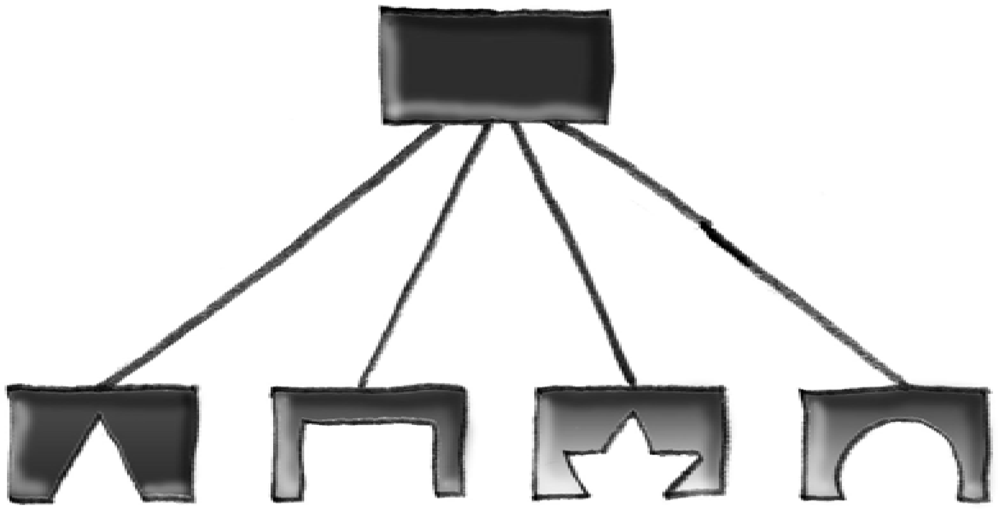
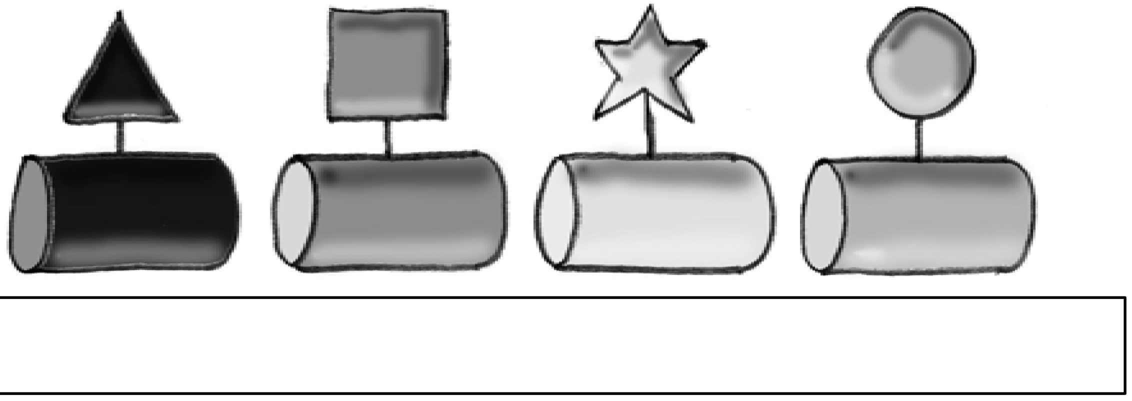
Swedish
Institute of
Computer
Science

