PERVASIVE DATA MANAGEMENT

DATA MANAGEMENT IN WIRELESS SENSORS NETWORKS

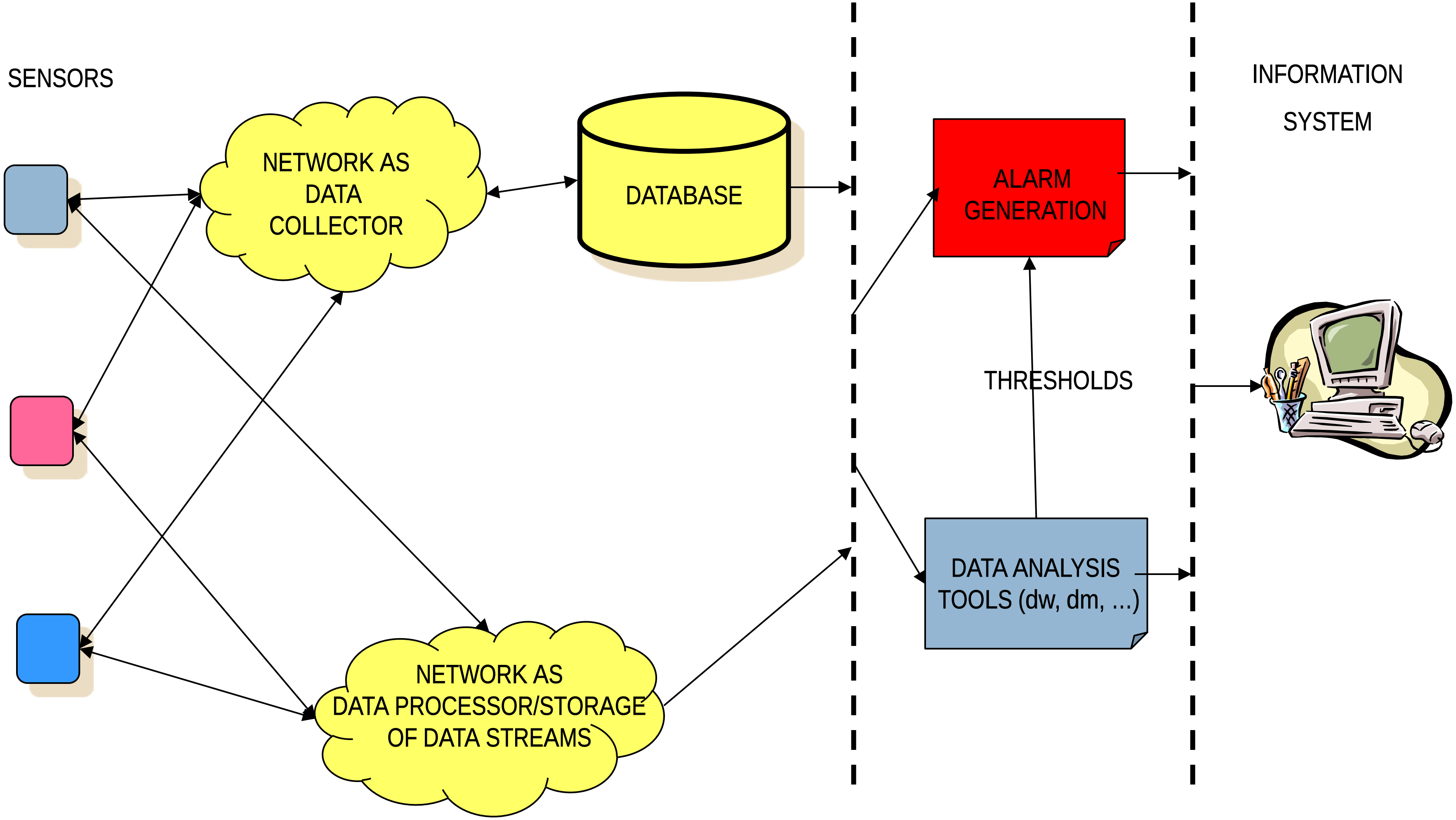
Prof. Fabio A. Schreiber

Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

DB VIEW OF SENSOR NETWORKS

- TRADITIONAL:
  - PROCEDURAL ADDRESSING OF INDIVIDUAL SENSOR NODES
    - **THE USER SPECIFIES HOW THE TASK IS EXECUTED, DATA IS PROCESSED CENTRALLY**
- DB-STYLE APPROACH:
  - DECLARATIVE QUERYING
    - **THE USER IS NOT CONCERNED ABOUT "HOW THE NETWORK WORKS" → IN-NETWORK DISTRIBUTED PROCESSING**

THE WSN NETWORK AS A DATABASE



WSN CONCEPTUALIZATION PROBLEMS

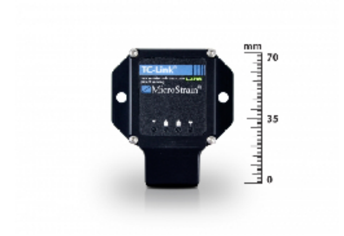
- HIDING THE DEVICES HETEROGENEITY TO THE APPLICATION USER
- USING A HIGH-LEVEL DECLARATIVE LANGUAGE TO SEND QUERIES AND COMMANDS TO THE DEVICES

A REAL LIFE SENSOR DATA SHEET

TC-Link®-1CH-LXRS™

Data Sheet

1 Channel Wireless Thermocouple Node



Introduction

The TC-Link® -1CH -LXRS® 1 Channel Wireless Thermocouple Node features a standard thermocouple input connector with an embedded cold junction temperature compensation sensor. On-board linearization algorithms are software programmable to support a wide range of thermocouple types including J, K, N, R, S, T, E and B. Its internal rechargeable battery allows remote, long term deployment.

Features & Benefits

High Performance

- Scalable, ultra-long-range wireless sensor network
- High-speed, synchronized platform accepts most analog sensors
- Reliable wireless data collection
- Low-power for extended battery life
- SensorCloud – integrated web solution

Ease of Integration

- Small, easy to integrate wireless form factor
- SDK for quick custom app development
- Rapidly deployed wireless solution

Cost Effective

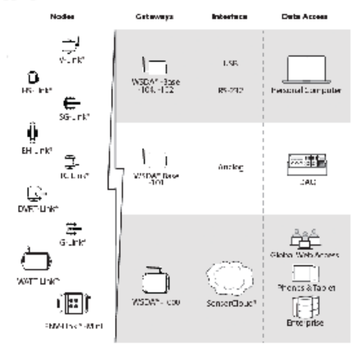
- Significantly reduced development cost
- Competitive OEM and volume discount schedule

Applications

- civil structure sensing, concrete maturation
- industrial sensing networks, machine thermal management
- food and transportation systems, refrigeration, freezer performance monitoring
- advanced manufacturing, plastic processing, composite cure monitoring
- cryogenic applications

System Overview

At the heart of MicroStrain's® LXRS™ Lossless Data Wireless Sensor Networks are WSDA® gateways, which use our exclusive beaconing protocols to synchronize precision timekeepers within each sensor node in the network. The WSDA® also coordinates data collection from all sensor nodes. Users can easily program each node on the scalable network for simultaneous, periodic, burst, or data logging mode sampling with our Node Commander® software, which automatically configures network radio communication to maximize the aggregate sample rate. Optional SensorCloud® enabled WSDA® support autonomous web-based data aggregation.



Wireless Sensor Network (WSN)

MicroStrain® Little Sensors, Big Ideas.® www.microstrain.com

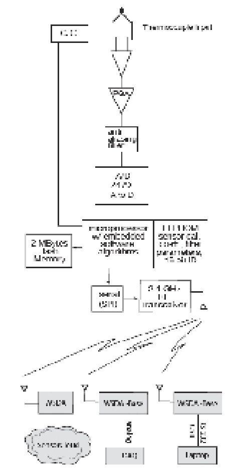
TC-Link®-1CH-LXRS™ 1 Channel Wireless Thermocouple Node

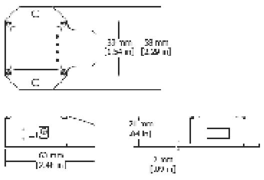
Functional attributes

Specifications:

Thermocouple inputs supported	software selectable: one, type-J, K, N, R, S, T, E, or B, input channel, one ambient CJC channel
Standard thermocouple measurement range	J: -210 to 760 °C; K : -200 to 1372 °C; N: -200 to 1300 °C; R: -50 to 1664 °C; S: -50 to 1664 °C; T: -200 to 400 °C; E: -200 to 1000 °C; B: 250 to 1820 °C
Temperature measurement accuracy	±0.1 % full scale or ±2 °C, whichever is greater (does not include errors due to TC wire or transducer)
Temperature repeatability	±0.1 °C (does not include errors due to TC wire or transducer)
Temperature resolution	0.0625 °C
Cold junction compensation range	-20 °C to 85 °C
Thermocouple connector	type 1 standard mini (SM) connectors for flat pin TC inputs
Analog to digital (A/D) converter	24 bit sigma-delta A/D
Sample Rate	programmable, from 2 Hz to 1 sample every 17 minutes, for datalogging or LDC modes
Datalogging mode	log up to 90,000 data points
Nodes per gateway	supports up to 100 nodes per gateway
Sample rate stability	datalogging and LDC modes ±25 ppm
Radio frequency (RF) transceiver carrier	2.4 GHz direct sequence spread spectrum, license free worldwide (2.405 to 2.480 GHz) – 16 channels, radiated power programmable from 0 dBm (1 mW) to 20 dBm (100 mW); European models limited to 10 mW
Range for bi-directional RF linktt	programmable communication range from 70m to 2,000m
RF data packet standard	IEEE 802.15.4, wireless communication architecture
PC Communications	115,200 baud over USB
Internal Li-ion battery	550 mAh, high capacity, Lithium-ion primary battery
Power consumption (battery life) with 550 mAh battery	2 samples per second - 0.8 mA (23 days) 1 sample per second - 0.48 mA (1.5 months) 3 samples per minute - 0.1 mA (6 months) 1 sample per minute - 0.09 mA (7 months, 13 days)
Operating temperature	-20 °C to +60 °C with standard internal battery and enclosure, extended temperature range optional with custom battery and enclosure; -40 °C to +85 °C for electronics only
Maximum acceleration Limit	500 g standard (high g option available)
Dimensions	58 mm x 63 mm x 21 mm (with enclosure)
Weight	48 grams (with enclosure and battery)
Enclosure Material	ABS plastic
Compatible Base Stations	WSDA®, WSDA® -Base (Analog), WSDA® -Base (USB/RS-232)
Software	Node Commander® Windows XP/Vista/7 compatible

