What's NExT? Sensor+Cloud?!

Kian-Lee Tan
National University of Singapore

Outline

- Emerging IT trends
  - Sensors, Web
- Cloud, SensorCloud & Challenges
- Preliminary work done at NUS
  - epiC
  - HipCloudS
- The NExt Center
- Conclusion

Sensors are everywhere!

Applications

- Ubiquitous healthcare
- Smart-home
- Telematics
- …

They are here to stay …

Sensor deployment just starting, with some estimates ~5-10B cell phones by 2015, and 7 trillion wireless devices serving 7 billion people in 2020 (WWRF)

The Vision

Mark Weiser
(1952 – 1999)
XEROX PARC
"Palo Alto Research Center"
http://www.parc.xerox.com

"In the 21st century the technology revolution will move into the everyday, the small and the invisible…"
Mark Weiser, 1988

"*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it*"
Mark Weiser, 1991

“By 2015, wirelessly networked sensors in everything we own will form a new Web. But it will only be of value if the ‘terabyte torrent’ of data it generates can be collected, analyzed and interpreted.”

- Gartner

The emergence of Web2.0

Web2.0 is now …

Web^2 (Ref [3])

Our Information Shadow

"Everything and everyone in the world casts an

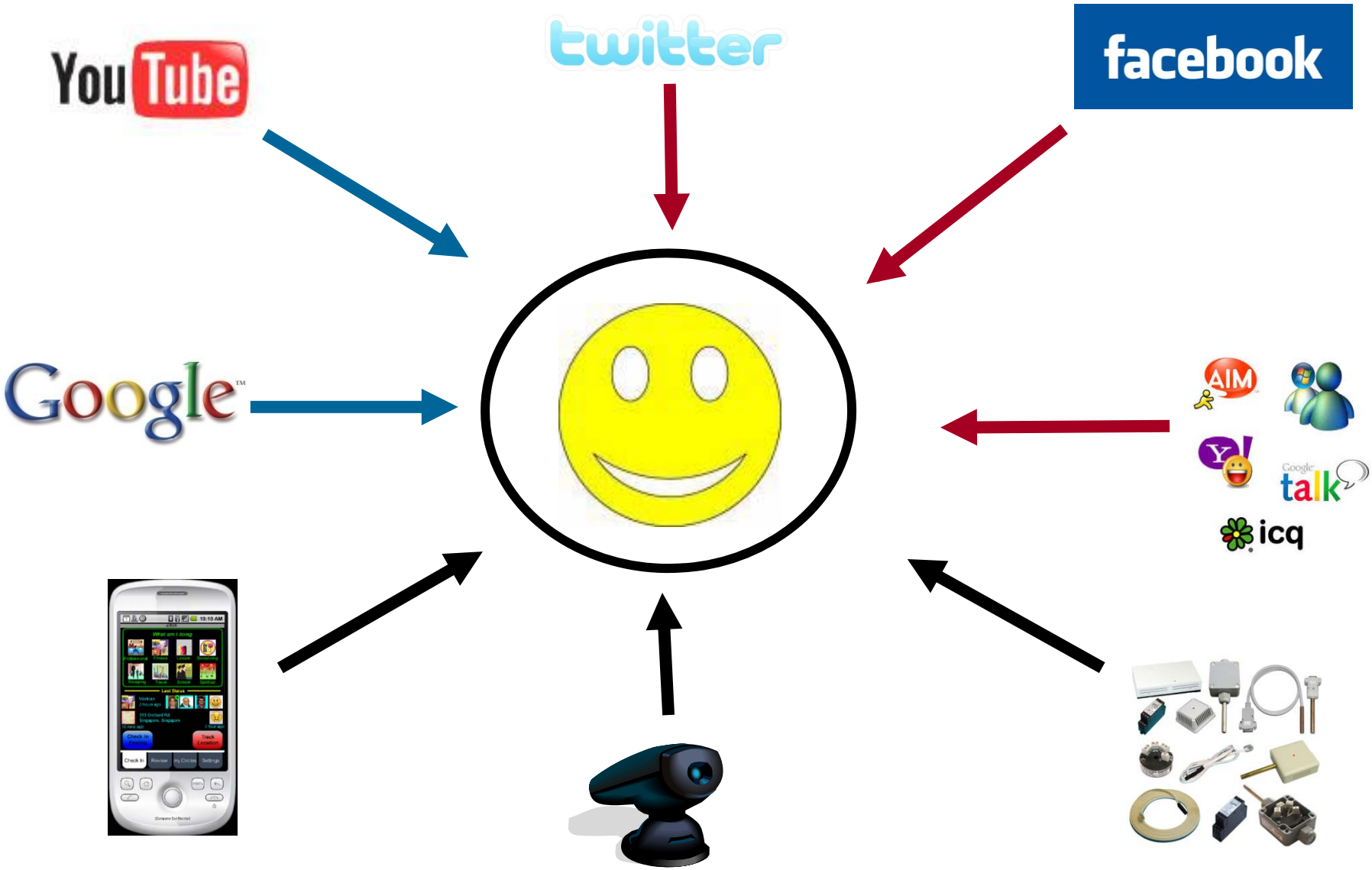
'Information Shadow'….

Increasingly, the web is the World.

Our cameras, our microphones, are becoming the eyes and ears of the Web, our motion sensors, proximity sensors its proprioception, GPS its sense of location…."

O'Reilly , Battelle 2009

Multi-Sensor Info-Rich Environment



What can we do?

