

Building Web Applications without a DBMS

Donald Kossmann

28msec, Inc. & ETH Zurich

Symposium in honor of Klaus Dittrich

- July 3 - 4, University of Zurich
 - Please, mark your calendars
- International program with current DB topics
- More information:
<http://www.ifi.uzh.ch/krdsym/>
 - (under construction)

Building Web Applications without a DBMS

Donald Kossmann

28msec, Inc. & ETH Zurich

Building Web Applications without a DBMS

Donald Kossmann

28msec, Inc. & ETH Zurich

Agenda

- Why DBMS are evil?
- Reinventing Data Management on the Web
- Why XQuery?
- Putting it all together (28msec, Inc.)

Goal

- Web 1.0: Everybody publishes *data*
 - e.g., personal Web pages
- Web 1.5: Big guys publish *services*
 - e.g., Salesforce.com, Oracle, Microsoft, ...
- Web 2.x: *Everybody* publishes *services*
 - e.g., car pooling, parent portal, ...

Software Engineering 101

- Step 1: Brainstorming – Have ideas
- Step 2: Build it – write code
- Step 3: Run it – make \$\$\$
- Step 4: Evolution (new ideas) – goto Step 2

Software Engineering 101

- Step 1: Brainstorming – Have ideas
- Step 2: Build it – write code
- Step 3: Run it – make \$\$\$
- Step 4: Evolution (new ideas) – goto Step 2

All this is fun

The devil is in the detail

- Step 1: Brainstorming – Have ideas
- Step 2: Build it – write code
 - find the right schema
 - build test infrastructure
- Step 3: Run it – make \$\$\$
 - deployment: SW + HW configuration
 - Administration: patches, crashes, ...
 - Management: monitor cost
- Step 4: Evolution (new ideas) – goto Step 2



All this is ~~fun~~ expensive

Why are DBMS great? [Kemper 2006]

- **Redundancy, Inconsistency**

- storage is cheap; reality is inconsistent [Vogels07]



- **Declarative programming**

- data independence more important than ever



- **Multi-user environments**

- but, ACID is the wrong model



- **Durability**

- but, update in place is „history“ [Gray 2006]



- **Security**

- but, nobody uses DB-level security



- **Cheap**

- but, Joe Doe cannot afford Oracle



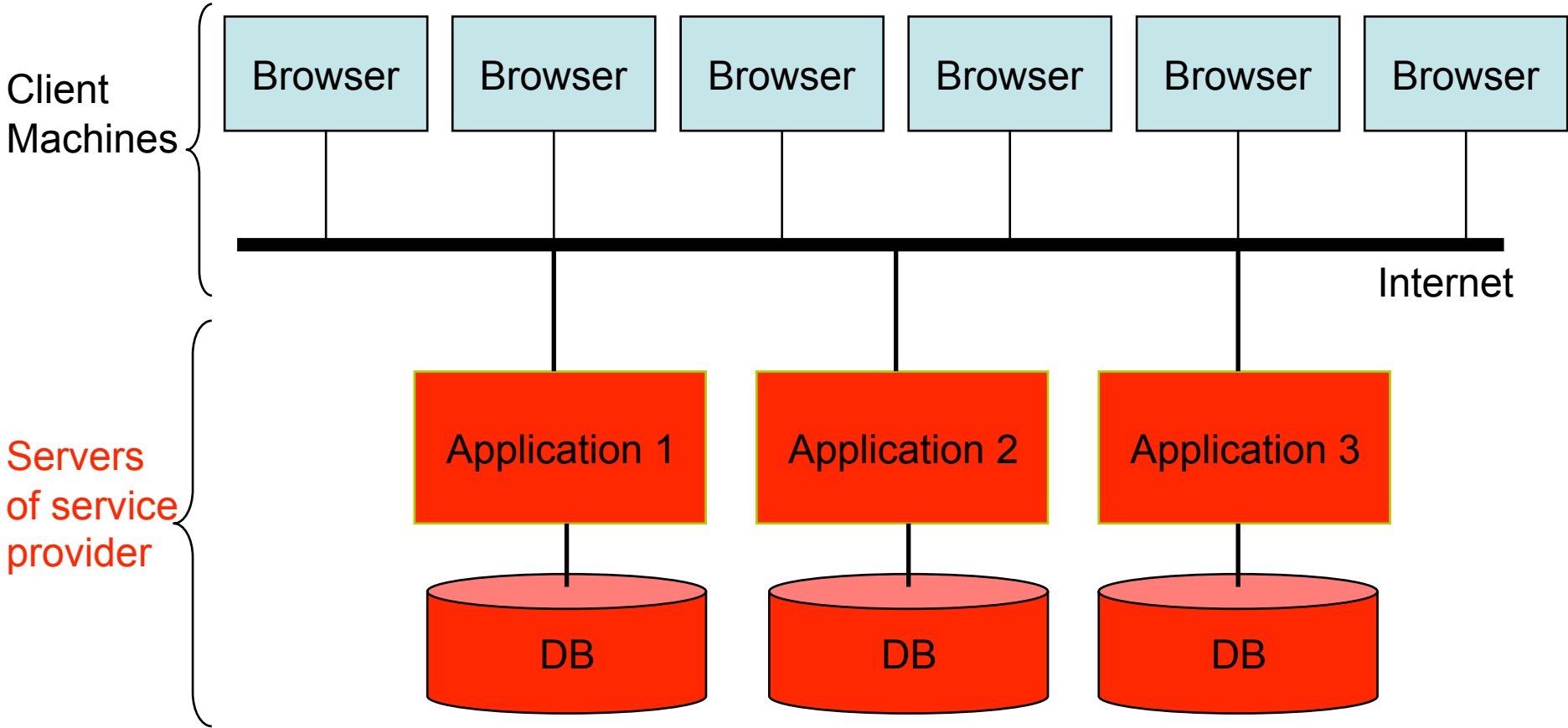
Why are DBMS expensive?