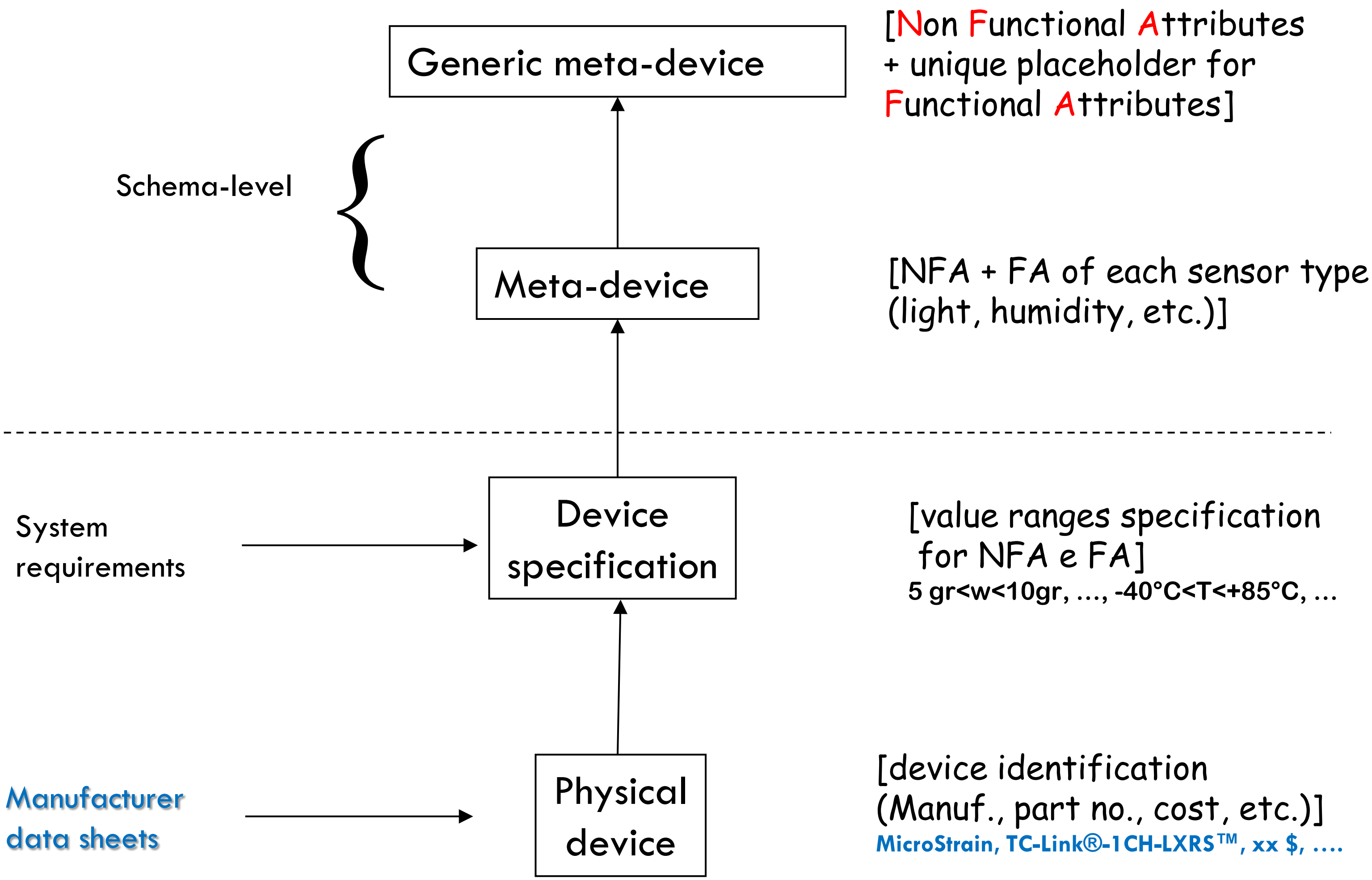
Non Functional Attributes





MicroStrain®
Microstrain Inc.
459 Hurricane Lane, Suite 102
Williston, VT 05495 USA
www.microstrain.com
ph: 800-449-3878
fax: 802-863-4093
sales@microstrain.com

ABSTRACTING PHYSICAL DEVICES



ABSTRACTING PHYSICAL DEVICES: AN XML DTD (1)

<!ELEMENT gmd: GenericMetaDevice (gmd: weight, gmd: size, gmd: MTBF, gmd: envcond, gmd: precision, gmd: metadevice) – non functional properties of every metadevice-- >

<!ELEMENT gmd: weight (#PCDATA)>

<!ELEMENT gmd: size (gmd: length, gmd: width, gmd: height)>

<!ELEMENT gmd: length (#PCDATA)>

<!ELEMENT gmd: width (#PCDATA)>

<!ELEMENT gmd: height (#PCDATA)>

<!ELEMENT gmd: MTBF (#PCDATA)>

<!ELEMENT gmd: precision (#PCDATA)>

………………………………………….

NON FUNCTIONAL ATTRIBUTES OF WHICHEVER DEVICE TO BE USED IN THE SYSTEM

<!ELEMENT gmd: envcond (gmd: tempmax, gmd: tempmin, gmd: relhummax, gmd: relhummin)>

<!ELEMENT gmd: tempmax (#PCDATA)>

<!ELEMENT gmd: tempmin (#PCDATA)>

<!ELEMENT gmd: relhummax (#PCDATA)>

<!ELEMENT gmd: relhummin (#PCDATA)>

…………………………………..

ENVIRONMENTAL ATTRIBUTES OF WHICHEVER DEVICE TO BE USED IN THE SYSTEM

ABSTRACTING PHYSICAL DEVICES: AN XML DTD (2)

<!ELEMENT gmd: metadevice (md: light | md: humidity | md: temperature | md: waterlevel) – **specific metadevices functional properties** -- >

<!ELEMENT md: light (md: lightmax, md: lightmin, md wavelenghtmax, md: wavelenghtmin>
<!ELEMENT md: lightmax (#PCDATA)>
<!ELEMENT md: lightmin (#PCDATA)>
<!ELEMENT md: wavelenghtmax (#PCDATA)>
<!ELEMENT md: wavelenghtmin (#PCDATA)>
<!ELEMENT md: humidity (md: relhummax, md: relhummin)>
<!ELEMENT md: relhummax (#PCDATA)>
<!ELEMENT md: relhummin (#PCDATA)>
<!ELEMENT md: temperature (md: tempmax, md: tempmin)>
<!ELEMENT md: tempmax (#PCDATA)>
<!ELEMENT md: tempmin (#PCDATA)>
<!ELEMENT md: waterlevel (md: watlevmax, md: watlevmin)>
<!ELEMENT md: watlevmax (#PCDATA)>
<!ELEMENT md: watlevmin (#PCDATA)>

FUNCTIONAL ATTRIBUTES OF A SPECIFIC DEVICE

HIGH LEVEL DM LANGUAGES FOR WSN

Tiny DB had the merit of being the first proposal of a declarative language for sensors data manipulation

- "TinyDB is a query processing system for extracting information from a network of TinyOS sensors."

Other approaches, e. g.

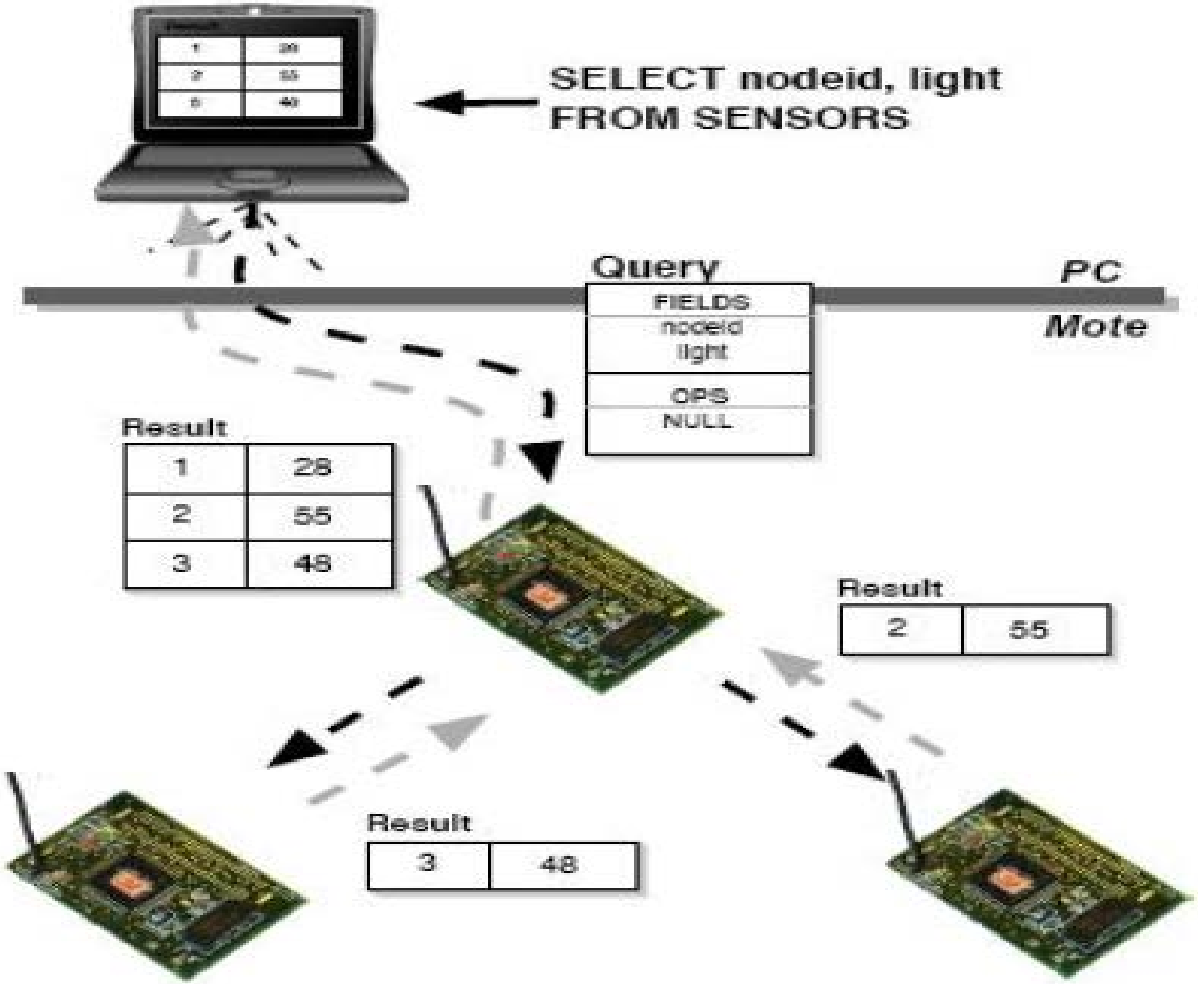
DSN – a completely declarative approach based on Snlog, a logic language

- Logic languages not easy for the systems practitioners

GSN middleware

- completely developed in Java and it is executed on the computers composing the backbone of the acquisition network.