Novel features of the MOMENT query language SQUEME

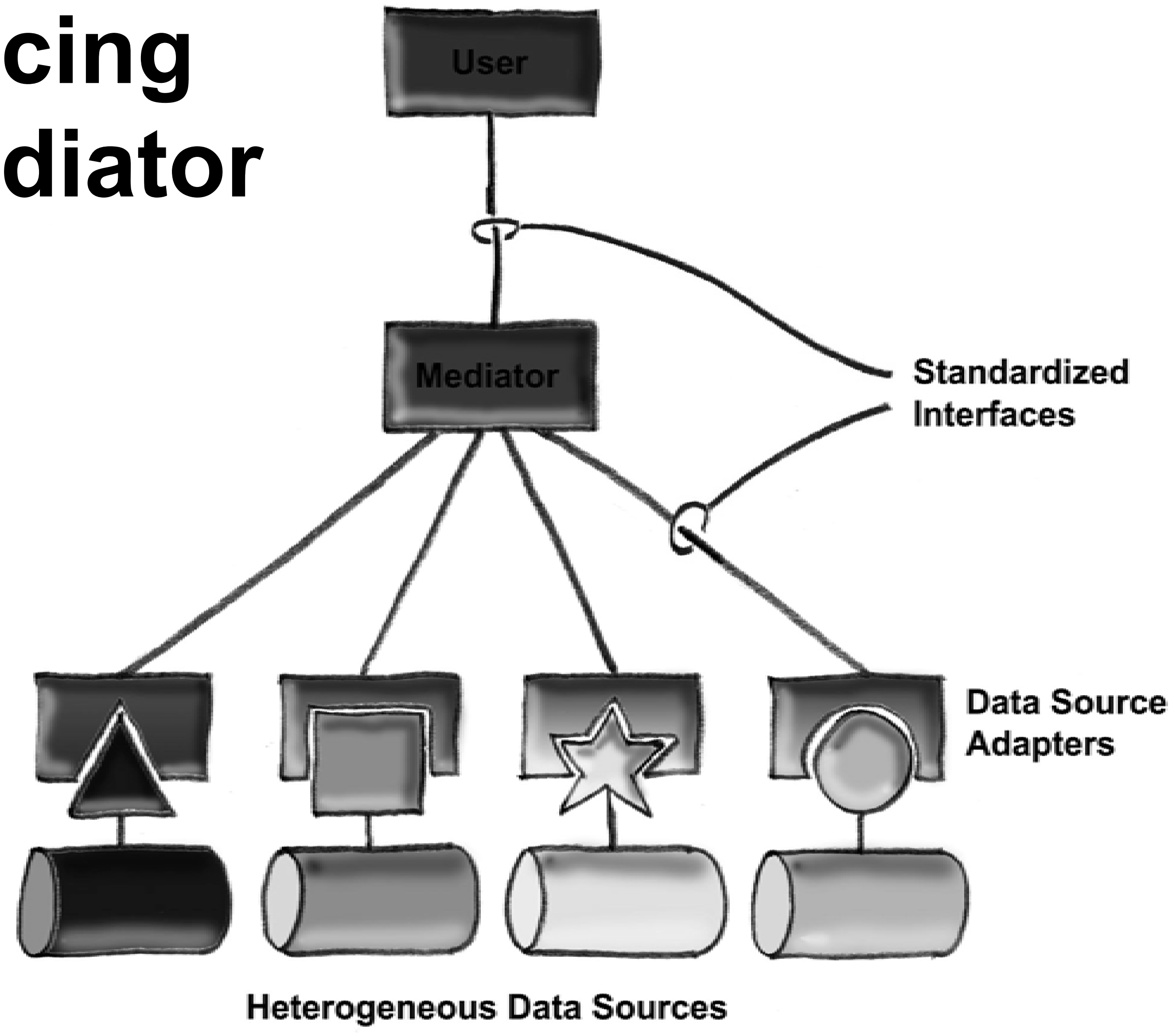
Martin Nilsson, SICS
Contact: from dot ecworkshop at drnil dot com



