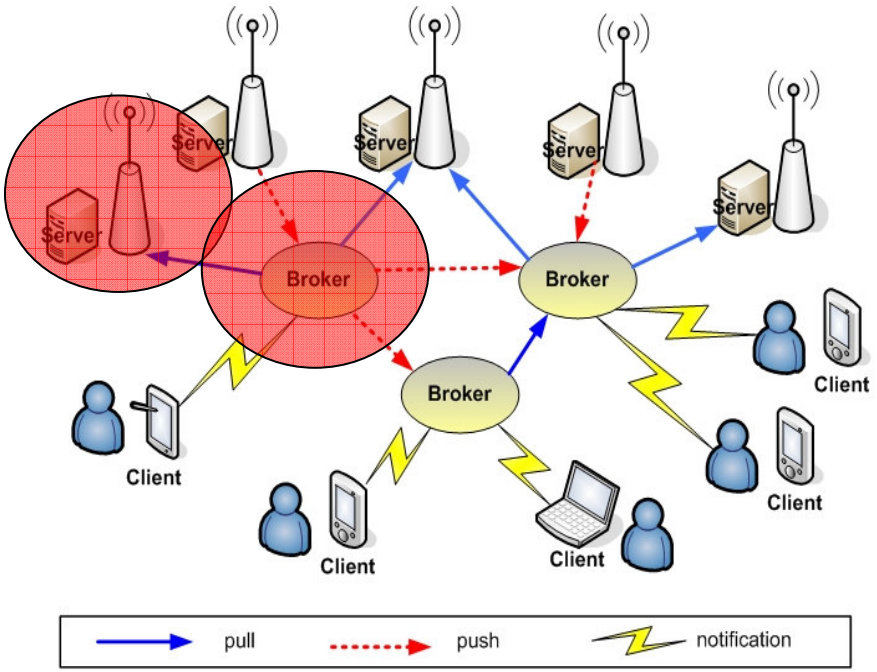
Basic Architecture

- Server:
  - Server abilities:
    - Push every update.
  - Server constraints:
    - Polite probe constraint
      - Minimal interval of disconnection
      - Maximal number of probes per epoch
  - Update models



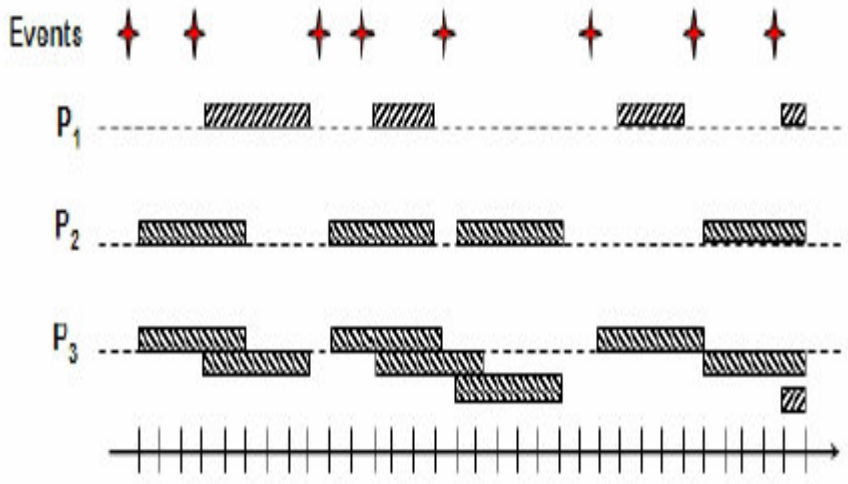
Basic Architecture

- Mediator:
  - Matches capabilities
  - Manages constraints
  - Increases scalability
  - Offloading
  - Adaptive scheduling