- DBMS do not help with any devils
 - Logging, security, etc. are done at app level
- DBMS pronounce all the little devils
 - DBMS requires schema upfront
 - DBMS complicates test infrastructure
 - DBMS needs to be installed and configured
 - DBMS requires a machine + disks
 - DBMS needs to be administrated, upgraded
- DBMS are inflexible, dictate architecture
 - Mike S's „one size does not fit all“ talk goes here

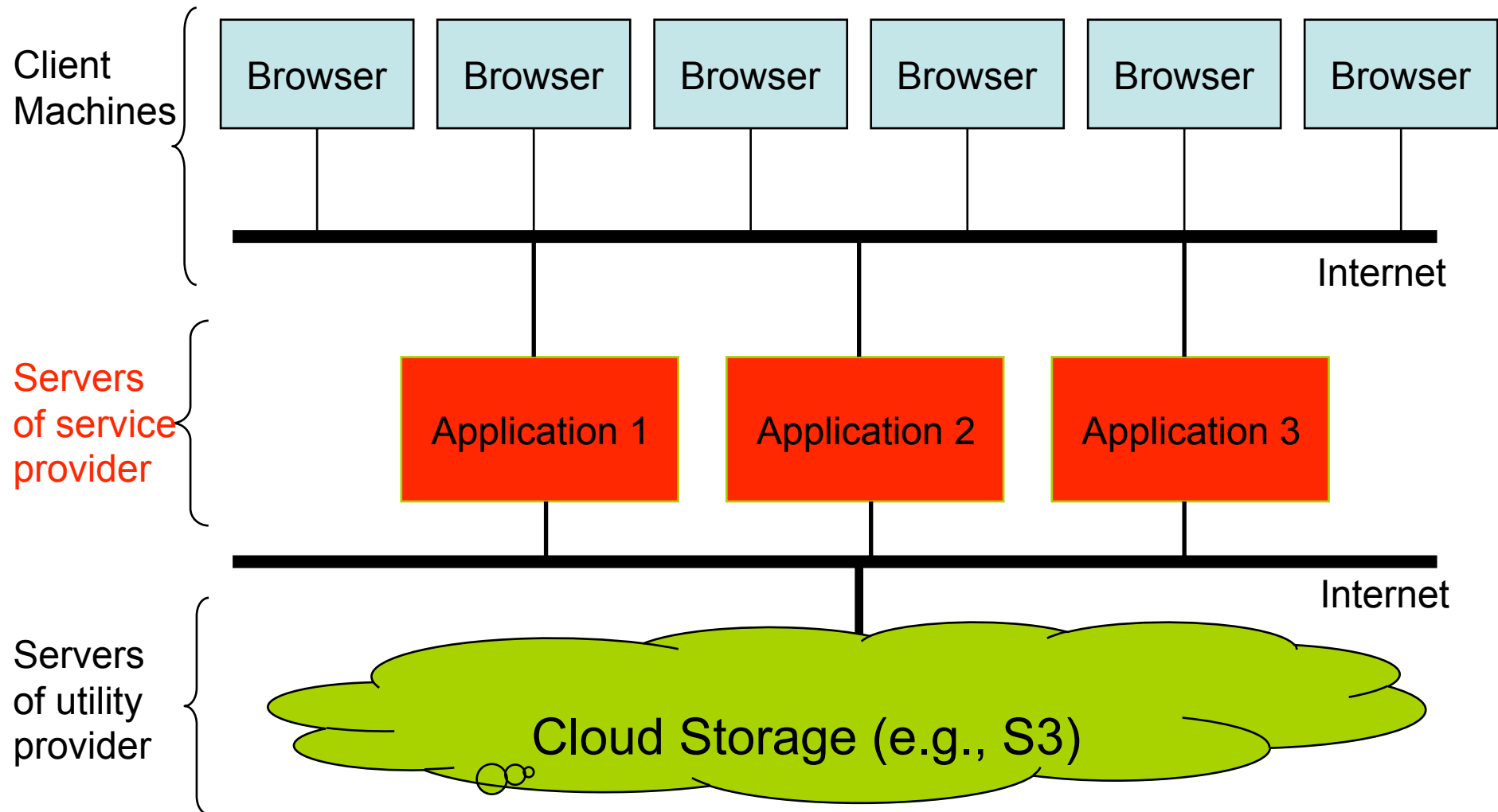
Agenda

- Why DBMS are evil?
- **Reinventing Data Management on the Web**
- Why XQuery?
- Putting it all together