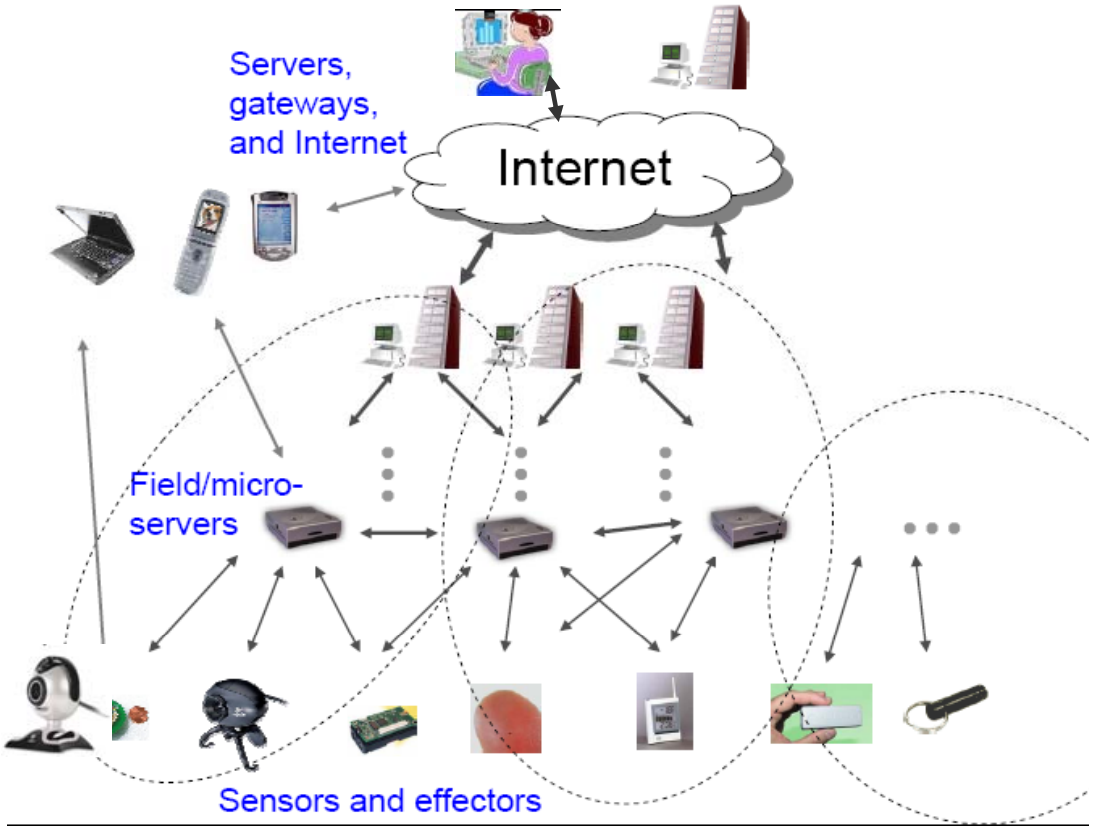
- Research Focuses

  At the Micro-level:

  - Multi-sensor analytics
  - Human (groups) behavioral analysis & recognition

  At the Macro-level:

  - Fusion of multi-sensor info to infer crowd behavior & region wide trends



- Mobile device analytics – at user level, friends-level...
- Aggregation of multisource info
- Again the issues of search, filter, or alert