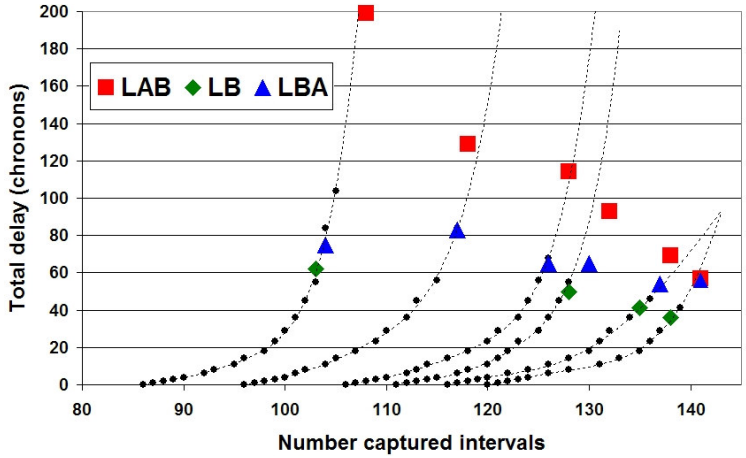
The Proxy Dilemma

"…Given a set of client profiles (currently with no preferences between them) and an upper bound on the number of resources that can be monitored at each chronon, maximize the number of notifications to clients while minimizing the total delay…"



Solving the Proxy Dilemma

- First attempt: Pareto sets

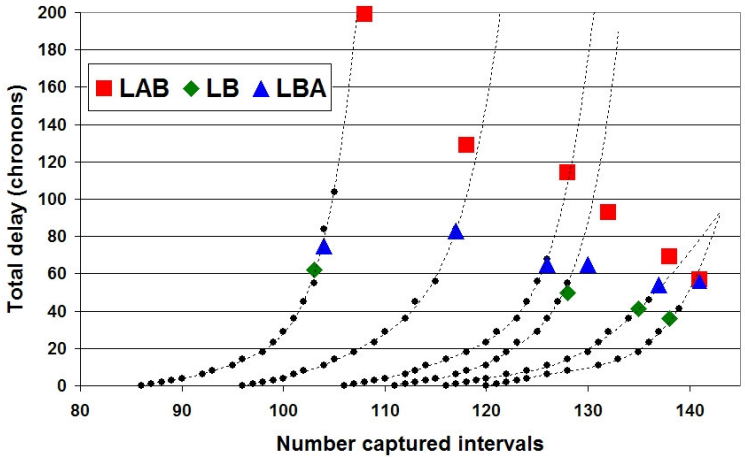


LEMMA 1. *Given n resources, K chronons, and a constraint $\vec{C}$ on the number of probes per chronon, all feasible schedules can be enumerated in $\Theta(n^{KC_{\max}})$ time, where $C_{\max} = \max_{j=1,2,\ldots,n}(C_j)$.*

Expensive

Solving the Proxy Dilemma, cont.