State of the Art



Step 1: Remove the DBMS

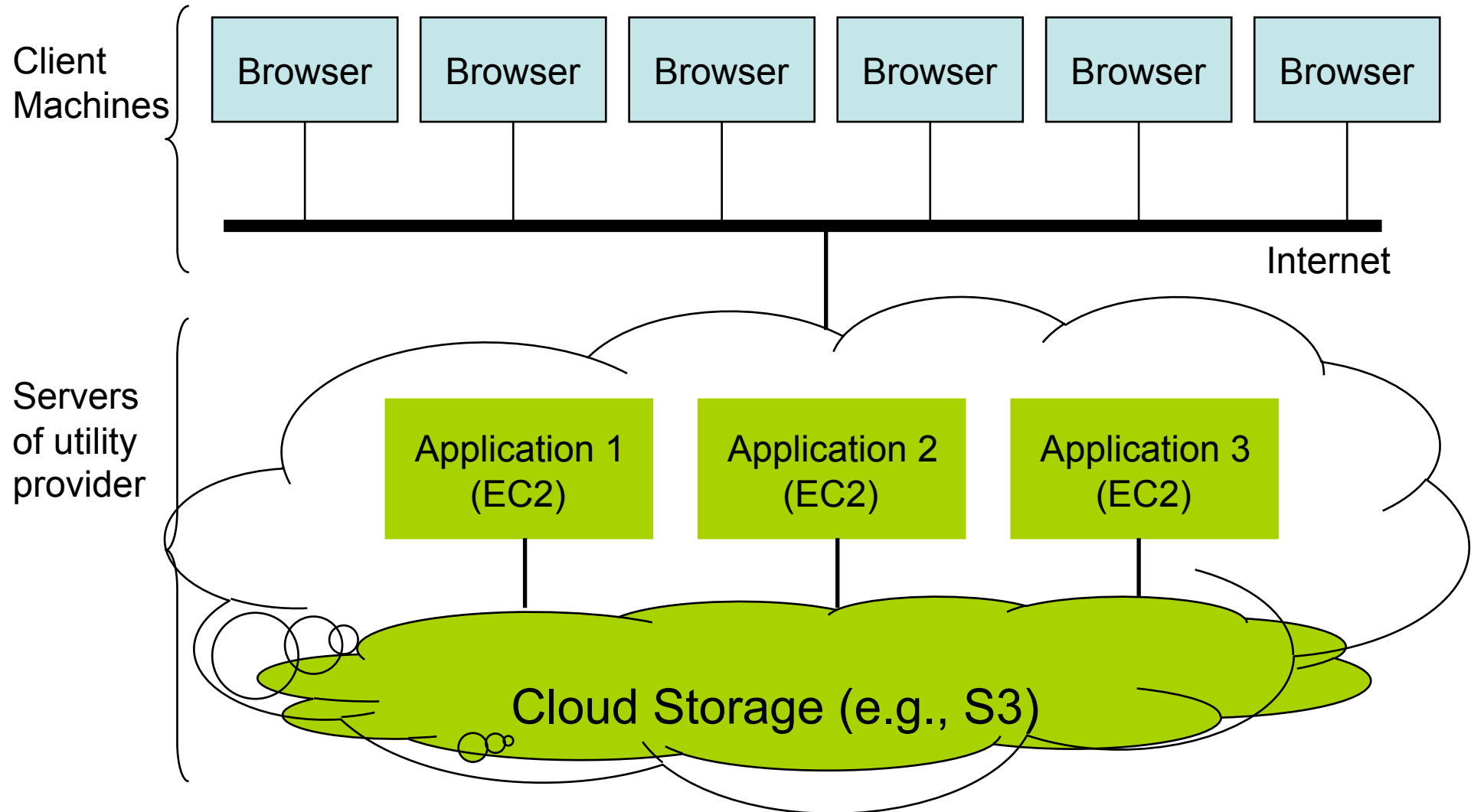


Step 1: Removing the DBMS

- S3 is like a huge disk; deals with many devils
 - infinite throughput
 - infinite capacity
 - never breaks; 100% read and write availability
 - no installation
- How about costs?
 - storage costs are okay
 - communication is expensive and slow (caching!)
 - still need to maintain servers for application
 - reduced consistency: eventual consistency