The future is SMART(er)-*

Application Scenarios

- Augmented Reality

- Interaction with social networks

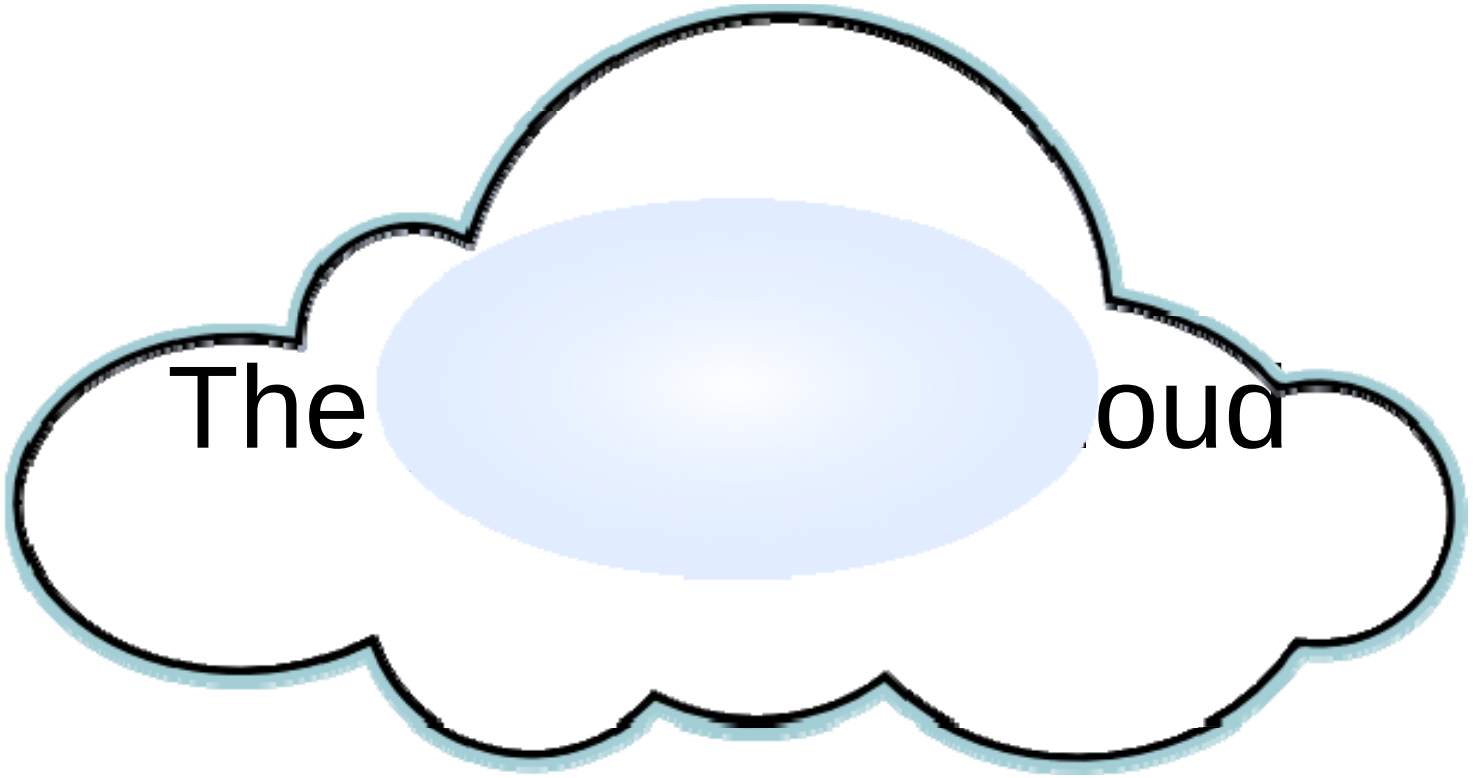
- The stranger you know

Outline

- Emerging IT trends
  - Sensors, Web
- Cloud, SensorCloud & Challenges
- Preliminary work done at NUS
  - epiC
  - HipCloudS
- The NExt Center
- Conclusion

A Scenario (Source: Ref [4])

FACT: Today's cell phones are not so powerful!