QUERY / DATA FLOW



HOW DIFFERENT WSN QUERIES ARE FROM GENERAL QUERIES

SENSOR DATA

- TIME STAMPED
- SENSORS DELIVER DATA IN STREAMS
  - CONTINUOUS DATA PRODUCTION
  - OFTEN AT WELL DEFINED TIME INTERVALS
  - NO EXPLICIT REQUEST FOR THAT DATA.
- QUERIES NEED TO BE PROCESSED IN NEAR REAL-TIME
  - EXPENSIVE TO SAVE ALL DATA TO DISK
  - DATA STREAMS REPRESENT REAL-WORLD EVENTS WHICH NEED TO BE RESPONDED TO (e.g., traffic accidents and attempted network break-ins),
- NOT ALL SENSOR READINGS ARE OF INTEREST
- UNCERTAIN, INCOMPLETE INFORMATION

QUERY PROCESSING IN WSN

- WHEN SHOULD SAMPLES FOR A PARTICULAR QUERY BE TAKEN (acquisitional issue)
- WHICH SENSOR NODES HAVE DATA RELEVANT TO A PARTICULAR QUERY (indexing/optimization)
- IN WHAT ORDER SHOULD SAMPLES BE TAKEN AND HOW SHOULD IT BE INTERLEAVED WITH OTER OPERATIONS (indexing/optimization)
- IT IS WORTH EXPENDING COMPUTATIONAL POWER OR BANDWIDTH TO PROCESS AND RELAY A SAMPLE (stream processing/approximate answering)

QUERIES IN WSNs

QUERY TYPES

ONE-SHOT vs. LONG RUNNING

- (PARKING LOT) `SELECT nodeid,loc FROM sensors WHERE space=empty`
- (ENVIRONM. APPL.) `SELECT nodeid,temp FROM sensors EVERY 1 hour`

- EXHAUSTIVE (`SELECT *`) vs. AGGREGATE
  - SPATIAL AGGREGATION
  - TEMPORAL AGGREGATION
- TIME BASED vs. EVENT-BASED
  - `SELECT nodeid,temp FROM sensors EVERY 1 hour` (TIME BASED)
  - `ON EVENT temp>100 FROM sensors SELECT nodeid,temp FROM sensors EVERY 1 SEC` (EVENT BASED)
- ACCURATE vs. APPROXIMATE
  - `SELECT nodeid,temp FROM sensors ERROR 2 CONFIDENCE 95%`
- URGENT vs. DELAY TOLERANT

QUERY DISSEMINATION PHASE

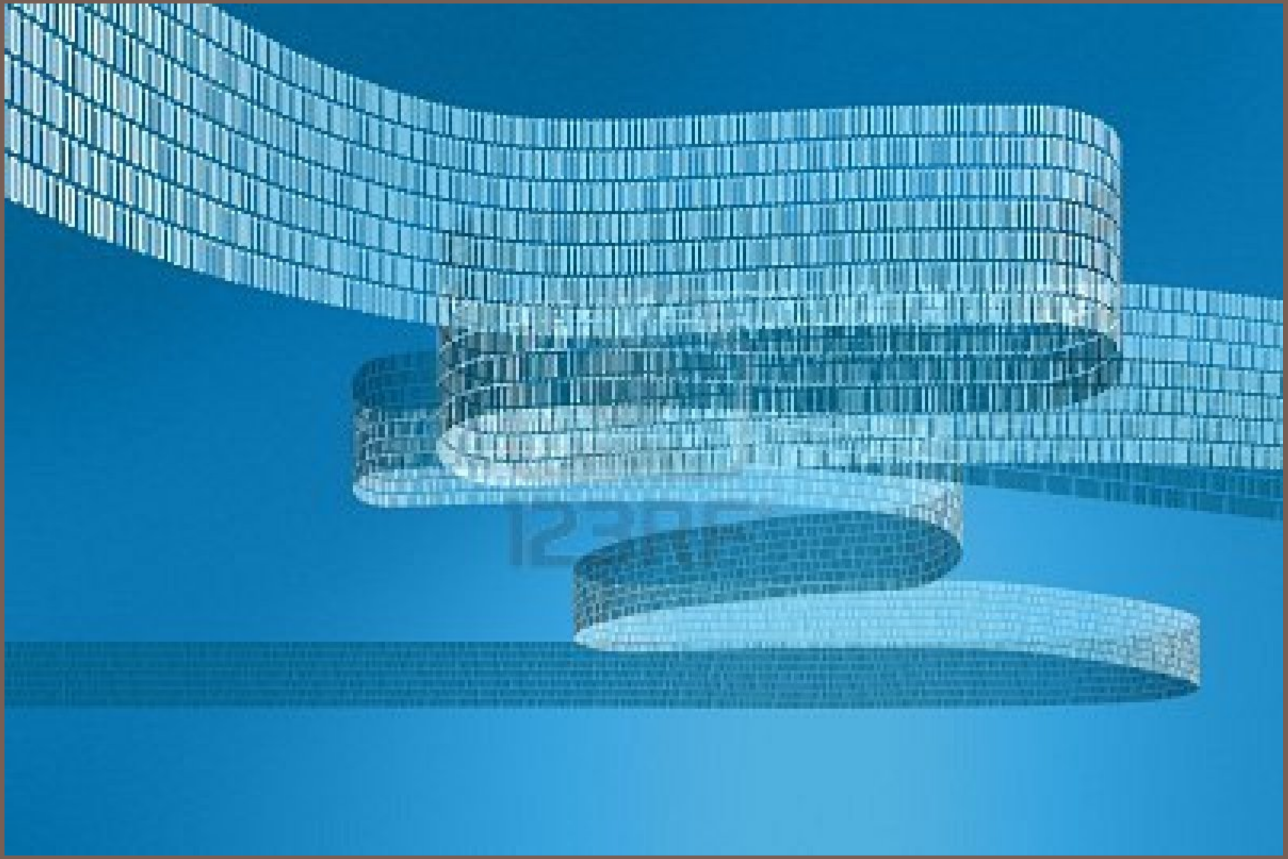
- **BROADCASTING** – QUERIES ARE DOWNLOADED TO **ALL** NODES
  - **SIMPLE FLOODING – EVERY NODE RETRANSMITS THE RECEIVED MESSAGE TO ALL ITS NEIGHBOURS, ONLY DISCARDING DUPLICATES**
  - **TREE FLOODING – BUILD A SPANNING TREE (ONCE), SEND ALONG THE TREE BRANCHES, DISCARD DUPLICATES**
  - **ENERGY EFFICIENT FLOODING – REQUIRES THE KNOWLEDGE OF TWO-HOP NEIGHBOURS**
- **SELECTIVE BROADCASTING** – ONLY **RELEVANT** NODES ARE AFFECTED
  - **SEMANTIC ROUTING TREES – EACH NODE IN THE TREE KNOWS THE ATTRIBUTES OF ITS CHILDREN AND THEIR CAPABILITY TO ANSWER A CERTAIN QUERY**
  - **GEOGRAPHIC ROUTING – EACH NODE HAS LOCATION INFORMATION AND QUERYES ARE SPATIALLY BOUNDED**
  - **CONTEXT/CONTENT BASED ROUTING – MESSAGES ARE ROUTED ON THE BASIS OF THEIR SEMANTICS AND NOT ON A PREDEFINED ADDRESS**

QUERIES IN WSNs
STORAGE / TRANSMISSION OPTIMIZATION

RESULTS COLLECTION PHASE – PROCESSING AND DISSEMINATION ARE DISTRIBUTED AMONG THE NODES ALLOWING CROSS-LAYER OPTIMIZATION

- IN-NETWORK PROCESSING
  - **AGGREGATION, COMPRESSION, APPROXIMATION**
  - **MODEL DRIVEN DATA ACQUISITION** – EXPLOIT TEMPORAL AND SPATIAL CORRELATIONS AMONG DATA
  - **PLACING OPERATORS IN THE NETWORK** – EXPLOIT THE DISTRIBUTIVITY OF OPERATORS TO PLACE THEM AS NEAR AS POSSIBLE TO THE SENSOR NODES
- INTERPLAY BETWEEN ROUTING AND QUERY PROCESSING
  - TRY TO SHAPE THE COMMUNICATION PATH IN ORDER TO OPTIMIZE CLASSES OF (SUB)QUERIES
- INTERPLAY BETWEEN MAC, ROUTING, AND QUERY PROCESSING
  - TDM-LIKE ACCESS PROTOCOLS TO AVOID (RE)TRANSMISSION COLLISIONS

DATA STREAM MANAGEMENT



http://www.123rf.com/photo_6855723

DATA STREAMS AND DSMS

- UNBOUNDED, RAPID, TIME-VARYING STREAMS OF DATA ELEMENTS, CONTINUOUS FLOWING ON THE WSN

- DATA STREAM MANAGEMENT SYSTEMS (DSMS) ARE DESIGNED TO PROCESS THEM CONTINUOUSLY, SINCE A STORE NOW AND PROCESS LATER APPROACH WILL NOT WORK DUE TO:
  - Response requirements: real time or quasi real-time
  - Streams are too massive, and also bursty.
  - Online filtering of interesting data for in-depth analysis later.