Need to go one step further

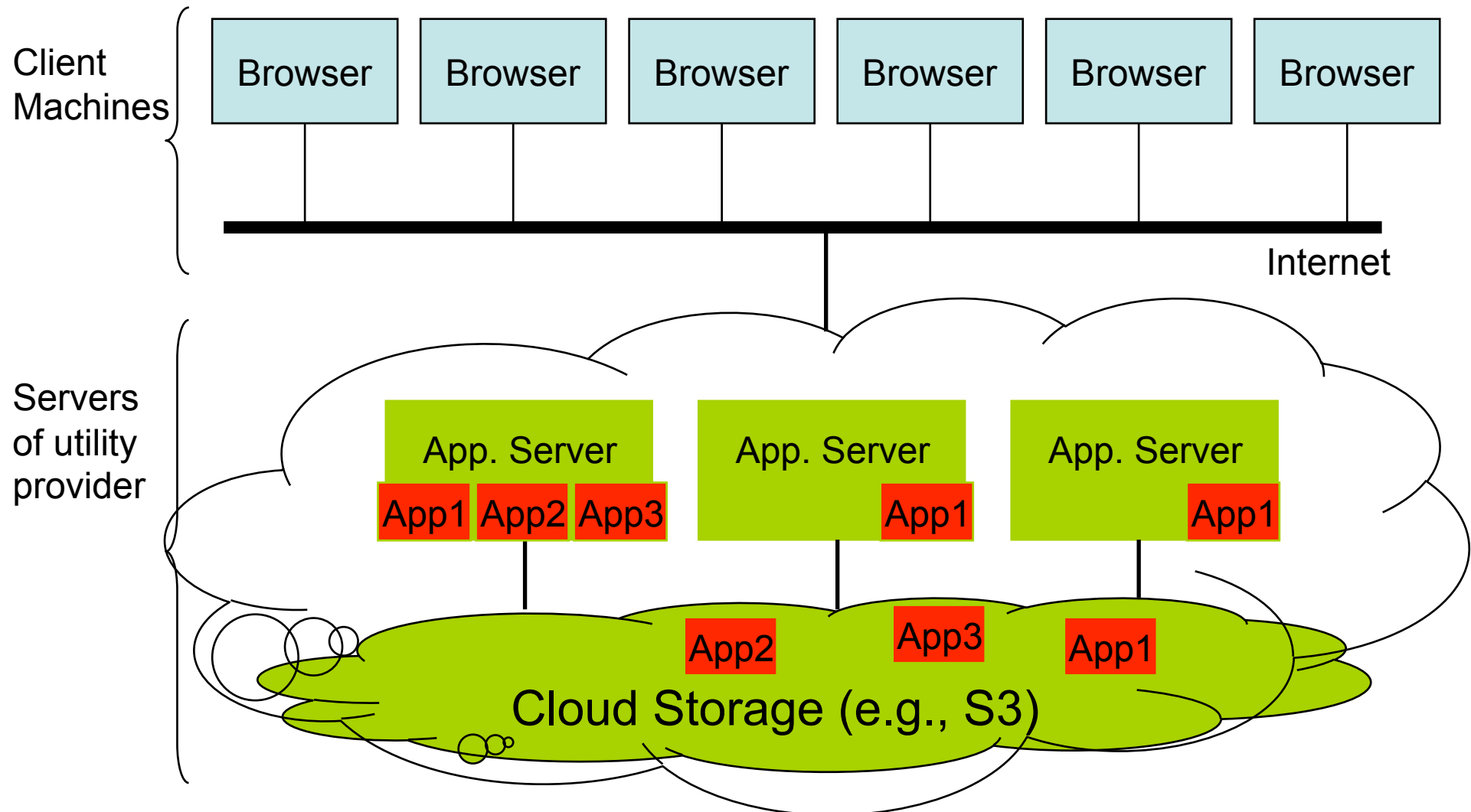
Step 2a: Move App into Cloud



Step 2a: Move App into the Cloud

- Solves many problems
 - no need to operate any servers
 - latency to cloud storage affordable
 - (fits nicely with traditional DBMS architecture, but causes the same problems there)
- How about cost?
 - CPU cycles in the cloud are expensive
 - need to administrate CPU cycle servers in cloud
 - too expensive to have one EC2 server per app.

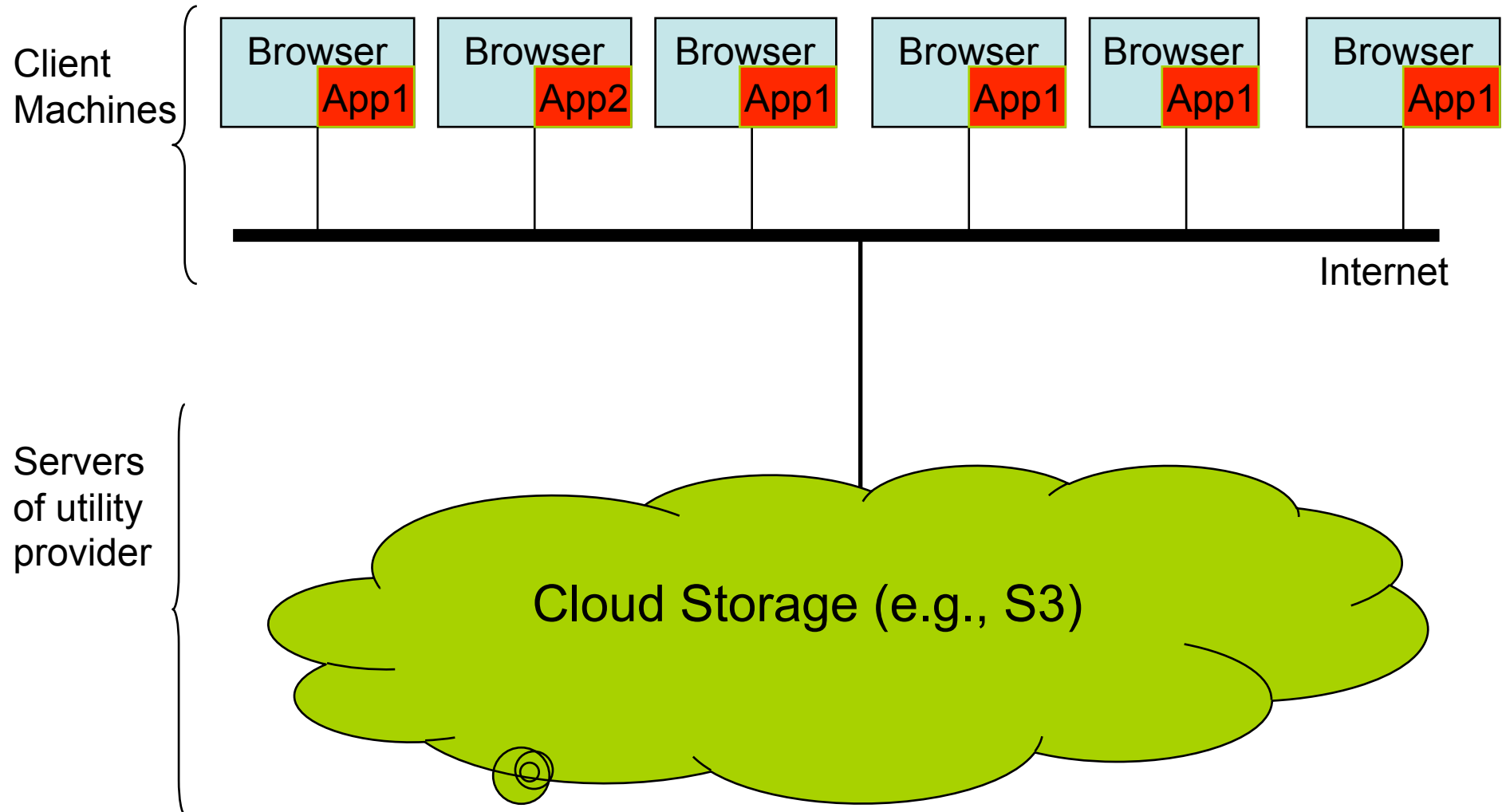
Step 2b: Move App into Cloud



Why not push DBMS into cloud?