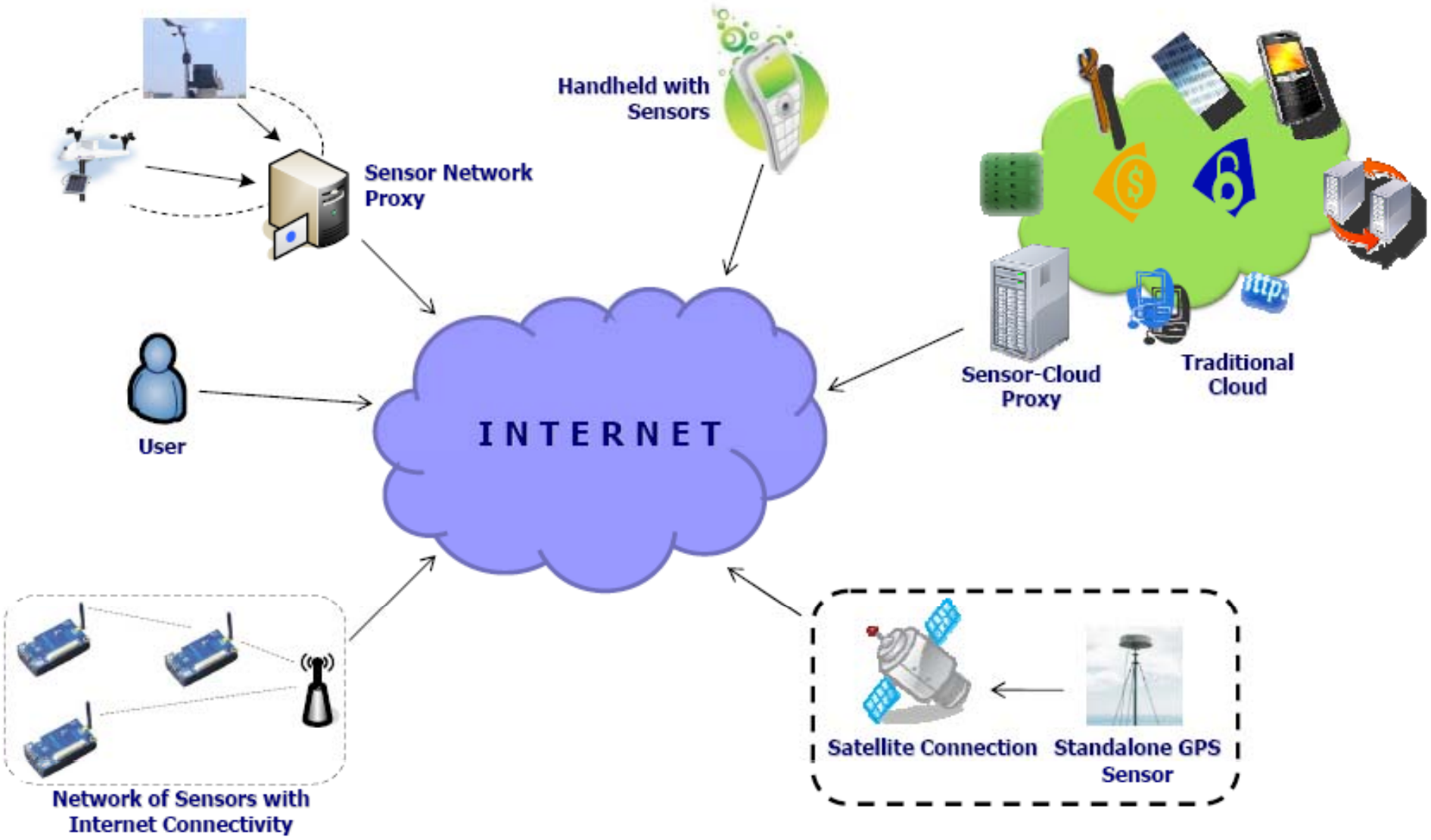

The
oud

Sensor-Cloud

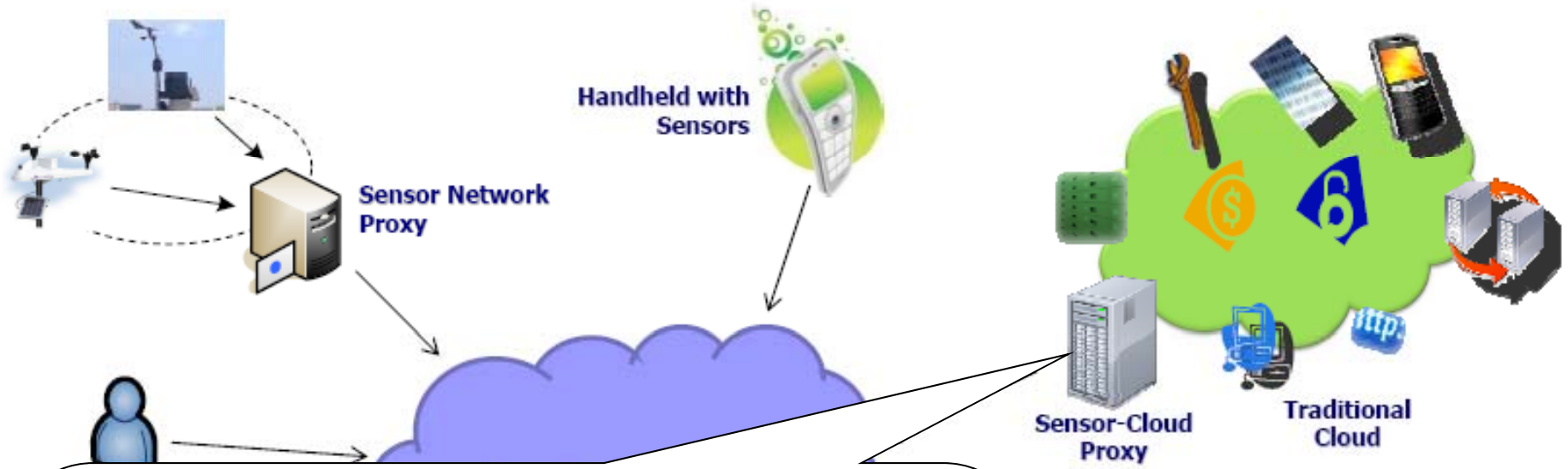
"An infrastructure that makes truly pervasive computation using sensors as interface between physical and cyber worlds, the data-compute clusters as the cyber backbone and the internet as the communication medium" (Ref [4])

- Offers storage resources for sensor data and computation power for end users and applications
- Enables large-scale sharing and collaborations among users and applications on the cloud
  - Data sharing
  - Sensor devices sharing
- Enables sensors as cloud services
- Supports complete sensor data life cycle from data collection to the backend decision support system

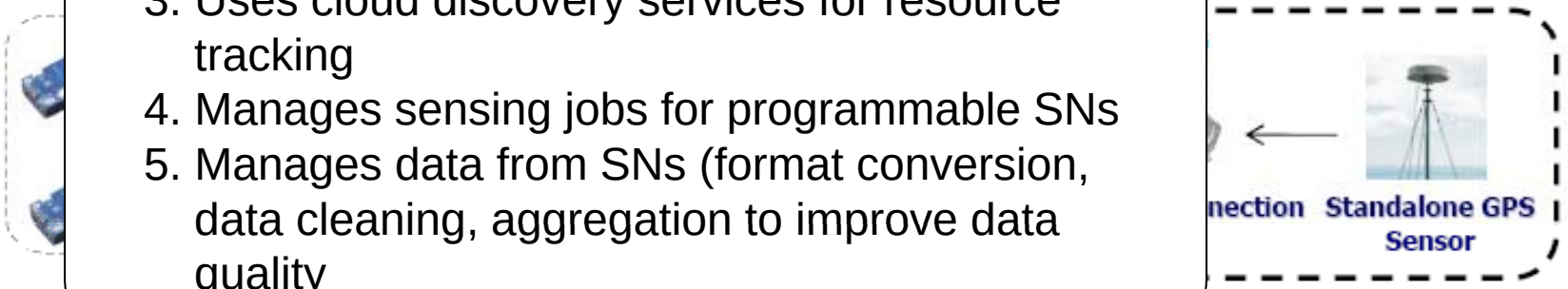
Sensor Cloud Architecture (Ref [4])



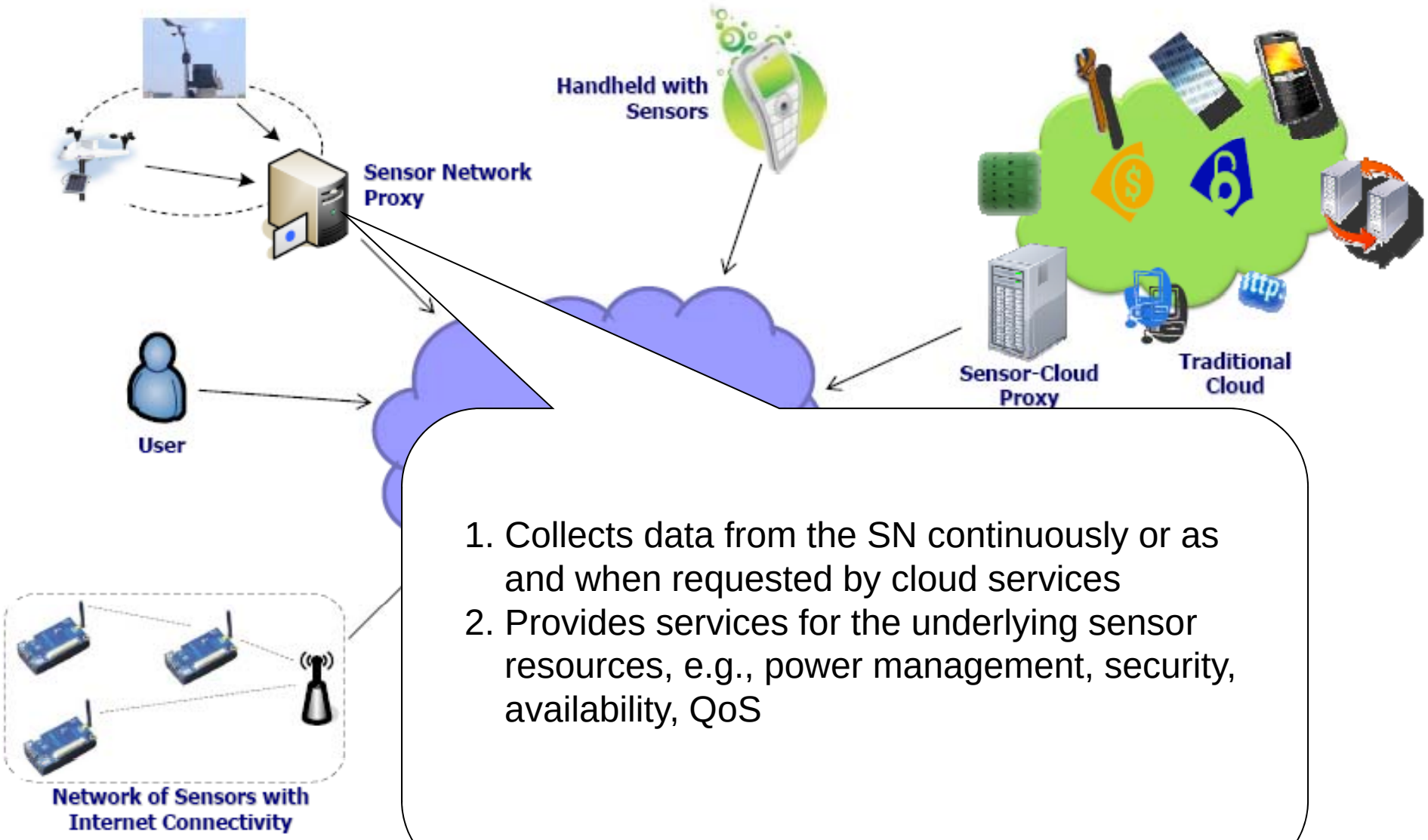
Sensor Cloud Architecture (Ref [4])