- MANY APPLICATIONS SIMILAR TO THOSE OF DBMS. MOST DSMS USE SOME FORM OF SQL

- COMPUTING ENVIRONMENTS QUITE DIFFERENT
  - PERSISTENT QUERIES ON TRANSIENT DATA, VS. TRANSIENT QUERIES ON PERSISTENT DATA
  - "A STREAM IS THE TIME DIFFERENTIAL OF A TABLE " (J. Hyde)

from C. Zaniolo – june 2009

DATA STREAMS AND DSMS APPLICATIONS

- COMPLEX EVENT PROCESSING (CEP): A CEP QUERY LOOKS FOR PATTERNS AMONG INDIVIDUAL EVENTS IN A SINGLE OR IN MULTIPLE STREAMS
  - TELECOM CALL RECORDS: FRAUD DETECTION
  - FINANCIAL APPLICATIONS: E.G. ALGORITHMIC TRADING
  - SECURITY APPLICATIONS: INTRUSION DETECTION, HOMELAND SECURITY
- MONITORING QUERIES: AGGREGATE LARGE NUMBERS OF RECORDS AND LOOK FOR TRENDS
  - NETWORK MONITORING AND TRAFFIC MANAGEMENT
  - WEB LOGS AND CLICK-STREAMS
- PUBLISH/SUBSCRIBE SERVICES
- SENSOR NETWORKS, RFID TAGS
- Extract, Transform, Load (ETL) TOOL FOR DATA WAREHOUSES IMPROVES CURRENCY w.r.t. THE BATCH TOOL

DBMS VS. DSMS

	DBMS	DSMS
Model	**persistent** data	**transient** data
Table	**set|bag** of tuples	**Infinite sequence** of tuples
Updates	**all**	**append only**
Query	**transient**	**persistent**
Query Answer	**exact**	often **approximate**
Query Eval	**multi-pass**	**one-pass**
Operator	**blocking OK**	**unblocking only**
Query Plan	**fixed**	**adaptive**
Data processing	**synchronous**	**asynchronous**
Concurrency control overhead	**high**	**low**

 Adapted from C. Zaniolo – june 2009

RELATIONAL DATA STREAMS

- EACH DATA STREAM CONSISTS OF RELATIONAL TUPLES
- THE STREAM CAN BE MODELLED AS AN APPEND-ONLY RELATION, BUT REPETITIONS ARE ALLOWED AND ORDER IS VERY IMPORTANT!
- ORDER BASED ON TIMESTAMPS OR ARRIVAL ORDER
  - EXTERNAL TIMESTAMPS
    - **INJECTED BY DATA SOURCE**
    - **MODEL REAL-WORLD EVENT REPRESENTED BY TUPLE**
    - **TUPLES MAY BE OUT-OF-ORDER--BUT IF NEAR-ORDERED THE DSMS CAN REORDER THEM USING SMALL BUFFERS**
  - INTERNAL TIMESTAMPS
    - **INTRODUCED AS SPECIAL FIELD BY THE DSMS**
    - **ACCORDING TO THE APPROXIMATE TIME OF ARRIVAL**
  - MISSING
    - **THE SYSTEM ASSIGNS NO TIMESTAMP TO ARRIVING TUPLES, BUT TUPLES ARE STILL PROCESSED AS ORDERED SEQUENCES BY OPERATORS WHOSE SEMANTICS EXPECTS TIMESTAMPS…**
    - **THUS OPERATORS MIGHT INSTANTIATE TIMESTAMPS AS/WHEN NEEDED**

QUERIES OVER DATA STREAMS

- BLOCKING / NON BLOCKING OPERATORS
  - MUST / NEED NOT SEE EVERYTHING IN THE INPUT BEFORE THEY CAN RETURN ANYTHING IN THE OUTPUT

- RELATIONAL OPERATORS
  - **`select`, `project`, `join`, AND `union` ARE MONOTONIC NB OPERATORS**
  - **SQL-2 `aggregates` ARE BLOCKING (FOR ARBITRARILY ORDERED INPUT)**

  - ON DATA STREAMS WE ARE ONLY INTERESTED IN MONONOTONIC FUNCTIONS
  - LET O' BE THE SUBSET OF OPERATORS OF A DB LANGUAGE $\mathcal{L}$ THAT ARE MONOTONIC. $\mathcal{L}$ WILL BE SAID TO BE NB-COMPLETE IF ALL OF ITS FUNCTIONS CAN BE EXPRESSED USING ONLY THE OPERATORS IN O'

BASIC DEFINITIONS 1

- **TIME DOMAIN** $\mathcal{T}$: a discrete, linearly ordered, countably infinite set of time instants $t \in \mathcal{T}$
- **STREAM** $\mathcal{S}$: a countably infinite set of elements $s \in \mathcal{S}$
- **STREAM ELEMENT**: s:<v, t^{app}, t^{sys}, bid>
  - v: Relational tuple conforming to a schema S
  - $t^{app} \in \mathcal{T}$: application time value (partially ordered)
  - $t^{sys} \in \mathcal{T}$: system time value (totally ordered)
  - $bid \in \mathcal{N}$: batch-id value

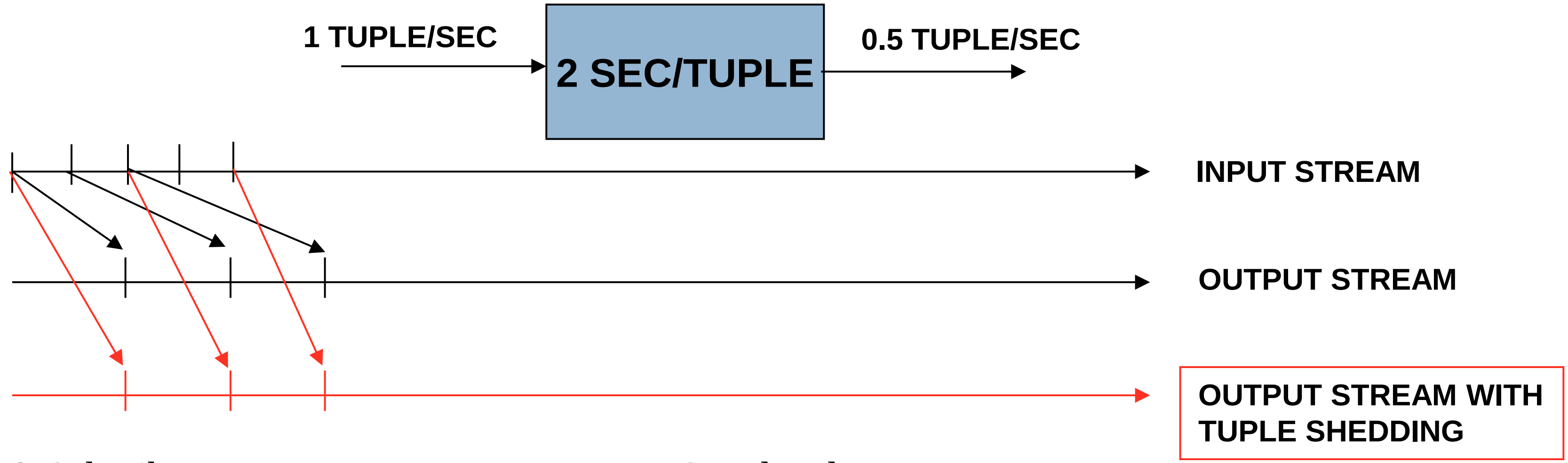
BASIC DEFINITIONS 2

- **BATCH B**: finite subset of S where all $b \in S$ have an identical t^{app}
- **TIME BASED WINDOW** : $W = (o,c]$ over S is a finite subset of data elements s where $o < s.t^{app} \leq c$
- **WINDOW PARAMETERS**:
  - SIZE ω : $\forall\ W = (o,c] \in \mathcal{W}\ ;\ c - o = \omega$
  - SLIDE β : distance between two consecutive windows $W_1, W_2\ ;\ o_2 - o_1 = \beta$

QUERIES OVER DATA STREAMS

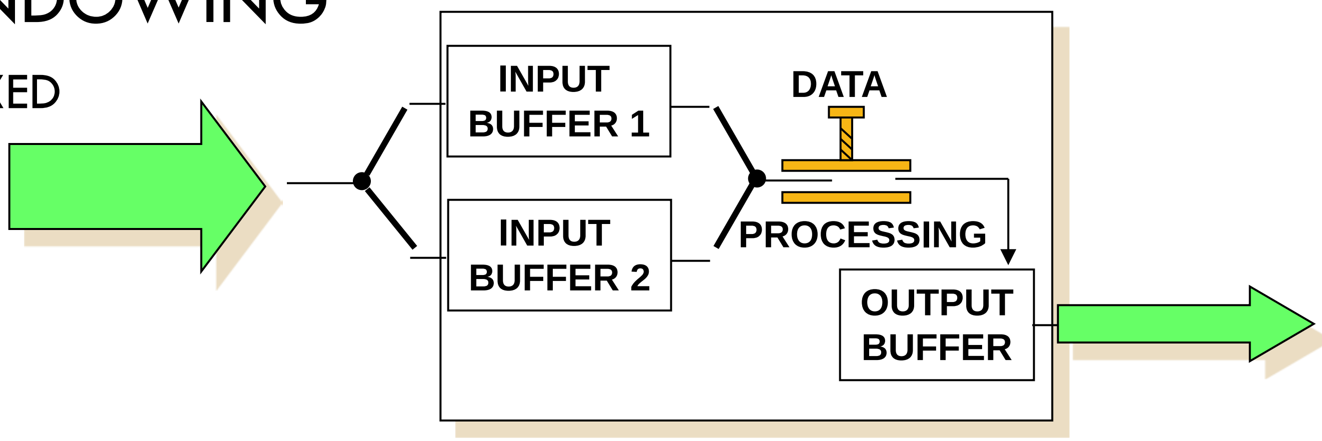
APPROXIMATE QUERIES

- TIMELINESS OFTEN REQUIRES LOW-LATENCY ANSWERS
- OVERLOADS CAN BE DEALT WITH BY **LOAD SHEDDING**

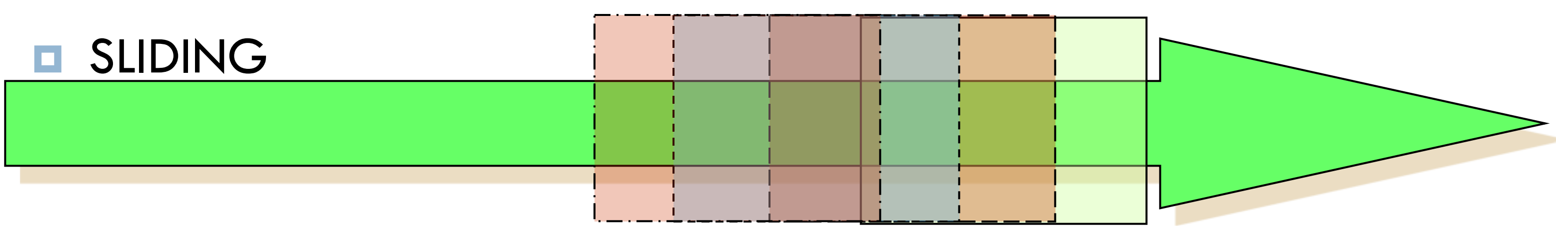


QUERIES OVER DATA STREAMS

- WINDOWING
  - FIXED



  - SLIDING



SYSTEMS

HIGH LEVEL DM LANGUAGES FOR WSN

Tiny DB had the merit of being the first proposal of a declarative language for sensors data manipulation

- "TinyDB is a query processing system for extracting information from a network of TinyOS sensors."

Other approaches, e. g.

DSN – a completely declarative approach based on Snlog, a logic language

- Logic languages not easy for the systems practitioners

GSN middleware

- completely developed in Java and it is executed on the computers composing the backbone of the acquisition network.