- **current DBMS are too expensive**
 - consume too many CPU cycles for nothing
 - DBMS optimizer will fail miserably
 - make it difficult to share resources (multi-tenant)
- **current DBMS do not scale well**
 - too tightly coupled to single / set of machines
 - EC2 requires the exact opposite
 - (still need to define table spaces etc.)
- **What is needed instead?**
 - **a stateless virtual machine** (think of a JVM)
 - protocols to synchronize concurrent updates to S3

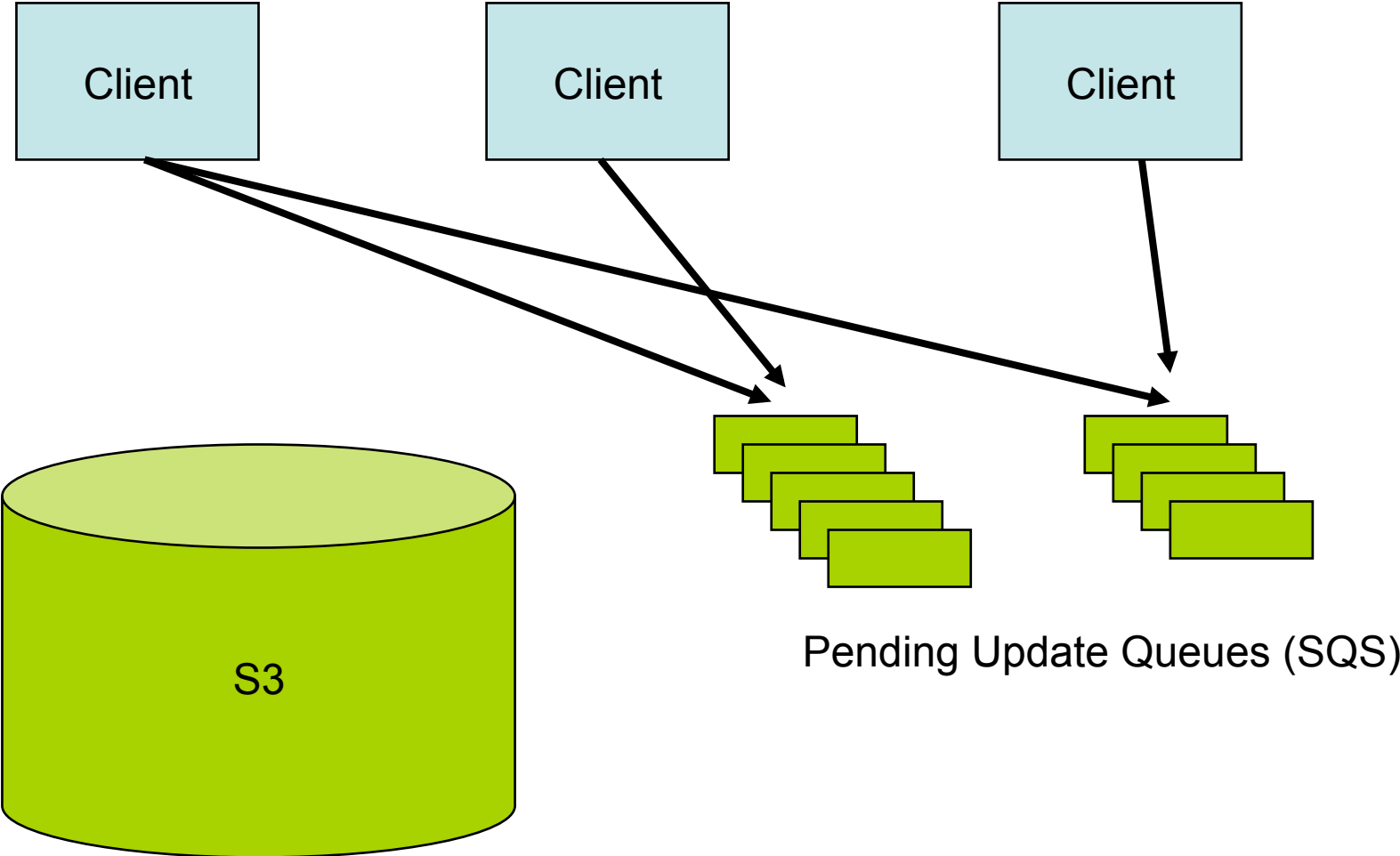
Step 3: Move App to Client



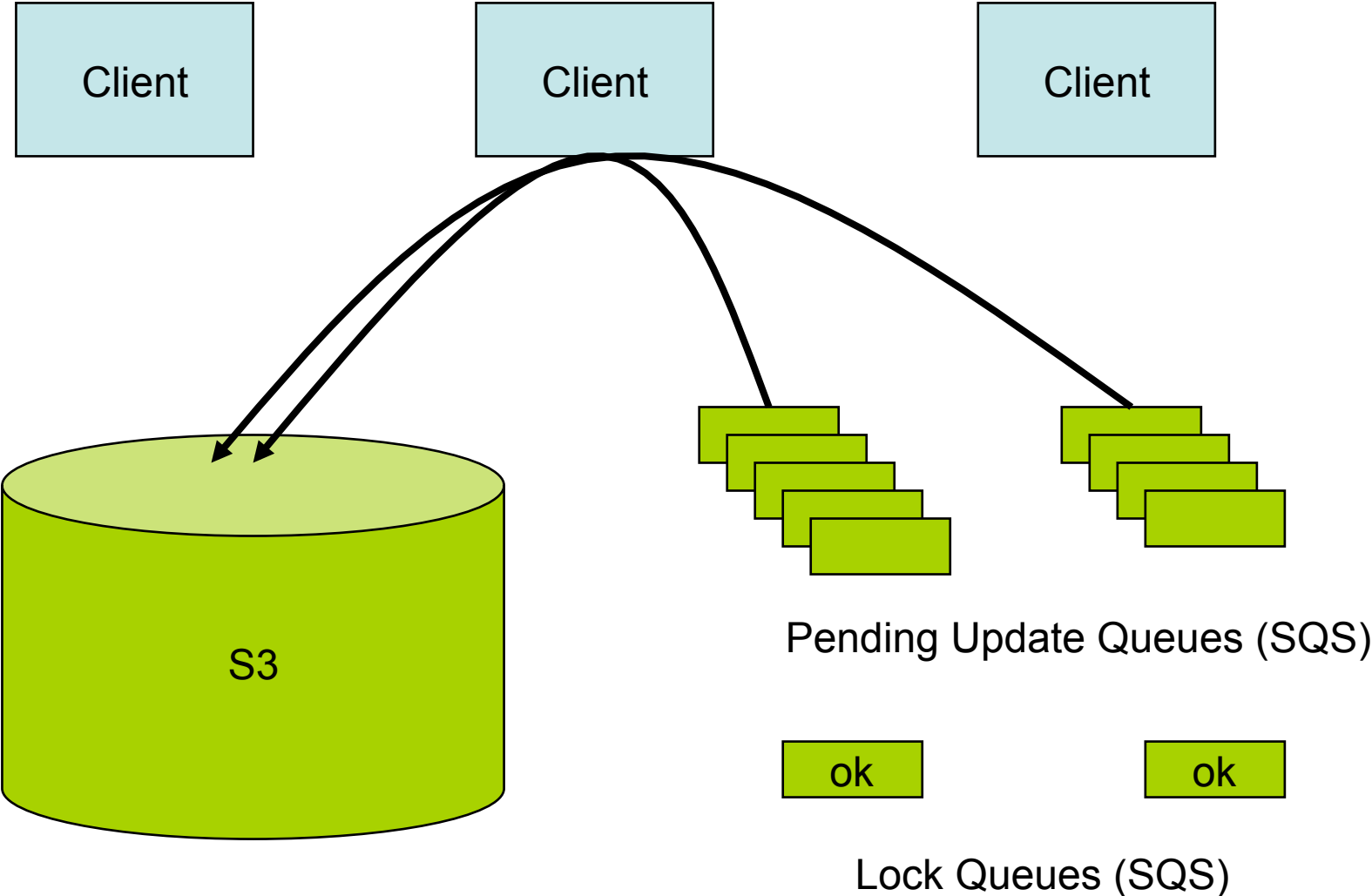
Step 3: Move App to the Client

- **Most cost effective solution**
 - fulfillment of client / server vision of the 90's
 - possibility to pass costs more directly to clients
- **„Interesting“ research** [Brantner et al. 2008]
 - synchronize concurrent updates (as in Step 2b)
 - interesting cost / consistency tradeoffs
 - interesting availability / consistency tradeoffs
 - client-side caching for performance / cost
 - how much infrastructure is needed?
 - e.g., for security, accounting, ...

Step 1: Clients commit update records to pending update queues



Step 2: Checkpointing propagates updates from SQS to S3



Agenda

- Why DBMS are evil?
- Reinventing Data Management on the Web
- **Why XQuery?**
- Putting it all together

Why XQuery?

- Prediction 1
 - XQuery is going to be the new Java
- Prediction 2
 - XQuery engine = JVM + DBMS
 - (XQuery engine is the new application server)

Why XQuery?

- Prediction 1
 - XQuery is going to be the new Java
- Prediction 2
 - XQuery engine = JVM + DBMS
 - (XQuery engine is the new application server)
- Disclaimer
 - in 1997, Sergey Brin asked me to invest a few \$100 into a company called „Google“ ...

Misconceptions about XQuery

- **X**

- XQuery is the only language for XML, but that does not mean that XML is all it can do
- CSV, JSON, HTML, ...
- spectrum: structured data to unstructured text

- **Query**

- XQuery has joins, group-by, sorting, etc., but that does not mean that it is only good for the DB
- by now, full-fledged programming language
 - XQuery 1.1 + XQuery Updates + XQuery Scripting