1. Manages SN connectivity between sensor resources and cloud
2. Exposes sensor resources as cloud services
3. Uses cloud discovery services for resource tracking
4. Manages sensing jobs for programmable SNs
5. Manages data from SNs (format conversion, data cleaning, aggregation to improve data quality
6. Can be virtualized to live in the cloud!



Sensor Cloud Architecture (Ref [4])



1. Collects data from the SN continuously or as and when requested by cloud services
2. Provides services for the underlying sensor resources, e.g., power management, security, availability, QoS

Challenges (I)

- Complex Event Processing and Management
  - How to support different kind of events and messages
  - How to synchronize events that arrive from different source in different time due to network delays.
  - How to identify the context (temporal, spatial, semantic) in which a situation detection is relevant
  - How to change event processing rules without stopping the system (hot updates)
  - How to support vast numbers of events and conditions in an optimal way
  - How to detect cycles in rule firing sequence.

Challenges (II)

- Massive scale and real-time data processing
  - All point continuous range queries (APCRQ)
- Large scale (distributed) systems
- Data and query privacy

Other issues like energy efficiency, harvesting collective intelligence, …

Outline

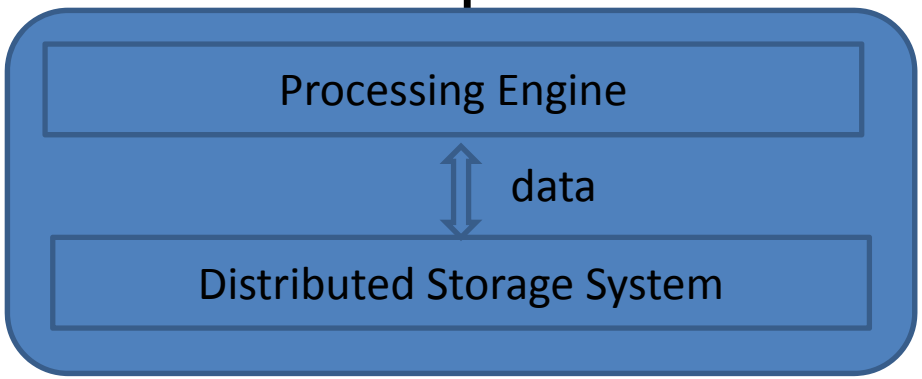
- Emerging IT trends
  - Sensors, Web
- Cloud, SensorCloud & Challenges
- Preliminary work done at NUS
  - epiC
  - HipCloudS
- The NExt Center
- Conclusion