Handling Massive Flows of Heterogeneous Network Measurements

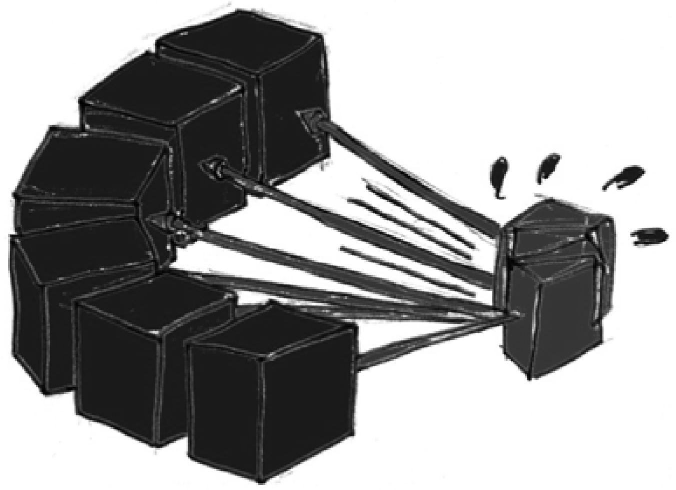


Introducing The Mediator

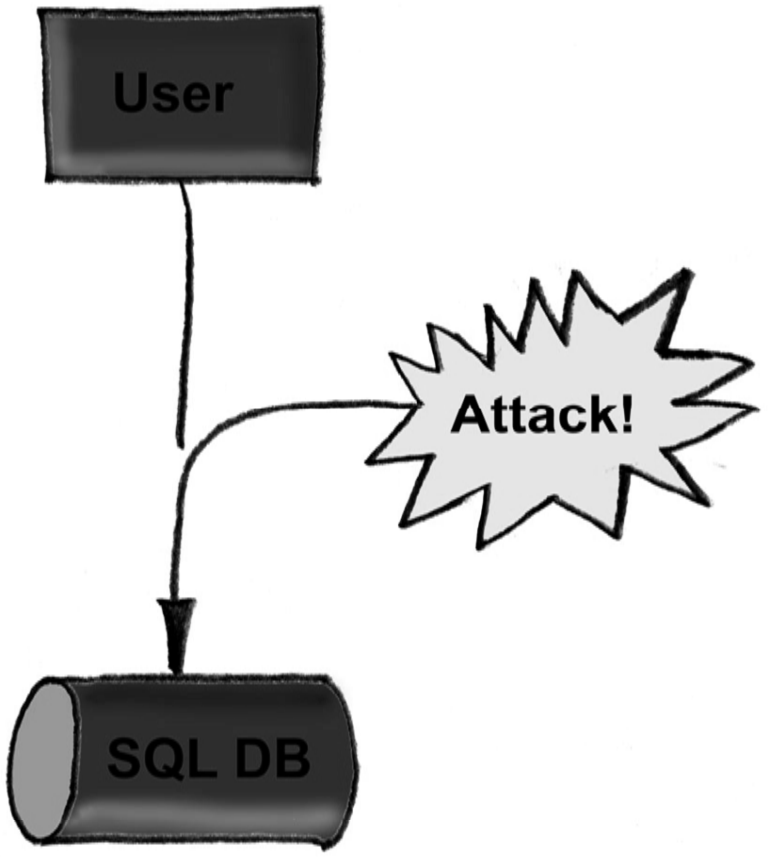


Problem 1: Overload

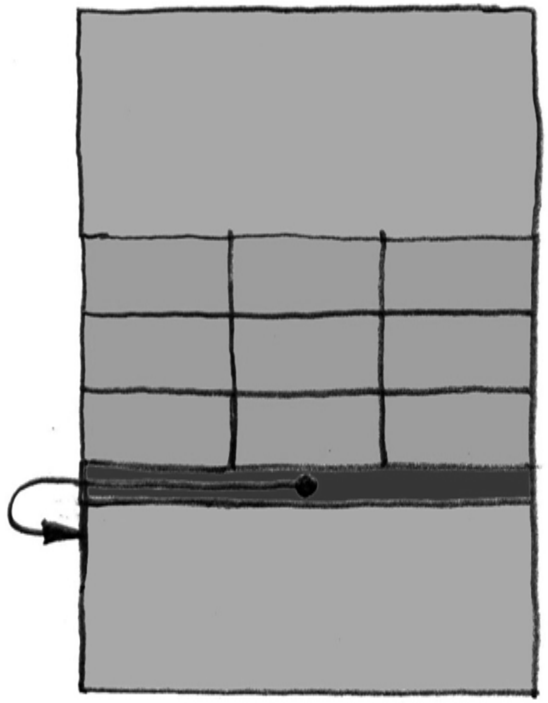
Server chokes client


Clients choke server

Problem 2: SQL attacks



Solution 1: Lazy tables



Only a segment of data is transferred on each connection

…plus a **continuation** for requesting the remaining data

Solution 2: SQUEME

SQL:

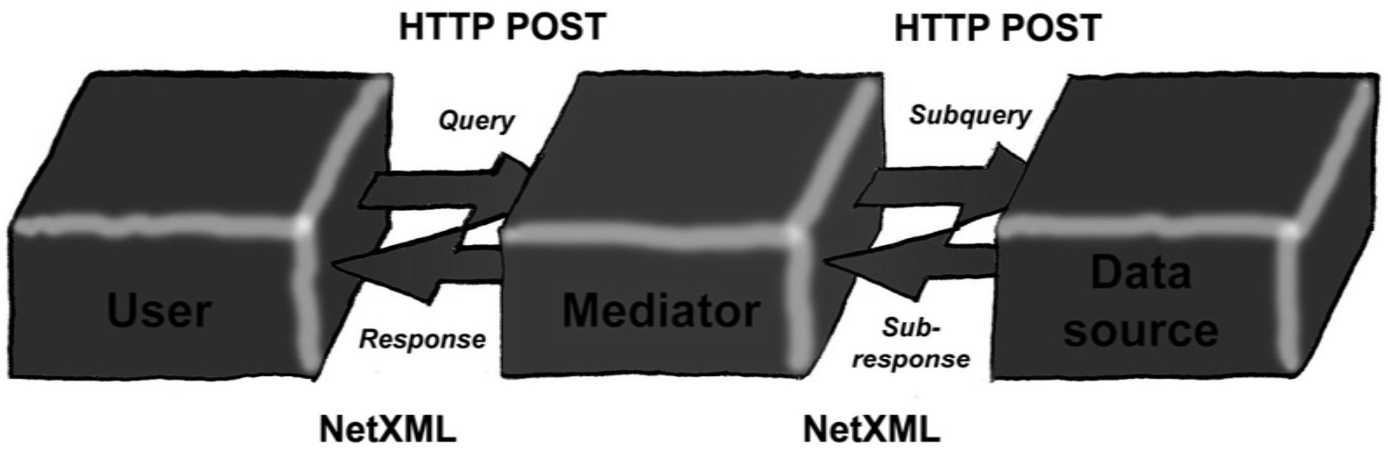
```
SELECT x FROM table WHEN x < 17;
```

SQUEME:

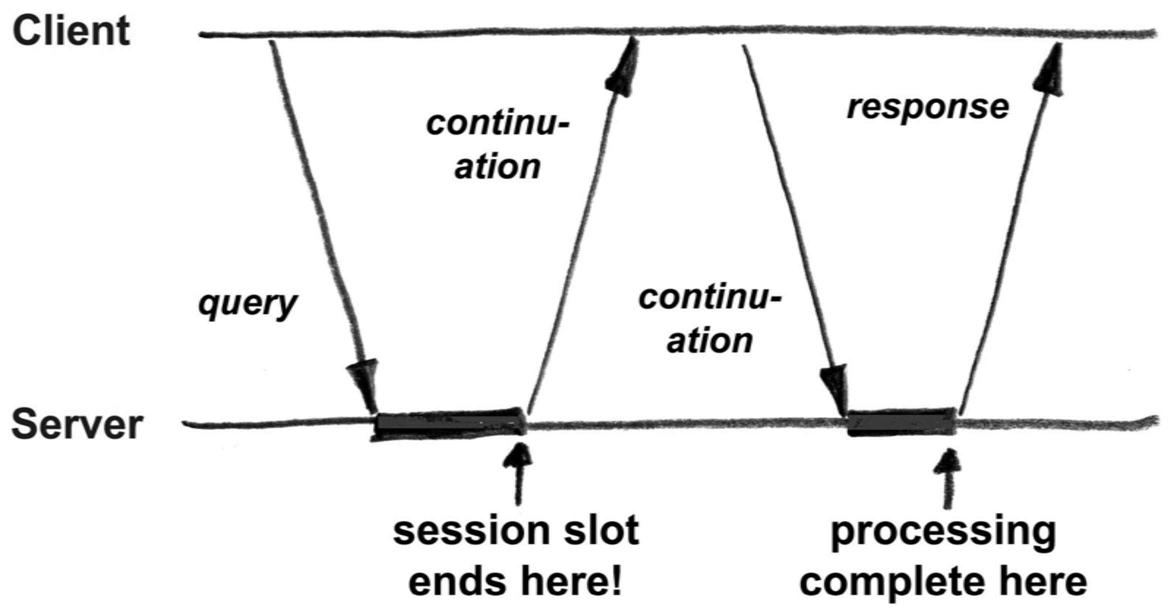
```
select=x &
from=table &
when=(less x 17)
```