 - other deficiencies fixable in extensible engine
- modules for structured programming
 - Some proposals to go beyond (e.g, Unity @ ETH)

More XQuery Folklore

- XQuery is complicated
 - yes, if all you do is SPJ and you like SQL
 - but, ask your children!
 - is snowboarding more complicated than skiing?
- XQuery is slow
 - partially, true!
 - but, products catch up quickly
- XQuery is bad for tenure
 - maybe for SIGMOD
 - not true for EDBT :-)

Why XQuery?

- XQuery runs everywhere (all three tiers)
 - Browser [Fourny et al. 2008]
 - Server / middleware (BEA et al.)
 - Sensors, very small devices [Fischer et al. 2006, Müller et al. 2007]
 - (and in the database (Oracle et al.))
- XQuery(++) does the job
 - declarative processing
 - native support for REST (and Web Services)
 - semi-structured data with full text capabilities
 - great for streaming data (sequence data model)
 - extensions for ref/deref (graphs)
 - versioning, time travel
- XML is here to stay
 - data in different shapes, text and structured data

XQuery in the Middleware

- Content-based routing (Cisco)

```
//body/description[. ft:contains „Bad Things“]
```

- Message transformation

```
<order> <customer> { //kunde } </customer>  
... </order>
```

- XHTML generation

```
<html> <header> ... </header>  
    <body> ... </body>  
</html>
```

XQuery in the Client / Browser

- **RSS Aggregation:** merge RSS feeds on sports
for $\$m$ in $\$f1//item, \$f2//item, \dots, \$fN//item$
where $\$m$ `ft:contains(„Sport“)` with stemming
order by $\$m/date$
return $\$m$
- **AJAX:** replace search box with Google result
replace `//searchbox` with
`ws:searchGoogle(„//searchbox/searchstring“)`
- **Greasemonkey:** Heart on Web sites with „Lili“
if (`. ft:contains „Lili“`)
insert ``
into `//body`

XQuery in Sensor Networks

- Data Cleaning in Sensor

- runs on SwissQM on TinyMotes (4KB)

- average of last 5 values, disregard min, max

- ```
forseq $w in $sensor/value sliding window
```

- ```
start position $s when true
```

- ```
end position $e when $e - $s eq 4
```

- ```
return (sum($w) - min($w) - max($w)) div 3
```