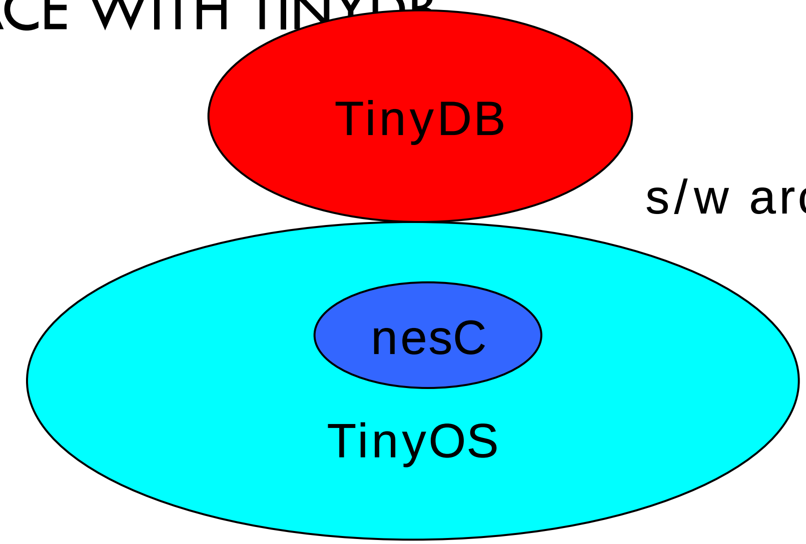
TinyDB: ACQUISITIONAL QUERY PROCESSOR

“TinyDB is a query processing system for extracting information from a network of TinyOS sensors.”

- DISTRIBUTED QUERY PROCESSOR RUNNING ON EACH OF THE NODES OF THE SENSOR NETWORK
- REDUCED SQL INTERFACE (WITH SOME ADDITIONAL CONSTRUCTS)
- QUERIES ISSUED FROM A PC
- COLLECTS DATA FROM MOTES IN THE ENVIRONMENT, FILTERS IT, AGGREGATES IT TOGETHER, AND ROUTES IT OUT TO A PC
- EXPLOITS POWER-EFFICIENT IN-NETWORK PROCESSING ALGORITHMS.
- MULTIPLE PERSISTENT QUERIES WITH DIFFERENT SAMPLING TIME

TinyDB – TinyOS

nesC USED TO PROVIDE HARDWARE ABSTRACTION & INTERFACE WITH TINYDB



s/w architecture of a mote

TinyDB → 20000 LINES OF C CODE → 58kB

TinyDB

- NETWORK TOPOLOGY
  - NEIGHBORS TRACKING
  - ROUTING TABLES MAINTENANCE
  - ENSURES THAT EVERY MOTE IN THE NETWORK CAN EFFICIENTLY AND (RELATIVELY) RELIABLY DELIVER ITS DATA TO THE USER.
- INCREMENTAL DEPLOYMENT VIA QUERY SHARING: TO EXPAND A TinyDB SENSOR NETWORK, YOU SIMPLY DOWNLOAD THE STANDARD TinyDB CODE TO NEW MOTES, AND TinyDB DOES THE REST. TinyDB MOTES SHARE QUERIES WITH EACH OTHER. NO PROGRAMMING OR CONFIGURATION OF THE NEW MOTES IS REQUIRED BEYOND INSTALLING TinyDB.
- OFFERS LOGGING CAPABILITIES: LOGS ON POSTGRESQL ON THE BASE STATION
- *MATCHBOX FILE SYSTEM*

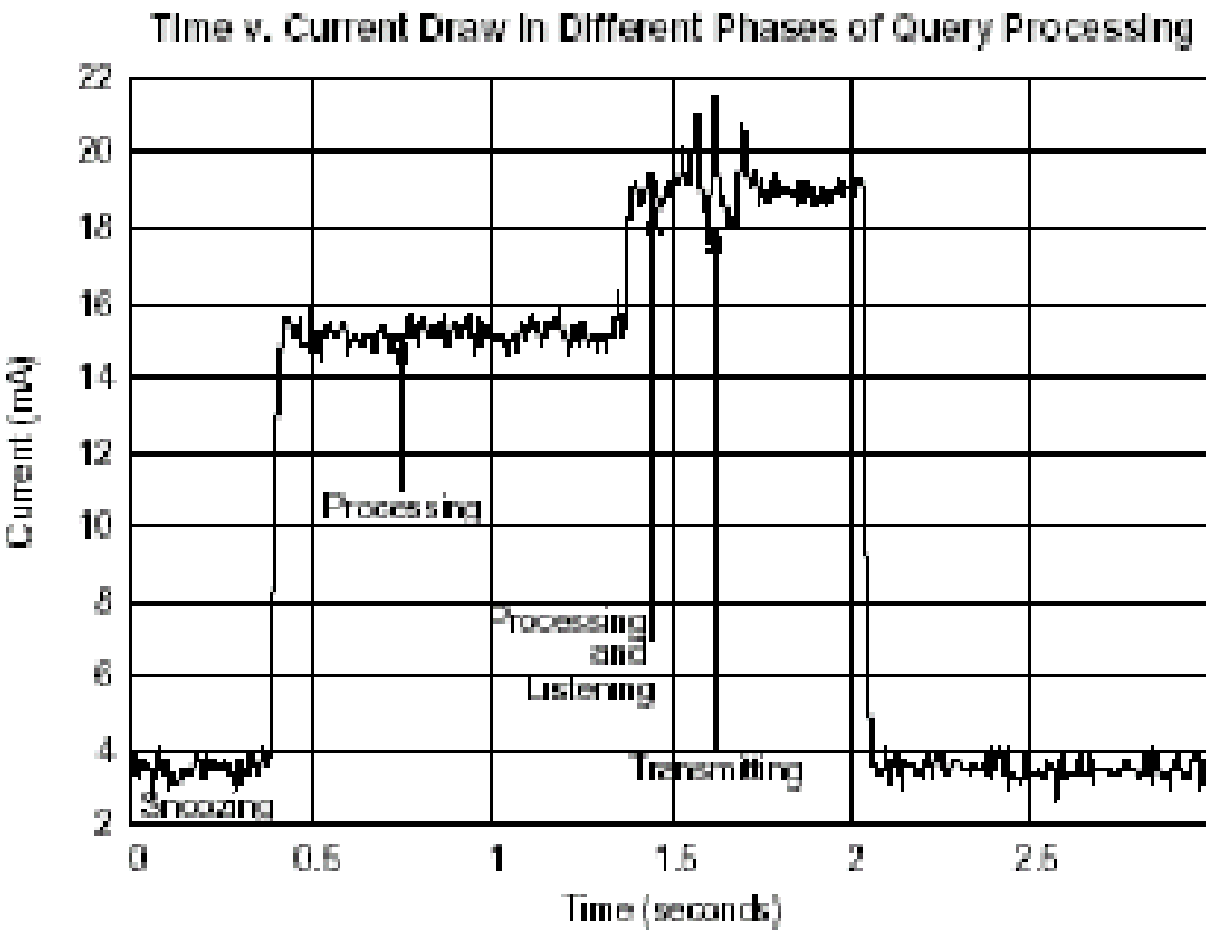
TinyDB ARCHITECTURE

- SENSOR NETWORK SOFTWARE (runs on each mote)
  - SENSOR CATALOG AND SCHEMA MANAGER
  - QUERY PROCESSOR
  - MEMORY MANAGER
  - NETWORK TOPOLOGY MANAGER
- CLIENT INTERFACE (Java classes running on the base station)
  - BUILD AND TRANSMIT QUERIES
  - RECEIVE AND PARSE QUERY RESULTS
  - EXTRACT INFORMATION ABOUT DEVICES ATTRIBUTES AND CAPABILITIES
  - GUI's TO
    - CONSTRUCT QUERIES
    - DISPLAY INDIVIDUAL SENSOR RESULTS
    - VISUALIZE DYNAMIC NETWORK TOPOLOGIES
- APPLICATION SOFTWARE

TinyDB DATA STRUCTURE

- SENSORS TABLE IS AN UNBOUNDED, CONTINUOUS DATA STREAM
  - SCHEMA COLUMNS ARE DIFFERENT PHYSICAL DATA
    - SAME FOR EVERY SENSOR → POSSIBLE NULL VALUES
  - ROWS ARE INDIVIDUAL SENSOR DATA
- PHYSICALLY, THE SENSORS TABLE IS PARTITIONED ACROSS ALL OF THE DEVICES IN THE NETWORK, WITH EACH DEVICE PRODUCING AND STORING ITS OWN READINGS
- COMPARING READINGS FROM DIFFERENT SENSORS REQUIRES THE COLLECTION OF DATA ON A SAME NODE (POSSIBLY THE ROOT)
- OPERATIONS SUCH AS SORT AND SYMMETRIC JOIN ARE NOT ALLOWED ON STREAMS. THEY ARE ALLOWED ON BOUNDED SUBSETS (WINDOWS) OF THE STREAM STORED AS MATERIALIZATION POINTS

POWER CONSUMPTION PROFILE IN TINYDB



TinyDB SQL

IMPLEMENTS ACTIVE CAPABILITIES

```
SELECT roomno, AVG(light), AVG(volume)
FROM sensors
GROUP BY roomno
HAVING AVG(light) > l AND AVG(volume) > v
SAMPLE PERIOD 30s FOR 5min
```

The query runs Delivering information Every 30 seconds for 5 minutes

```
ON evtTest:
SELECT light
FROM sensors
WHERE light>threshold
TRIGGER ACTION SetSnd(512)
```

Fire the query on a defined Event

Activate an action, here a beeper sound

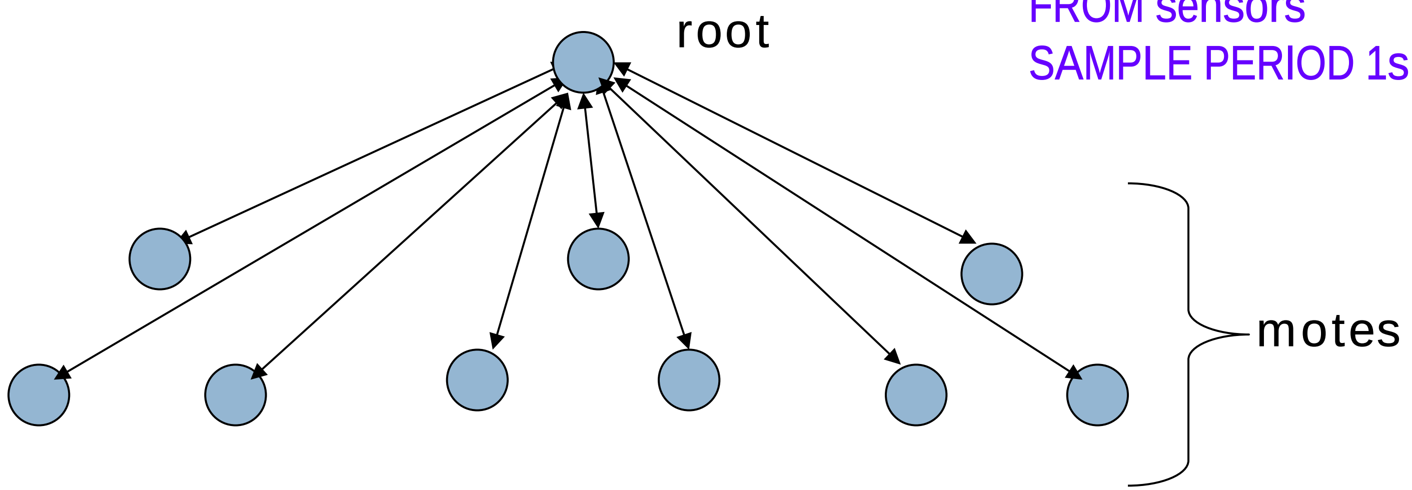
SOME TYPES OF QUERIES

- BROADCAST QUERY
- LANDMARK QUERY
- AGGREGATION QUERY
- SLIDING WINDOW QUERY
- EVENT BASED QUERY
- LIFE TIME BASED QUERY
- …

TYPES OF QUERIES - TinyDB

BROADCAST QUERY : The query from the Root is broadcasted to all the clients in the network, irrespective of the architecture, with a definite sampling period.