SQUEME = SQL semantics + Scheme syntax
encoded as HTTP POST

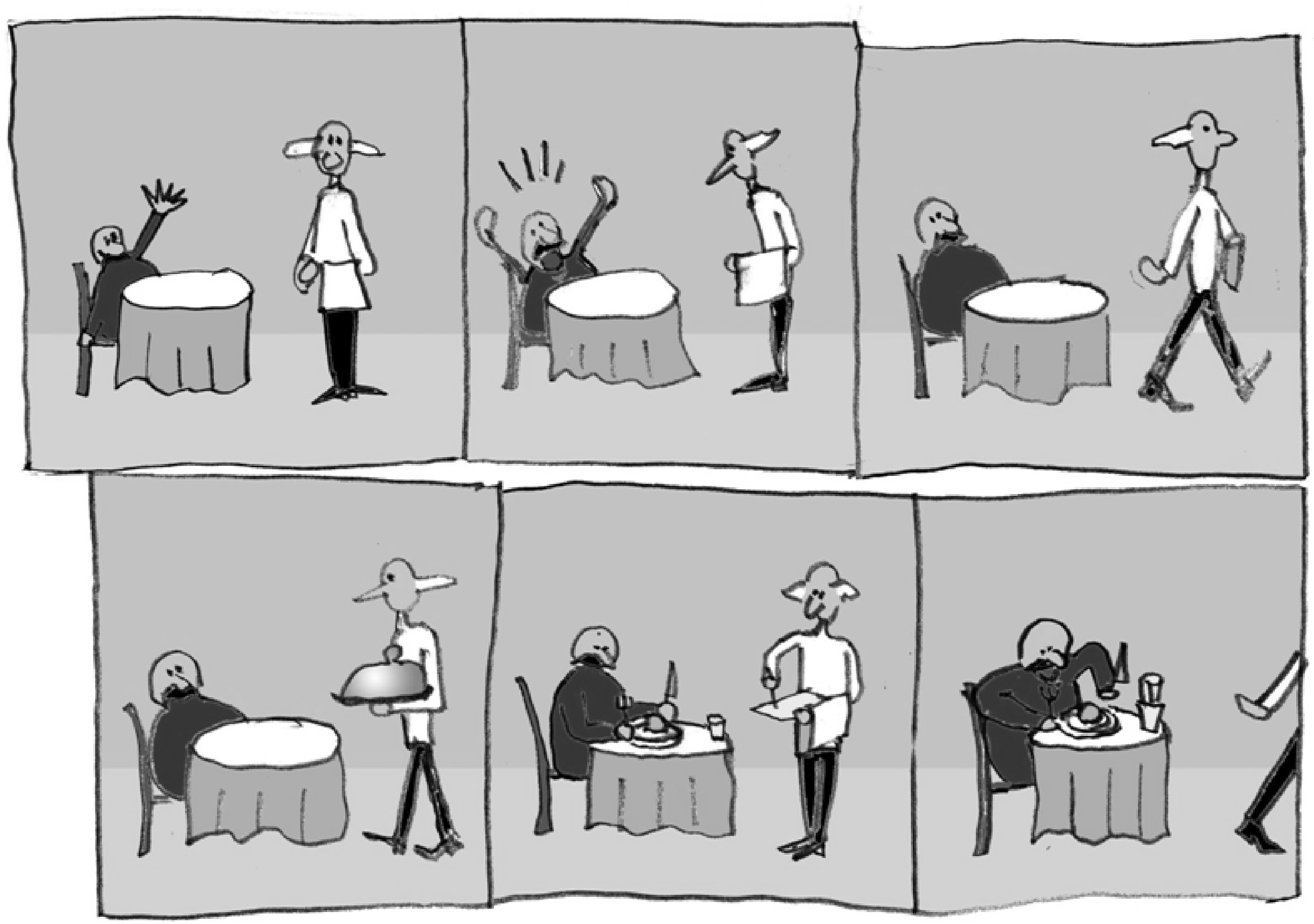