epiC: *e*lastic *p*ower-aware data *i*ntensive *C*loud

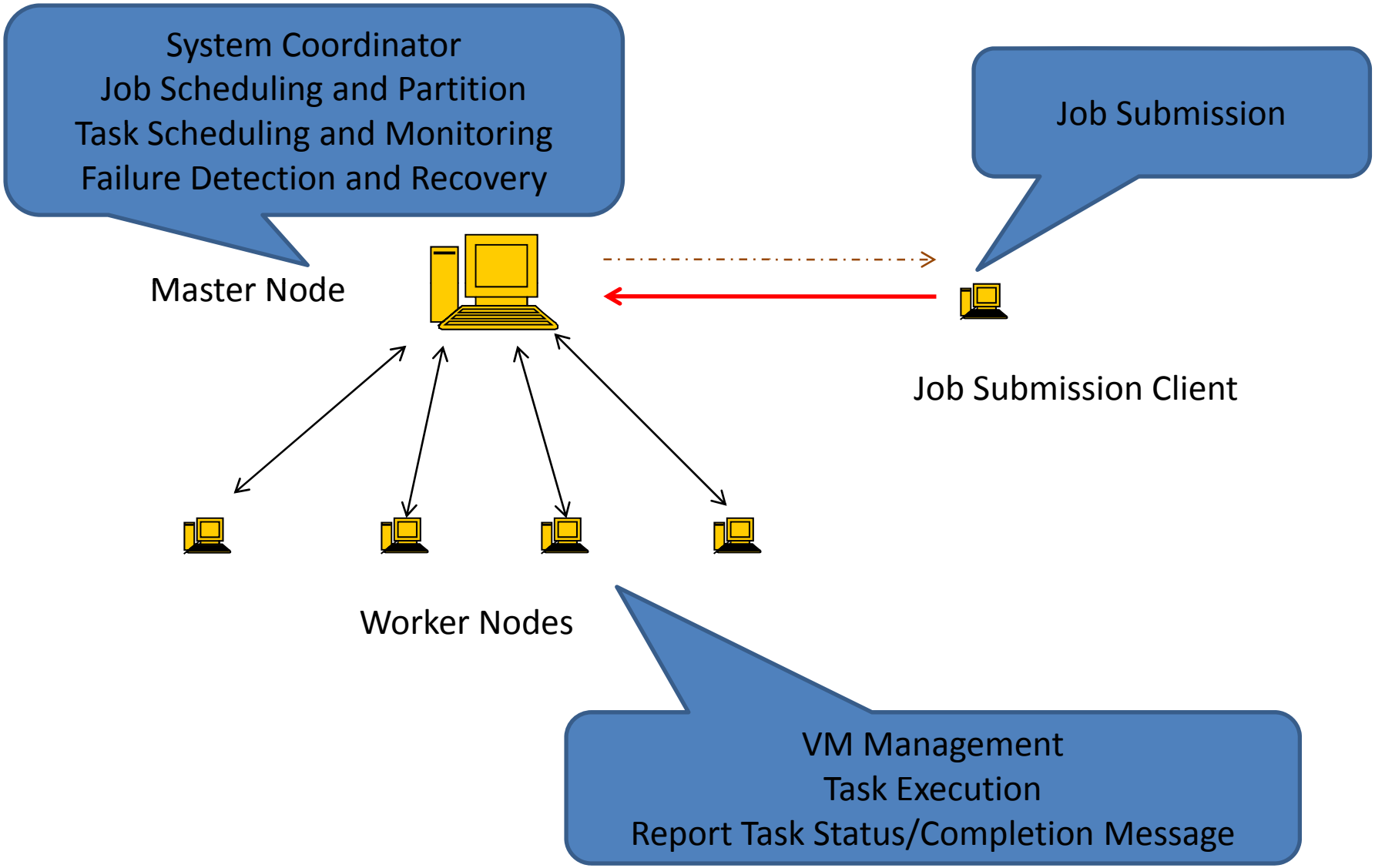
http://www.comp.nus.edu.sg/~epic/

epiC (Ref [6])

- A new DB-enabled cloud system with
  - Similar processing model to Dryad
  - Scalability, Flexibility, Fault Tolerance
  - Cost-based query optimization
  - Storage systems that support both OLTP and batch processing
  - Real-time processing
- Composes of two components