- Aggregation at Gateway

- alarm if 5 sensors report a value greater 20

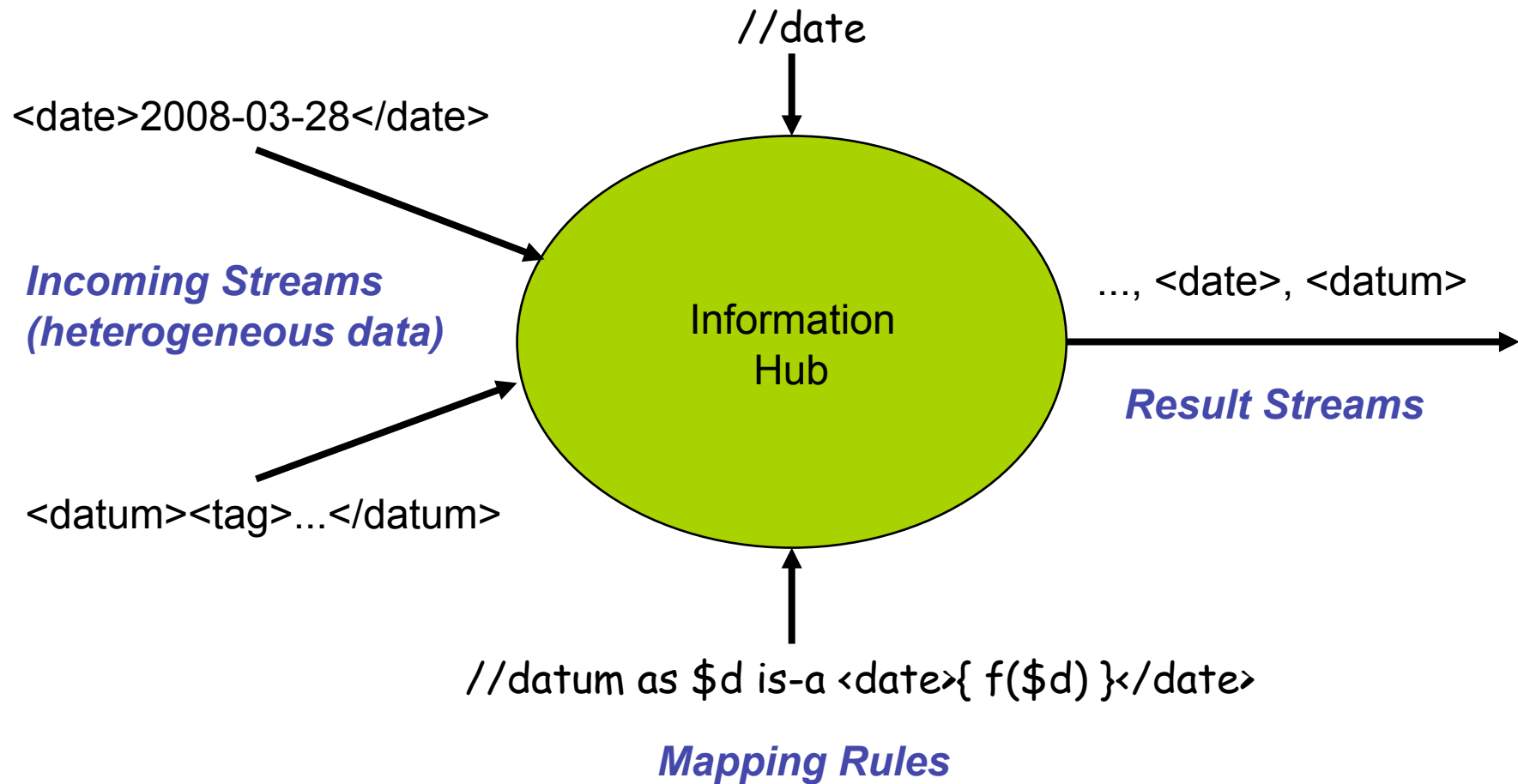
- ```
let $s := $sensors[value gt 20]
```

- ```
where count($s) gt 5
```

- ```
return <alarm/>
```