Basic communication



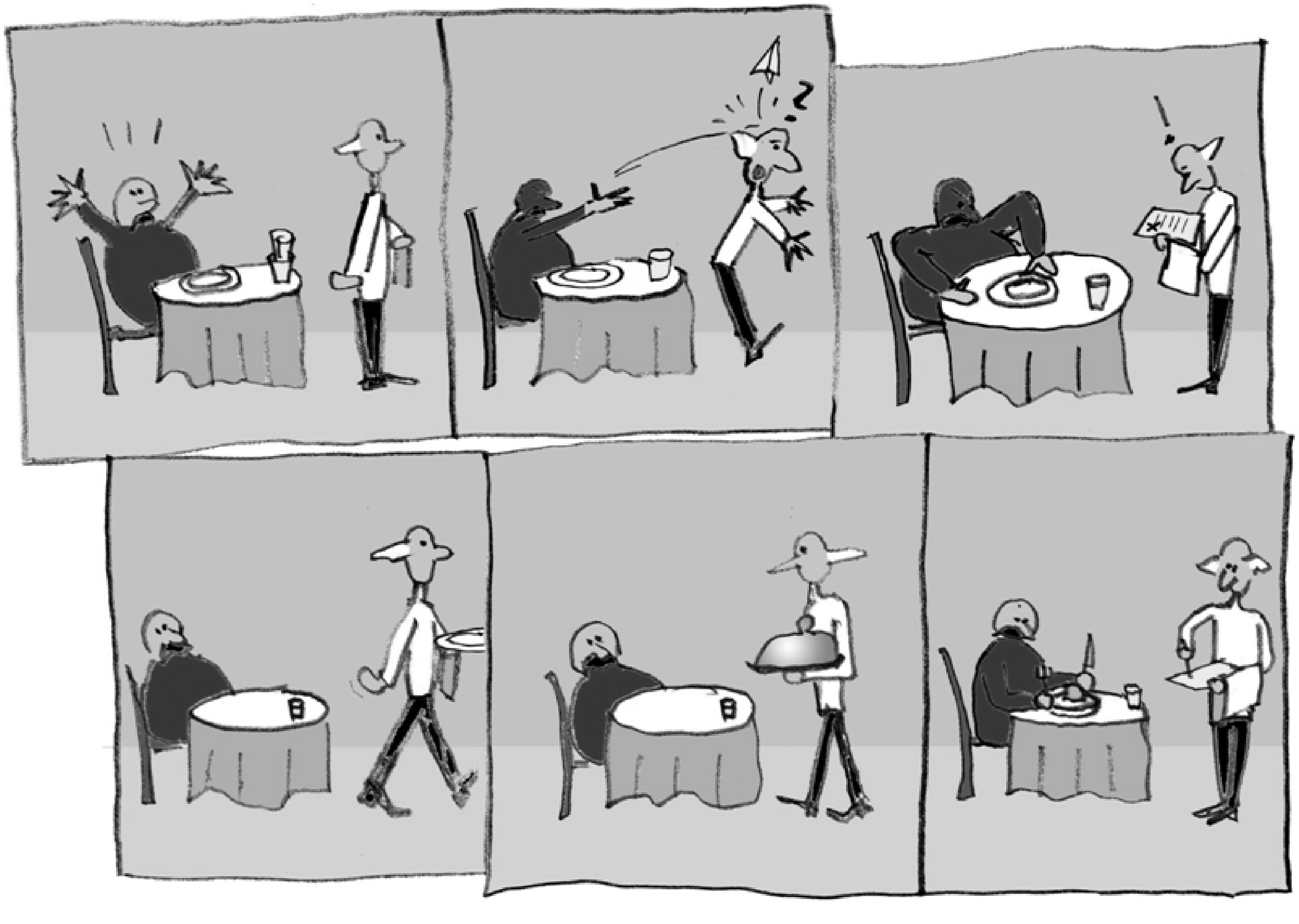
Continuations: The Mechanism behind Lazy Tables