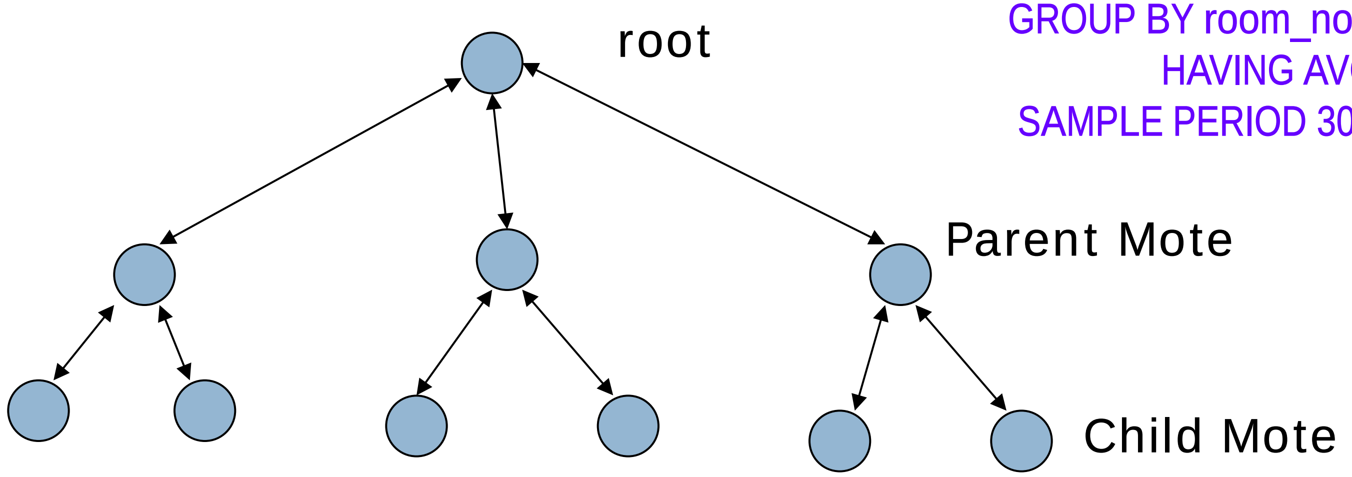
SELECT ALL [light, temp]
FROM sensors
SAMPLE PERIOD 1s FOR 10s



TYPES OF QUERIES - TinyDB

AGGREGATION QUERY :The query from the Root is transmitted to all parent motes and parent motes transmit query to child motes with a definite sampling Period.

```
SELECT AVG (volume), room_no
   FROM sensors
            WHERE floor = 6
   GROUP BY room_no
            HAVING AVG (volume) >   threshold
    SAMPLE PERIOD 30s
```



TYPES OF QUERIES - TinyDB

SLIDING WINDOW QUERY (Temporal Aggregate) :

This is similar to Aggregate Query with sampling interval being divided into windows.

```
SELECT WINAVG (volume, 30s, 5s)
    FROM sensors
    SAMPLE PERIOD 1s
```

TYPES OF QUERIES - TinyDB

EVENT BASED QUERY: This is an Asynchronous Query mechanism in which events are used as a triggering medium for initiating data collection.

ON EVENT bird – detect (loc) :

SELECT AVG (light), AVG (temp), event.loc

FROM sensors AS s

WHERE dist (s.loc, event.loc) < 10m

SAMPLE PERIOD 2s FOR 30s

TYPES OF QUERIES - TinyDB

LIFE TIME BASED QUERY : This query mechanism runs for definite lifetime and the rate of sampling will be as quick as possible in order to satisfy the goal of data collection

```
SELECT node_id, accel
FROM sensors
LIFETIME 30 days
```

OTHER TYPES OF QUERIES - TinyDB

- MONITORING QUERIES
- NETWORK HEALTH QUERIES
- EXPLORATORY QUERIES
- NESTED QUERIES (only on materialized points)
- ACTUATION QUERIES
- OFF-LINE DELIVERY

TinySQL LIMITATIONS

- `FROM` clause must always list exactly one table `sensors` (unless from flash-logged query)
- No support for `NOT` and `OR` boolean operators in `WHERE` and `HAVING` clauses
- No nested query support
- Renaming not supported (AS)
- Reduced Arithmetic expressions (only `column` *op* `constant`)

TinySQL EXTENSIONS

- `TRIGGER ACTION` clause (optional) specifies an action to be performed after query execution on every mote.
- `EPOCH DURATION <int>` clause is used to specify time between the repetitions of a query (the query is fired every `<int>` milliseconds)
- `On <event>:` is used to have a query firing in response to some (hw) event, for example the crossing of a threshold of a sensor or packet arrival on the radio.
- `INTO` clause is used to log a query into an on-mote flash memory (you can drop it with `DROP ALL`)

BUT FURTHER USEFUL DATABASE FUNCTIONALITIES ARE STILL LACKING…

- One VSDB should reside at least on every generic sensing device (e.g. Mica2)
- To compose a distributed/federated database
- Each VSDB should be context aware
- Each VSDB should be able to “appropriately” redirect queries to neighbours (P2P)
  - because of an internal fault or a generic unavailability
  - because it does not possess the information
  - because the other node “knows” something more, in order to complete the information
  - because the other node has a less power-consuming sensor on-board
  - design appropriate, optimized query processing plans (e.g. redirect subquery, cache subquery result, etc.)

DATA REDUCTION

DATA AGGREGATION

- REDUCES, BY LOCAL PROCESSING, THE AMOUNT OF DATA TO BE TRANSMITTED
- SAVES TRANSMISSION POWER

AGGREGATION OPERATIONS

- NATIVE
  - SUM, COUNT, AVG
  - MIN, MAX (singleton sensor values)
- USER DEFINED
  - TAG (Tiny AGgregation), CAG (Clustered AGgregation),
  - TiNA (Temporal coherency-aware in-Network Aggregation), q-digest (quantile digest)

COMPRESSION OPERATIONS

LTC (Lightweight Temporal Compression), Wavelets, …

DATA TRANSMISSION

GOAL: SAVE ENERGY BY INTELLIGENT ROUTING

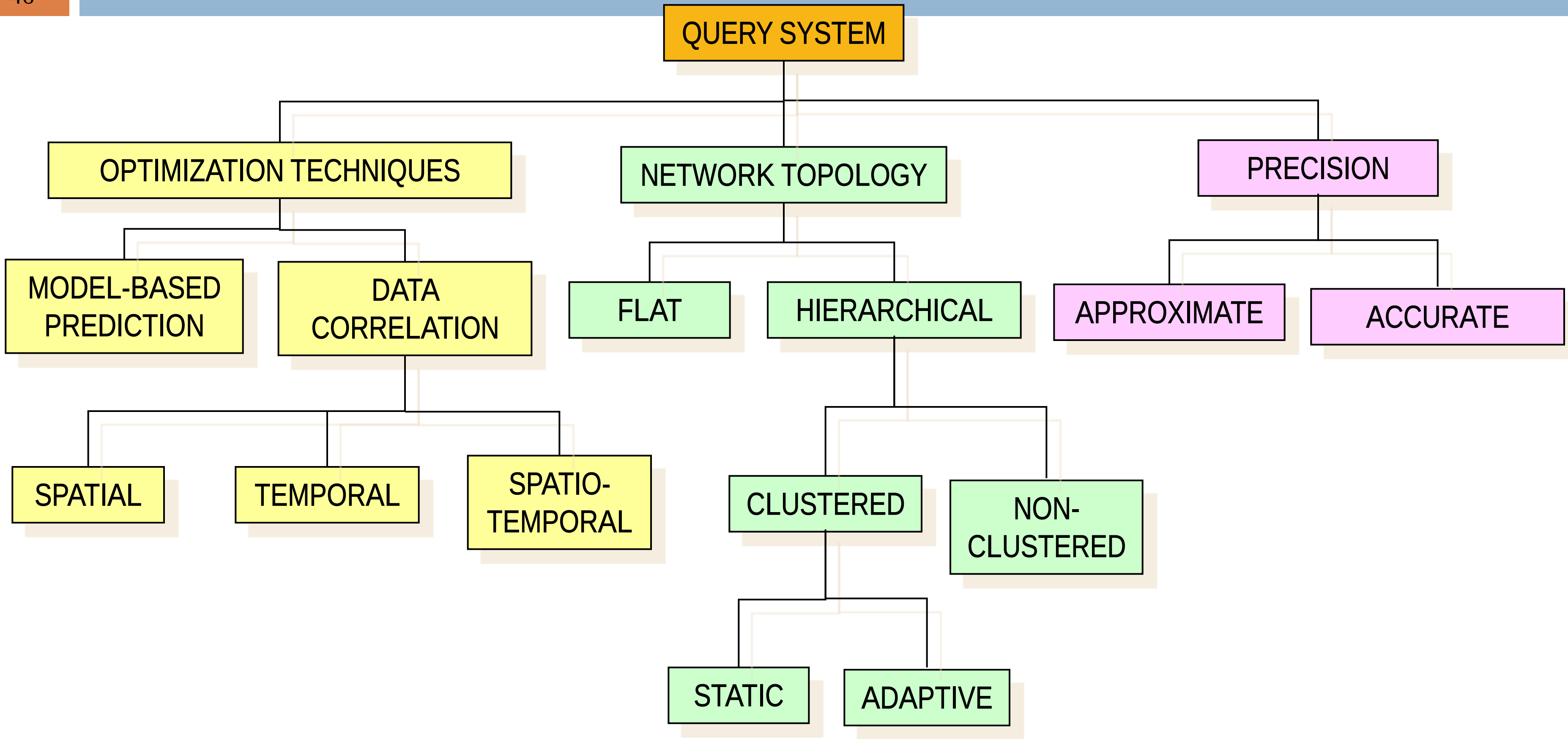
- ADDRESS CENTRIC
  - SHORTEST PATH
  - MULTI-HOP PROTOCOL