# Continuous Data Integration

*Subscriptions (continuous queries)*

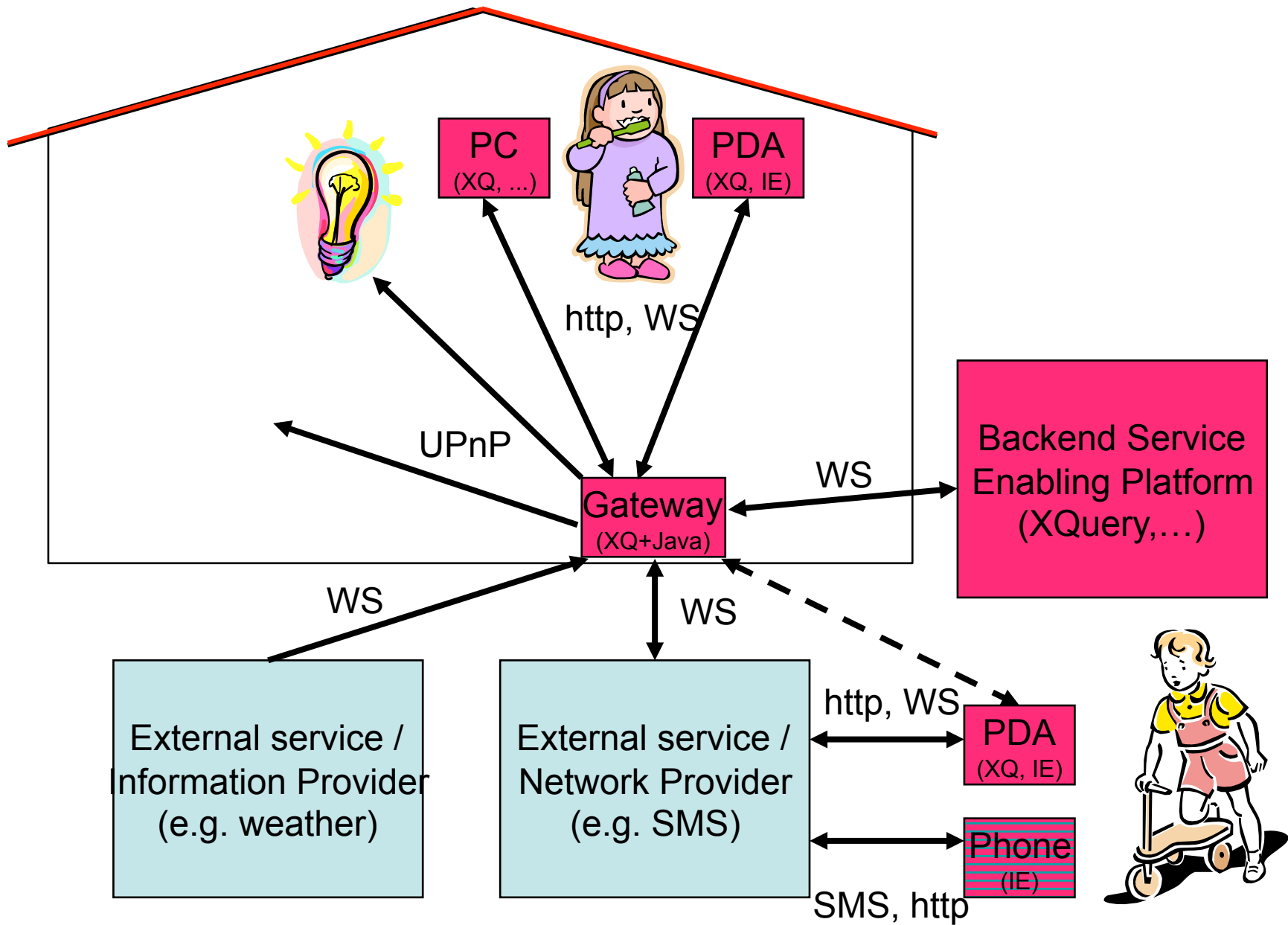


# Implementations

- Open Source Implementations (Apache 2.0 / BSD)
  - Zorba (C++): FLWOR Foundation - [www.flworfound.org](http://www.flworfound.org)
  - MXQuery (Java): ETHZ - [www.mxquery.org](http://www.mxquery.org)
  - many others...
- Commercial implementations
  - all major database vendors (IBM, Microsoft, Oracle)
  - all middleware vendors (BEA, DataDirect, ...)
- Supported platforms (MXQuery, Zorba)
  - TSky Motes, Intel 2 Sensors (SwissQM and xGate)
  - Internet Explorer Web browsers (June 2008)
  - PDA, Mobile Phone: CLDC 1.0 upwards
  - Playstation 3
  - Servers, Cluster of Machines

# SmartHome (Siemens)

- **Integration of home devices**
- **Orchestration of devices and their services**
  - UPnP based  
(XML descriptor, SOAP, SSDP, TCP/IP, etc.)
- **Service Composition, including integration of external web services**



# Other XQuery Projects (ETH)

- **Airline Alliances**
  - every student programs his/her own airline
  - form alliances
  - experiment: do this in Java/SQL first; then in XQuery
- **Public Transportation**
  - mobile phone computes best route (S-Bahn)
  - integrate calendar, address book, ZVV, GPS
- **Context-sensitive Remote Control**
  - mote captures „clicks“ and movements
  - mobile phone determines context and action (TV, garage, ..)
- **Lego Mindstorm**
  - move to warmest place in a room

# XQuery Projects (Stanford)

- **Celebrity Mashup**
  - photos, RSS feed, forums
- **Regional News Feeds**
  - icons with kind of news on Google Maps
- **Archiving and Crawling of Files**
- **PodCast Mashup**
- ...

# Agenda

- Why DBMS are evil?
- Reinventing Data Management on the Web
- Why XQuery?
- **Putting it all together**

# What is 28msec?

- A start-up that develops a platform to develop, deploy, and run Web applications
  - deployment is automatic; good support for other things
- Zorba XQuery engine
  - Open Source with Apache 2.0 License
- Development & deployment environment
  - Eclipse plug-in
  - Client-side running time environment (works off-line)
- S3 Storage Manager
  - Run XQuery scripts on S3 objects
- Web-based management console
  - Security and accounting (cost + revenue)
- XQuery Libraries (calendar, blogs, store, payment,...)

# What is an application?

- Developer's perspective (before deployment)
  - A set of XQuery++ modules (aka services)
  - A user interface
    - Calls the XQuery services via REST
    - Can be written in HTML+JavaScript, Flex, or XQuery
- Can be fully tested on the client (without S3)

# What is an application?

- Service provider perspective (after deploym.)
  - URI
  - Web-based management console
    - Cost, revenues, statistics, ...
  - all code and data stored in S3
    - can be retrieved for debugging and evolution
    - can be redeployed after changes
- Can be fully operated without additional infrastructure

# What is an application?

- Client perspective (i.e., end customer)
  - URI
    - call through the Web browser
    - or via REST for mashups
- Touch and feel like any other Web application

# Conclusion

- Cloud computing is the next wave
  - commoditization of basic computing
  - cost (\$) becomes predictable and measurable
  - the very big and the very small do it already  
the medium-sized will follow
- XQuery is the new Java
  - it is the right language for the Web
  - it is the right language for the Cloud (code mobility)
- Contact 28msec if you want to play with this
  - first (free) beta in the summer

<Thanks/>

**Green is the new blue**

<http://www.28msec.com>