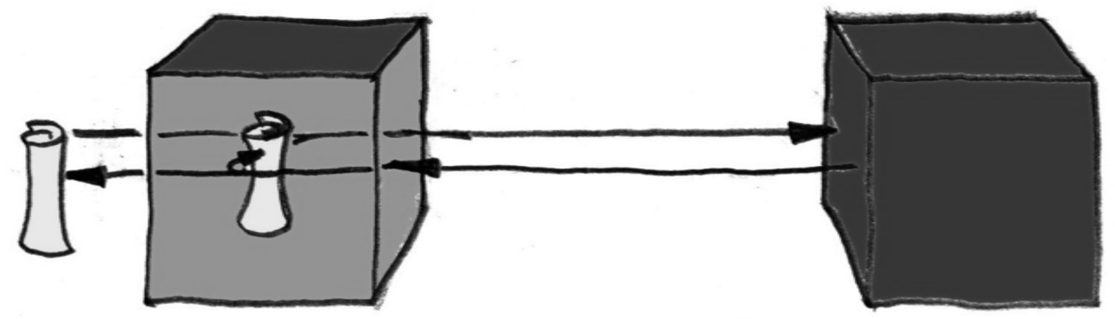
The Restaurant Metaphor 1



The Restaurant Metaphor 2



Lazy Table/Continuation features



- Enables **preemption** of server processing
- Enables transfer of **infinite data streams**
- **No persistent** data needed on server
- Makes **session ID superfluous**
- **Eliminates** risk for **session hijacking**
- **Only server** needs to be **able to read** it

SQUEME Main Keywords

- select
- from
- where
- group_by
- having
- order_by
- limit
- offset

Limit and **offset** are important for Lazy Tables!

SQUEME Features

- Handles **full SQL** semantics
- Easy to **parse**
- Easy to **validate/check** for injection attacks
- **Maps nicely to HTTP** POST/GET without tricky URL encoding
- Can be used **directly from browser**
- **”Classical” SQL** can easily be **generated** from SQUEME

SQUEME Example

SQL:

```
select  cast(delay/1000 as int) as delay , count_big(*) as count
from experiment.RawDelayData r, Experiment.EndToEnd m
where r.e2eID=m.e2eID and m.e2eid = 134440
group by source, destination, cast(delay/1000 as int)
order by source, destination, cast(delay/1000 as int);
```

SQUEME as HTTP GET query:

```
select=(as (cast (div delay 1000) int) delay),
            (as (count_big *) count) &
from=(as experiment.RawDelayData r),
          (as Experiment.EndToEnd m) &
where=and,
              (equal r.e2eID m.e2eID),
              (equal m.e2eid 134440) &
group_by=source,destination,(cast (div delay 1000) int) &
order_by=source,destination,(cast (div delay 1000) int)
```

Conclusions

Handling large data bases and other massive-flow sources of measurement data
in a safe and user-friendly manner is feasible, using the techniques of Lazy Tables and SQUEME.

These techniques have been implemented and tested in a Mediator, developed in the EC FP7 MOMENT project.