SCALE BADLY

- DATA CENTRIC
  - DATA FLOW FROM SOURCES TO SINKS INDEPENDENTLY OF NODE ADDRESSES
  - DIRECTED DIFFUSION, CLUSTERING, …

SCALE WELL

QUERY SYSTEMS FOR WSN's



DSN – A COMPLETELY DECLARATIVE APPROACH

- SNLOG, A DIALECT OF DATALOG, IS USED FOR THE COMPLETE SYSTEM STACK (applications → single-hop communication)
  - **VARIABLES, CONSTANTS, PREDICATES, FACTS, RULES, UNIFICATION**
- DISTRIBUTED EXECUTION
  - **LOCATION SPECIFIER @ ADDED TO PREDICATES**
- LIBRARY OR USER SPECIFIED BUILT-IN PREDICATES TO LINK THE DECLARATIVE PROGRAM TO THE SENSORS' HARDWARE
- USER CONTROLLED MEMORY USAGE
- PROGRAMMER DEFINED PREDICATES PRIORITIES WHEN FIRING RULES

RFID DATA WAREHOUSING

- RFID TUPLE BASIC STRUCTURE

  EPC, location, time
- ENORMOUS AMOUNT OF LOW LEVEL DATA
  - TIME REDUNDANCY FOR STATIONARY ITEMS (*stays*)

    EPC, location, time-in, time-out
  - SPACE REDUNDANCY ITEMS STAY TOGETHER FOR LONG TIME INTERVALS (e.g. inside a truck)

    EPC-list, location, time-in, time-out
- GROUP TRANSITIONS AND NOT STAYS
  - DIFFICULTIES IN QUERY ANSWERING
    - **CAN REQUIRE EXTENSIVE SEARCH AND JOINS**
- USE GENERALIZED IDENTIFIERS (*gids*) FOR GROUPING ITEMS WHICH MOVE TOGETHER

RFID DATA WAREHOUSING

NEED OF AGGREGATING DATA WHILE PRESERVING ITS PATH-LIKE STRUCTURE

RFID-CUBOID

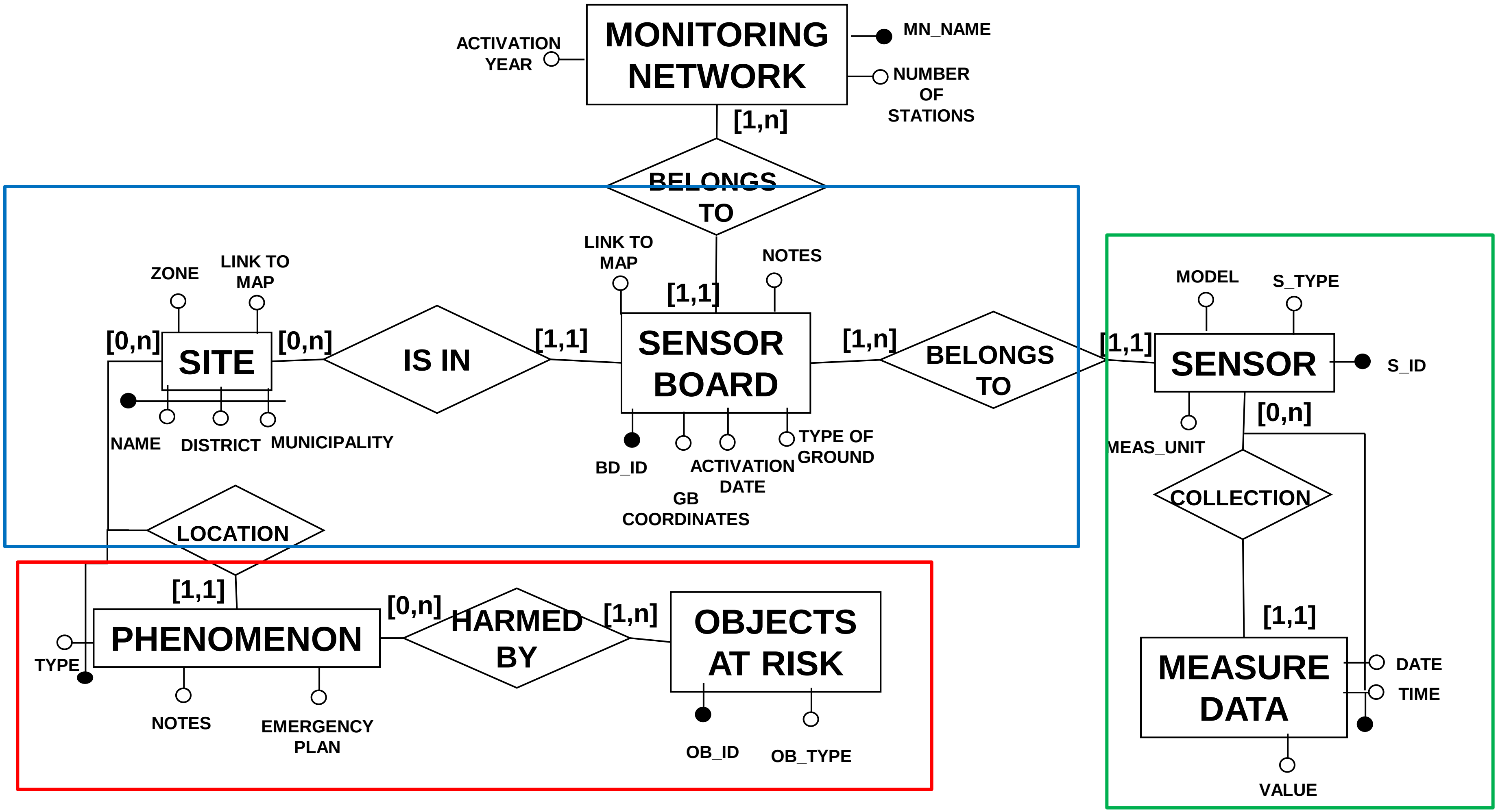
- FACT TABLE *stay* OF CLEANSED RFID RECORDS
- INFORMATION TABLE *info* STORES PATH- INDEPENDENT INFORMATION
- MAP TABLE LINKING THE RECORDS IN THE FACT TABLE WHICH FORM A PATH