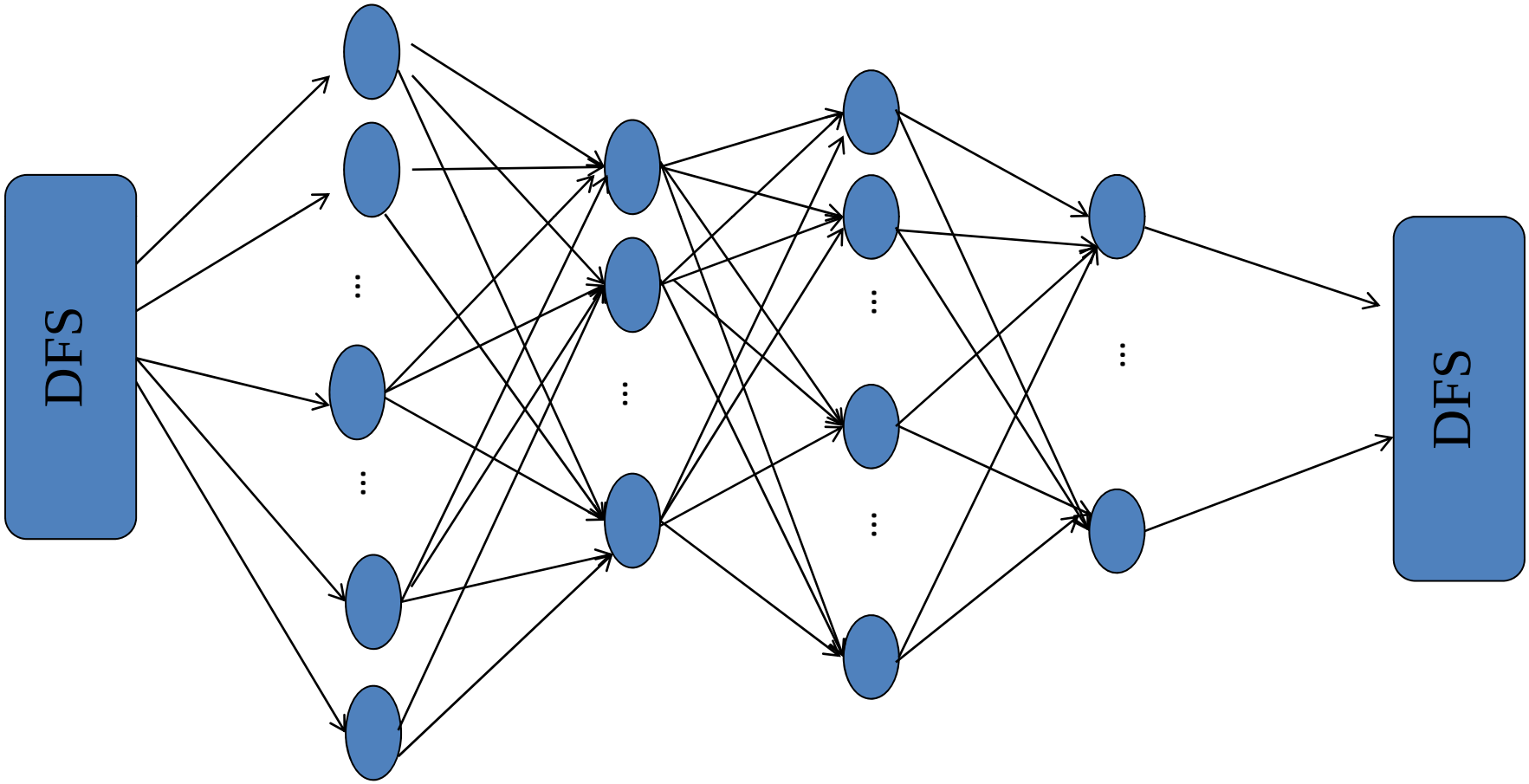


E^3: epiC Execution Engine



E^3 :Flexible Processing Model

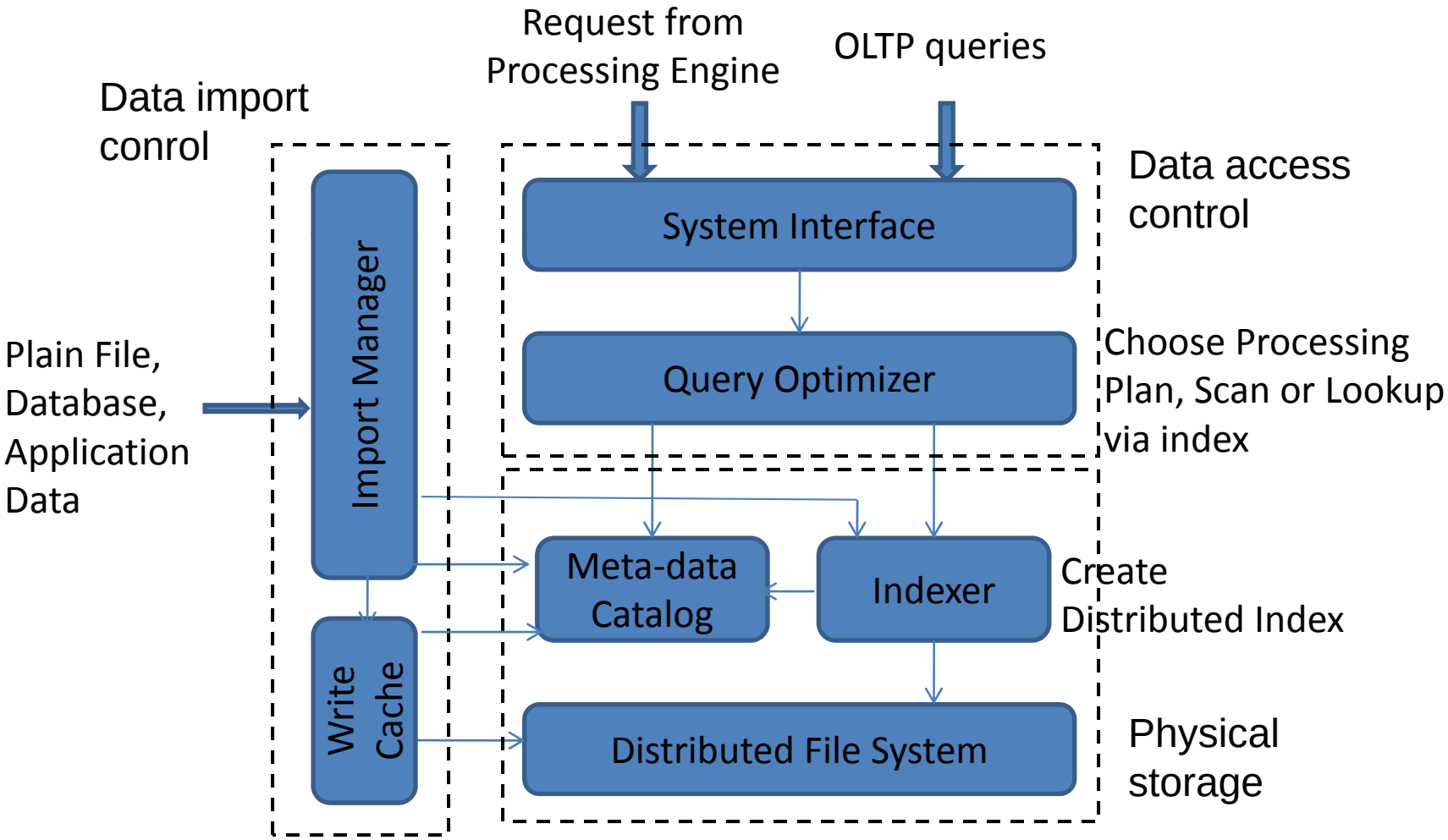
- Support multi-stages processing in one job