- Second attempt: Approximated Pareto sets

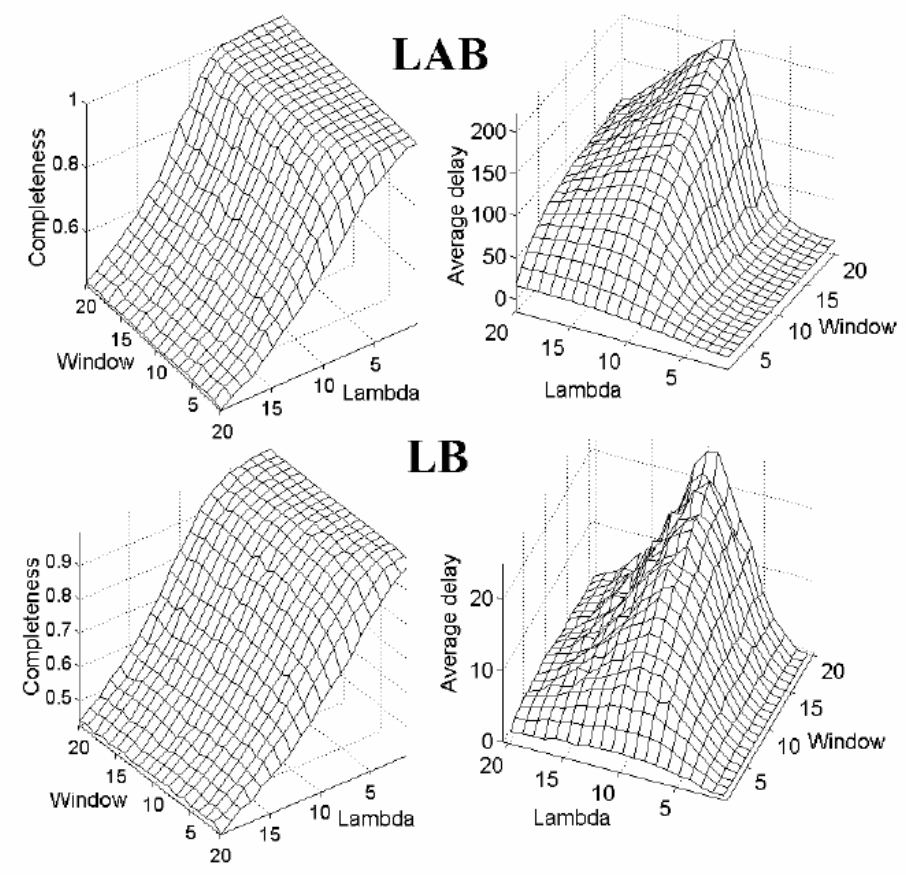


  - FPTAS solution to the General Assignment Problem (GAP)
  - Complexity: $O\left(c(\varepsilon)\frac{n^2K^2}{\varepsilon}\right)$

Still expensive

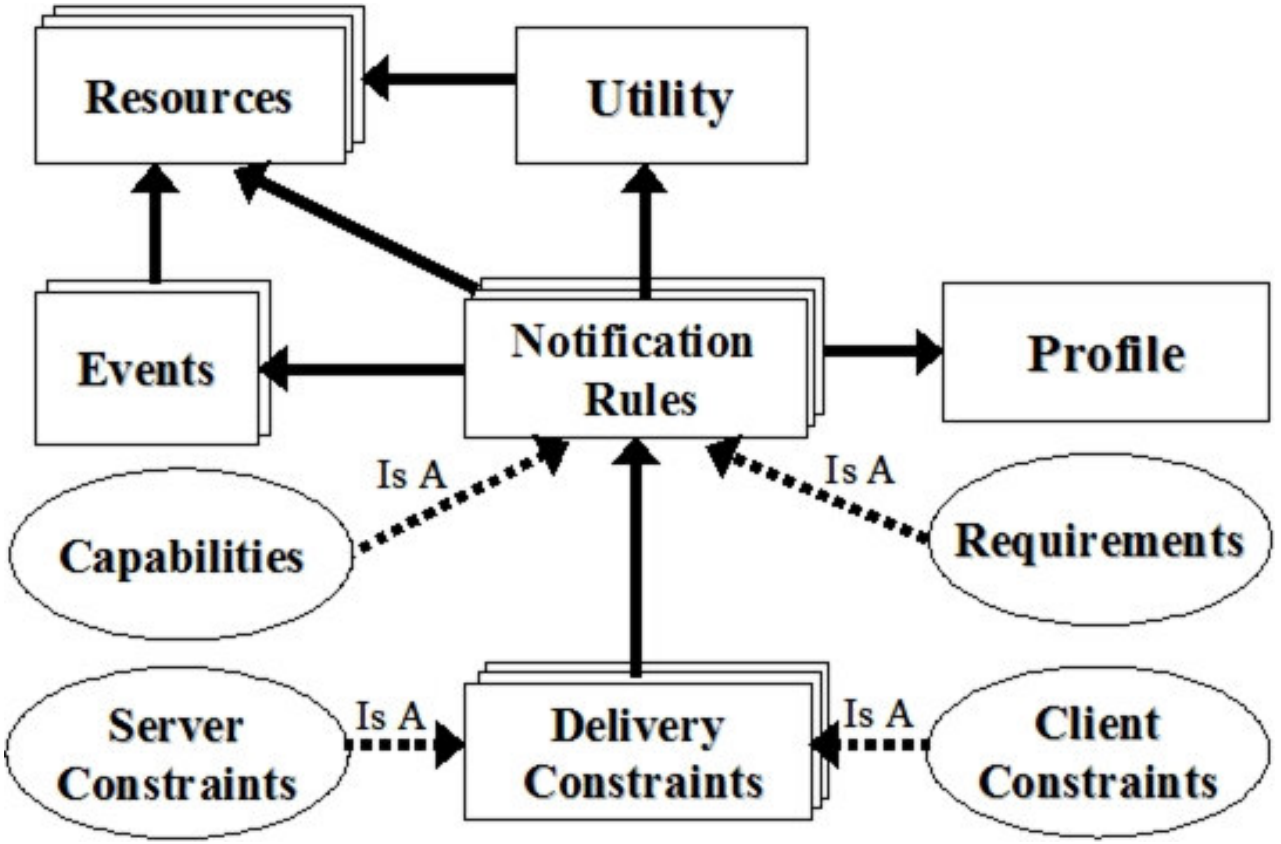
Solving the Proxy Dilemma, cont.