THE MAP TABLE (NOT PRESENT IN DWs) PRESERVES THE PATH STRUCTURE OF DATA

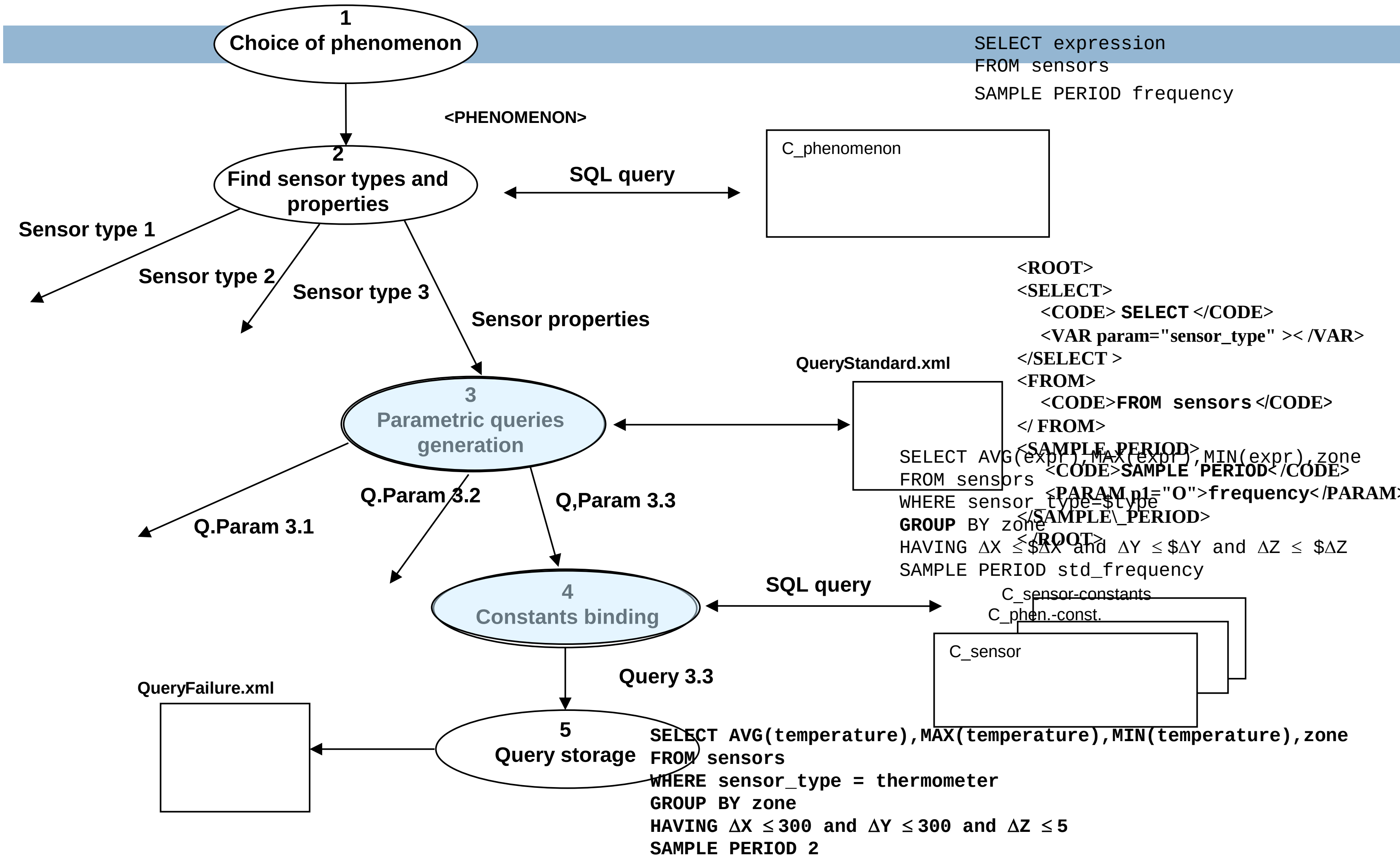
→ ABSTRACTION LEVEL

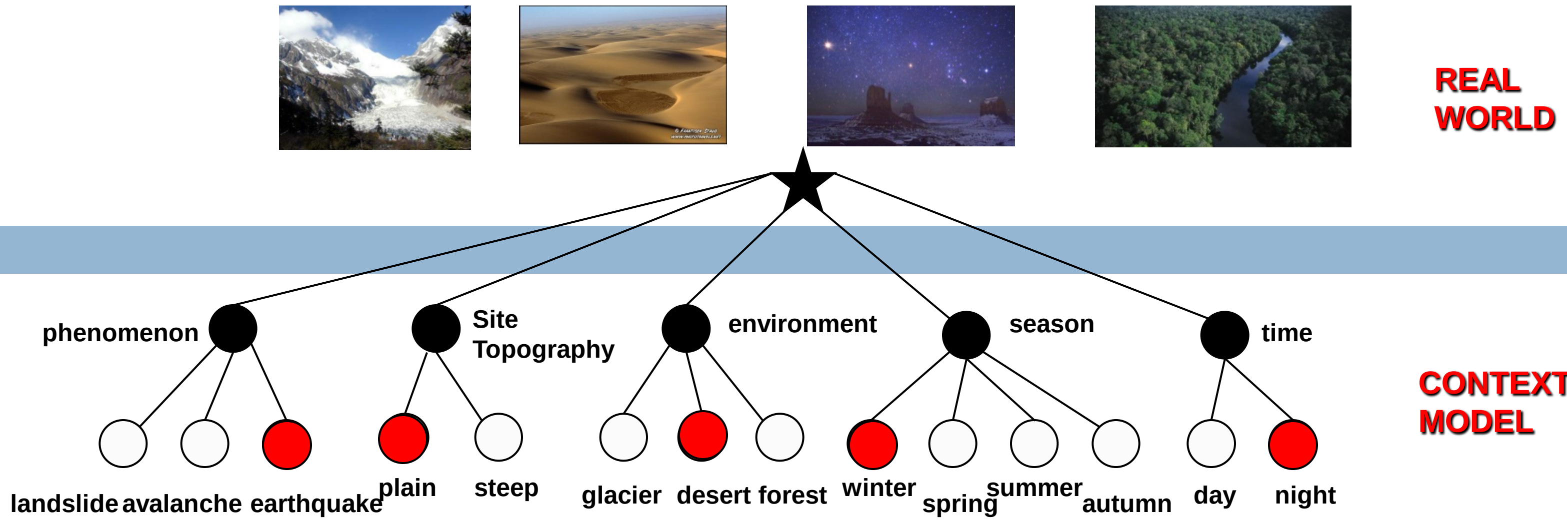
raw-RFID → cleansed RFID → RFID cuboids → popular cuboids

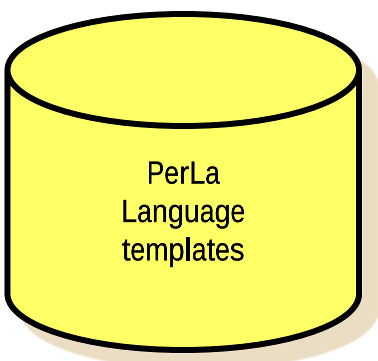
CONCEPTUAL SCHEMA OF A GEOPHYSIC MONITORING SYSTEM



PARAMETRIC GENERATION OF SENSOR QUERIES





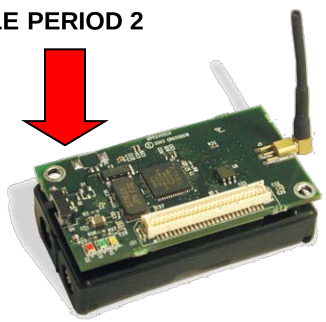



```
<ROOT>
<SELECT>
   <CODE> SELECT </CODE>
   <VAR param="sensor_type" >< /VAR>
</SELECT >
<FROM>
   <CODE>FROM sensors </CODE>
</ FROM>
<SAMPLE_PERIOD>
    <CODE>SAMPLE PERIOD</CODE>
    <PARAM p1="O">frequency</PARAM>
</SAMPLE\_PERIOD>
< /ROOT>
```

QUERY TAILORING

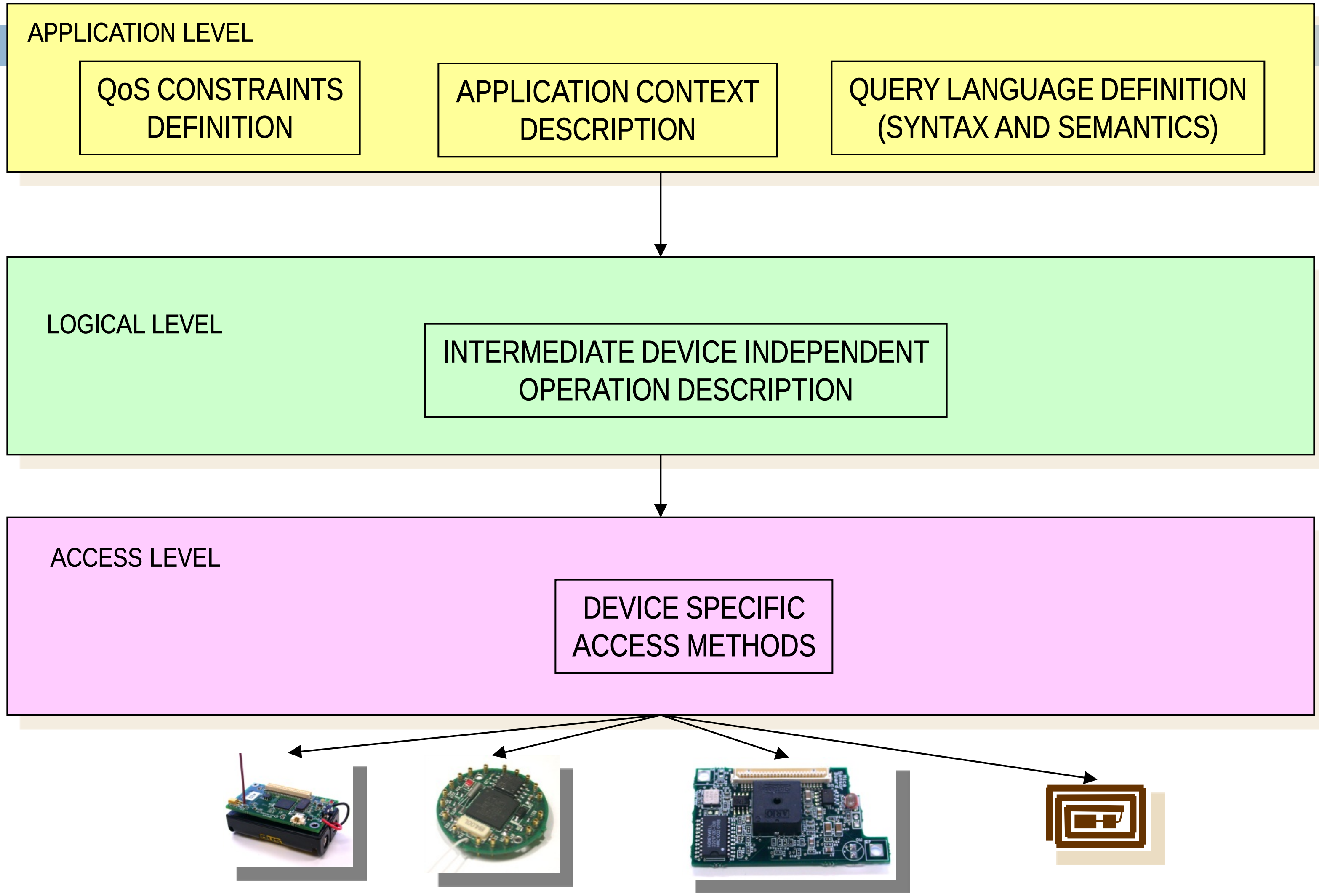
```
SELECT
AVG(accel),MAX(accel),MIN(accel),zone
FROM sensors
WHERE sensor_type= accelerometer
GROUP BY zone
HAVING DX < 300 and DY < 300 and DZ <
5
SAMPLE PERIOD 2
```

DOWNLOAD



QUERY STEREOTYPE

A WSN PORTABLE DATA LANGUAGE



HOMEWORK DEEPENING TOPICS 1

- DATABASES IN CONTROL SYSTEMS
- WSN ARCHITECTURES FOR INFORMATION PROCESSING AND MANAGEMENT
- QUERY PROCESSING IN WSNs
- QUERY OPTIMIZATION IN WSNs

HOMEWORK DEEPENING TOPICS 2

- DSMS ARCHITECTURES AND PROTOTYPE SYSTEMS
- ALGEBRAIC OPERATORS DEFINITION AND IMPLEMENTATION IN DSMS
- QUERY OPTIMIZATION IN DSMS
- JOINING DATA STREAMS

BIBLIOGRAPHY

- D.C. Chu et.Al. – The Design and Implementation of a Declarative Sensor Networ System – Technical Report UCBEECS-2006-132, www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-132.html
- H. Gonzales, J. Han, X. Li, D. Klabjan – Warehousing and Analyzing Massive RFID Data Sets – Proc. 22nd ICDE, (2006)
- Hyde J.- Data in Flight – Communications ACM, Vol. 53, n. 1, (2010)
- S.R. Madden, M.J. Franklin, J.M. Hellerstein, W. Hong – TinyDB: An Acquisitional Query Processing System for Sensor Networks – ACM-TODS, Vol. 30, n. 1, pp. 122-173, (2005)
- L. Mottola, G.P. Picco – Programming Wireless Sensors Networks: Fundamental Concepts and State of the Art – ACM Computing Surveys, Vol. 43, n. 3, Article 19, (2011)
- F.A. Schreiber – Automatic Generation of Sensor Queries in a WSN for Environmental Monitoring – 4th Intl. ISCRAM Conf. Delft, (2007)
- S. Yoon, C. Shahabi - An Experimental Study of the Effectiveness of Clustered AGgregation (CAG) Leveraging Spatial and Temporal Correlations in Wireless Sensor Networks - ACM Transactions on Sensor Networks, Vol. 3, n. 1, (2007)

BIBLIOGRAPHY LINKS

- http://www.sics.se/contiki/
- http://mantis.cs.colorado.edu/tikiwiki/tiki-index.php
- http://mantis.cs.colorado.edu/media/sensys_elf_external.pdf
- http://www.cs.tau.ac.il/~stoledo/Pubs/usenix2005.pdf
- http://www.tau.ac.il/~stoledo/Pubs/flash-survey.pdf
- http://nesl.ee.ucla.edu/projects/sos-1.x/
- http://www.tinyos.net/
- http://cvs.cens.ucla.edu/emstar/downloads.html
- http://www.tinyos.net/tinyos-1.x/doc/matchbox-design.pdf
- http://www.osservatori.net AA.VV – RFID tra presente e futuro – Collana quaderni AIP, Dip. Ing. Gestionale PoliMI, 2005