- Third attempt: Heuristics
  - We sacrifice inter-interval relationships
  - Interval potential function:
    - Look Ahead (LA)
    - Look Back (LB)
    - LAB (Look Ahead & Back)
    - LBA (Look Back & Ahead)

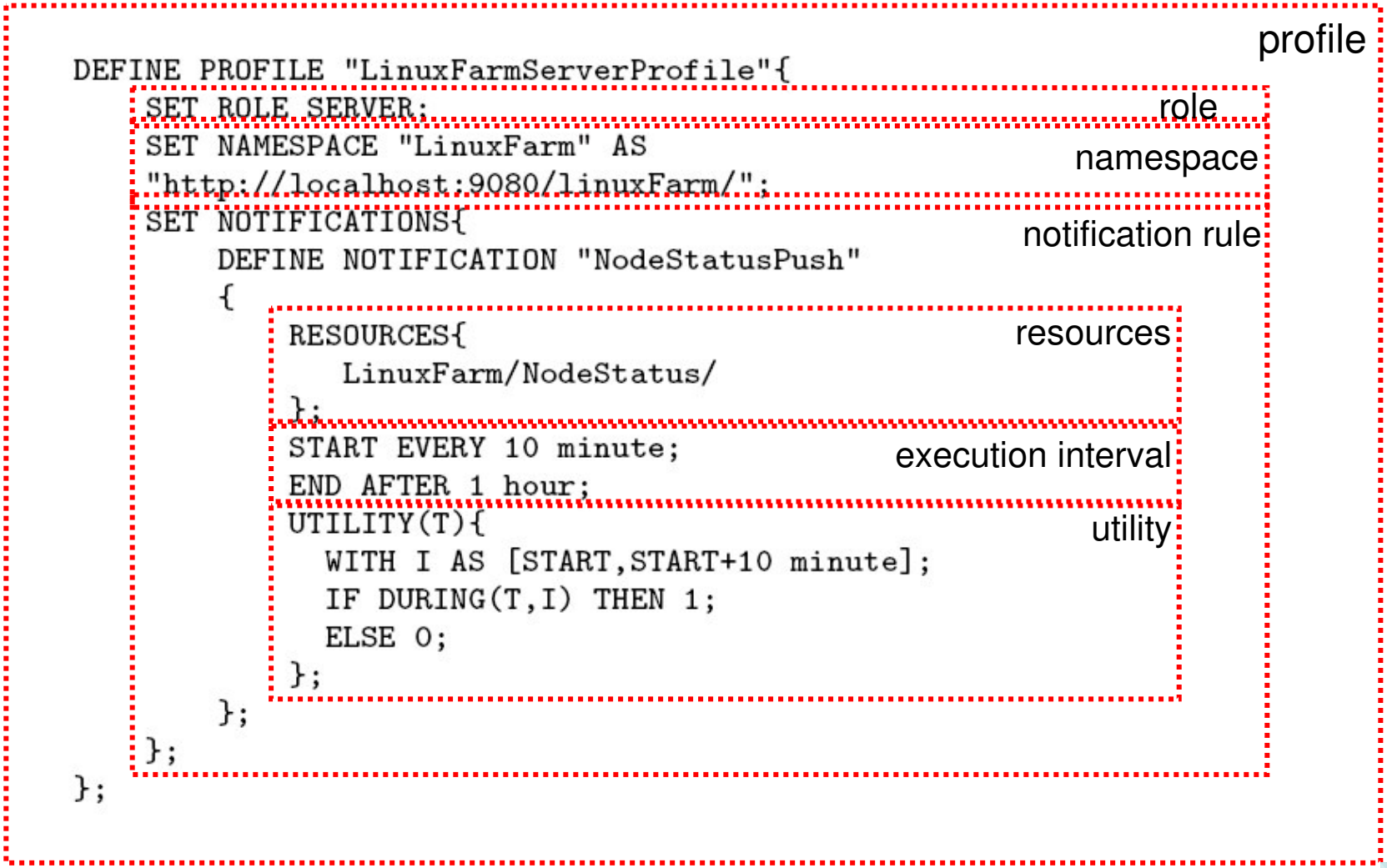



THANK
YOU

Profile Model

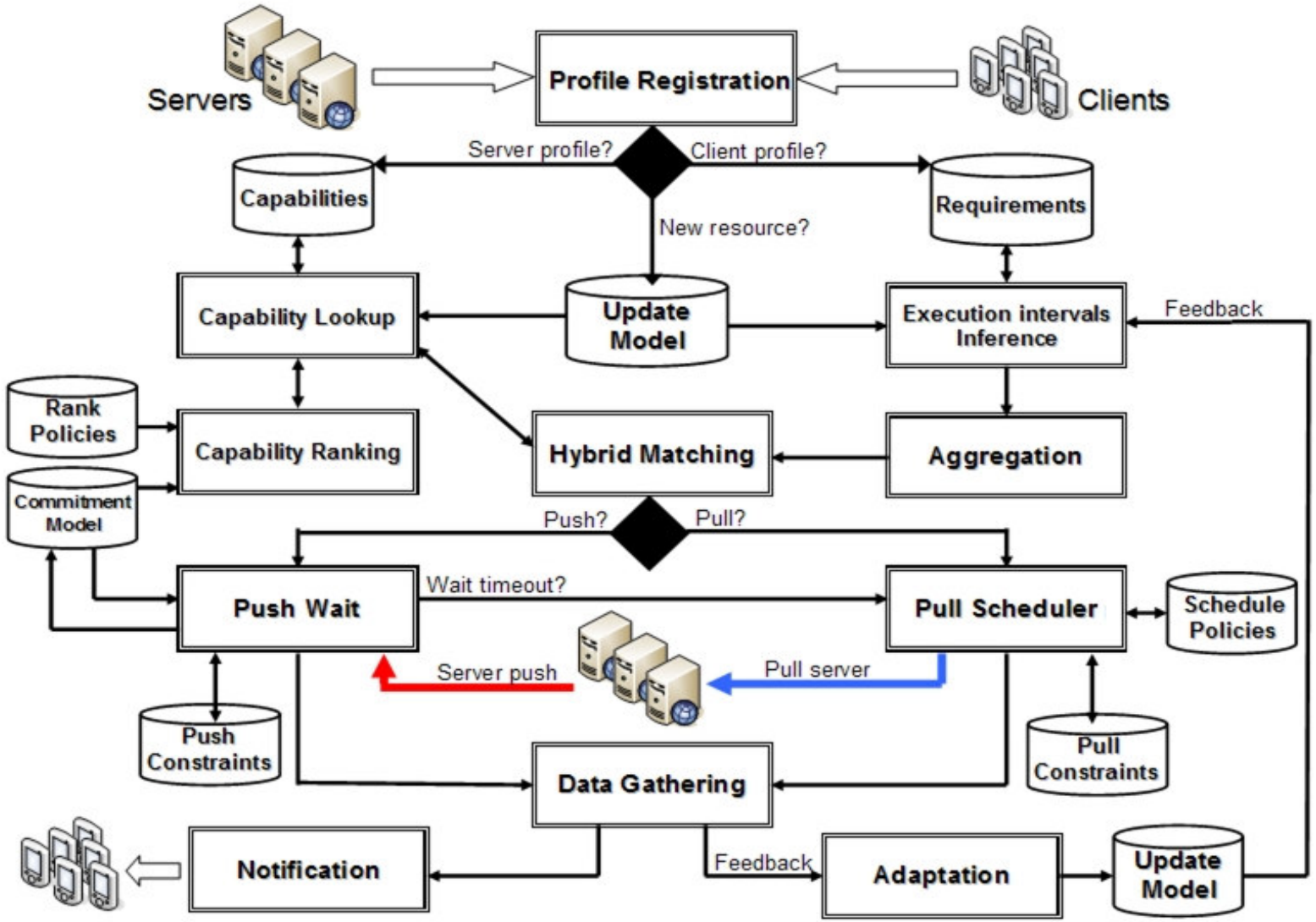


Profile example

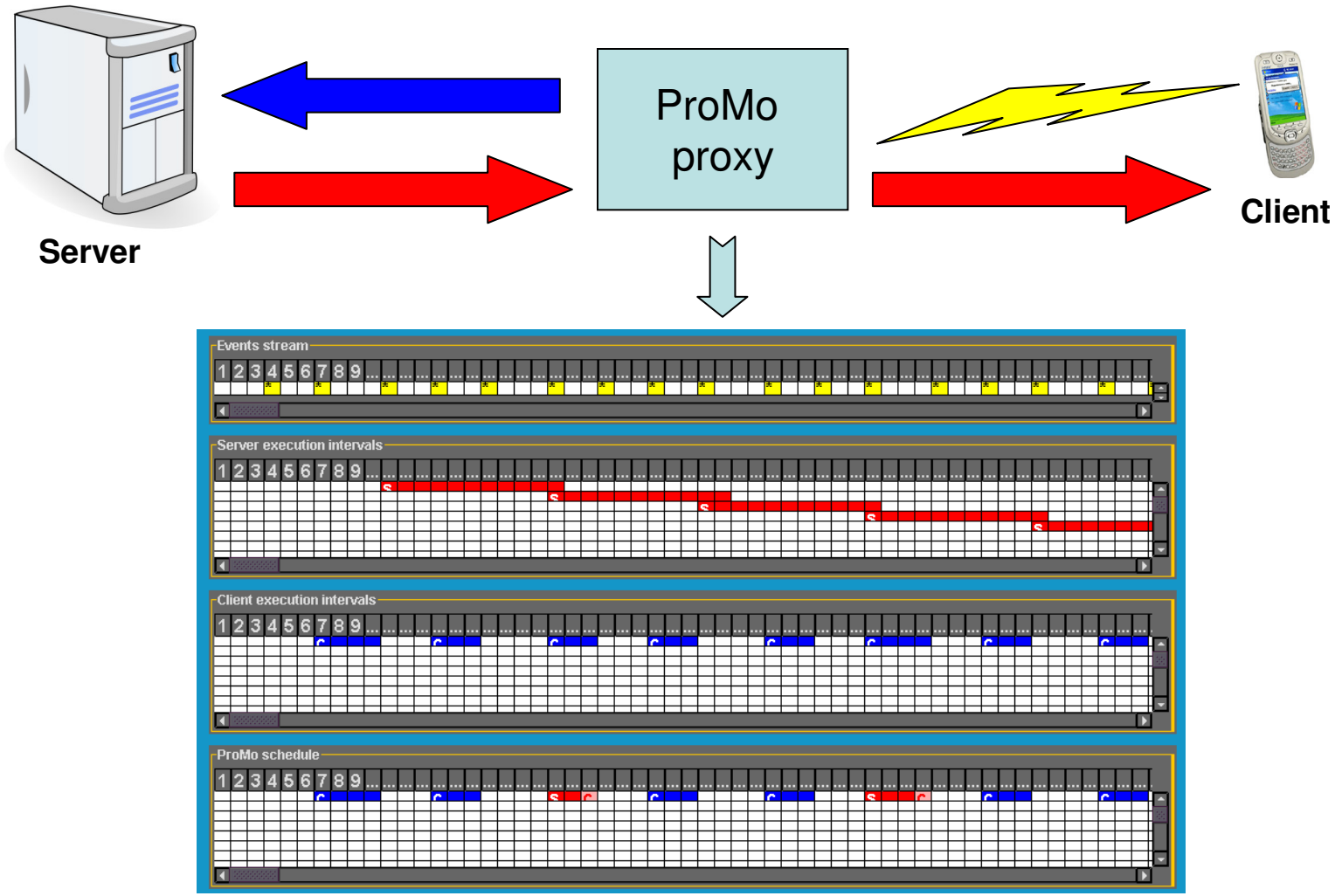



```
DEFINE PROFILE "LinuxFarmServerProfile"{
    SET ROLE SERVER;
    SET NAMESPACE "LinuxFarm" AS
    "http://localhost:9080/linuxFarm/";
    SET NOTIFICATIONS{
        DEFINE NOTIFICATION "NodeStatusPush"
        {
            RESOURCES{
               LinuxFarm/NodeStatus/
            };
            START EVERY 10 minute;
            END AFTER 1 hour;
            UTILITY(T){
              WITH I AS [START,START+10 minute];
              IF DURING(T,I) THEN 1;
              ELSE 0;
            };
        };
    };
};
```

ProMo broker design



Capability matching



Promo demo available to download at: http://tx.technion.ac.il/~haggair/myPapers/ProMoDemo.zip

Constraint management (samples)

- Maximize client utility (or minimize client cost)
  - s.t. system resources (e.g., # of probes per chronon)
- Minimize system resource utilization
  - s.t. client needs
- Minimize system resource utilization
  - s.t. server constraints (e.g., politeness)