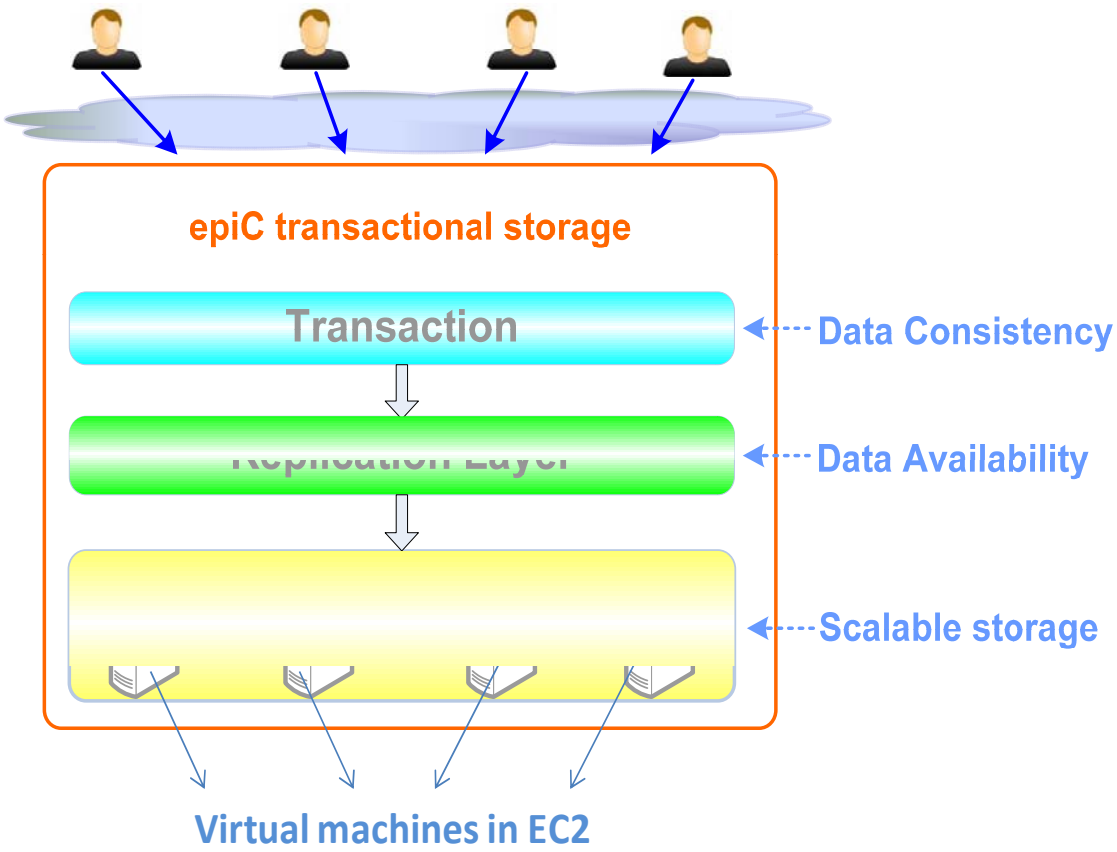
E^3 : More features

- DB-like
  - Provide system defined functions
  - Support user defined functions
  - Support high level SQL-like language in the future
  - Easy and efficient development for users
- Flexible VM management
  - VM Manager in each worker node
  - VMs are provided according to its own free resources (CPU cores, Disk, memory etc.)
- Fault Tolerance
  - Selected stages of intermediate results are materialized in local disks

epiC's Distributed Storage System

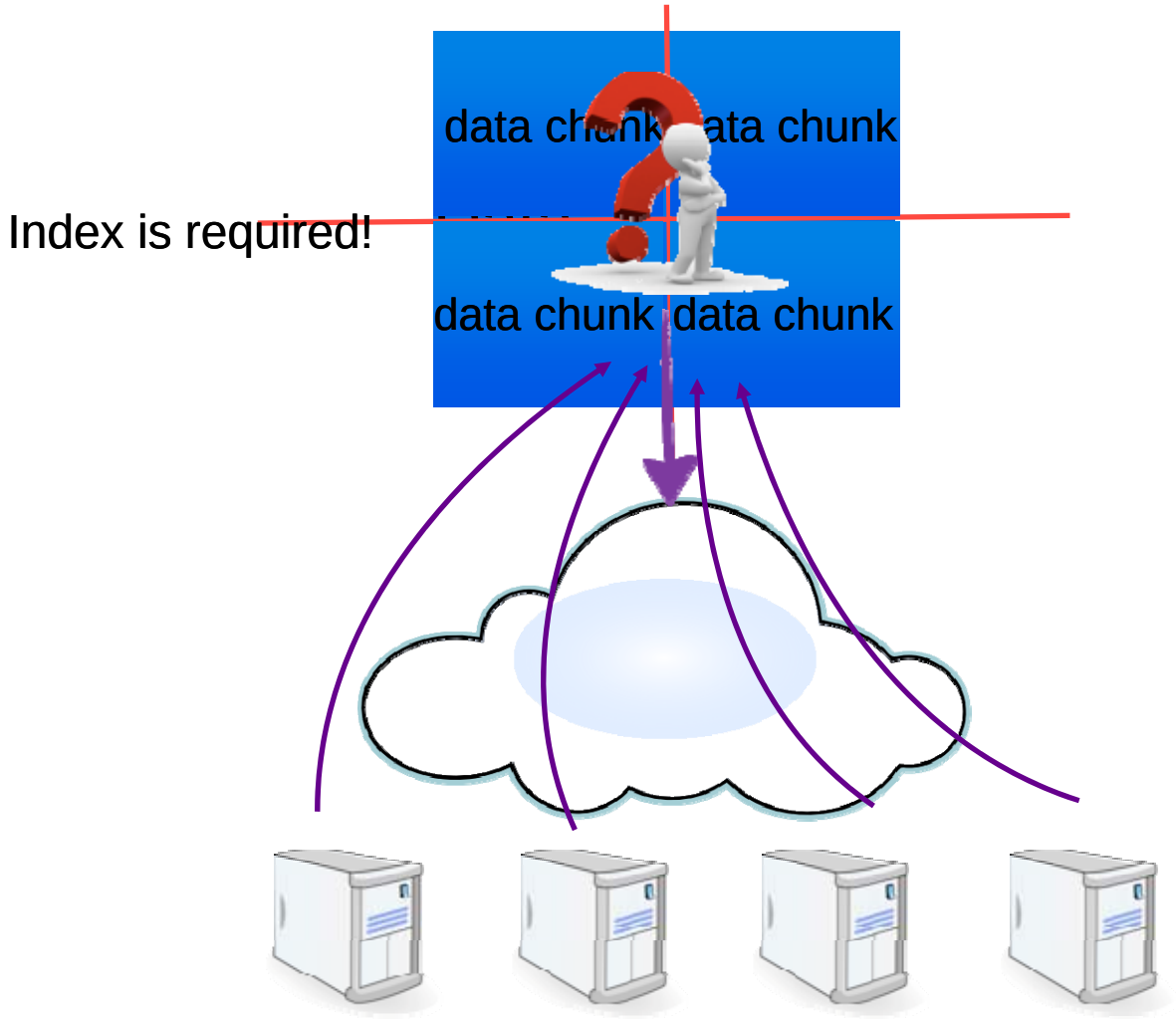


Transaction Management in Storage Layer



Efficient B-Tree Based Indexing for Cloud Data Processing (Ref [5])

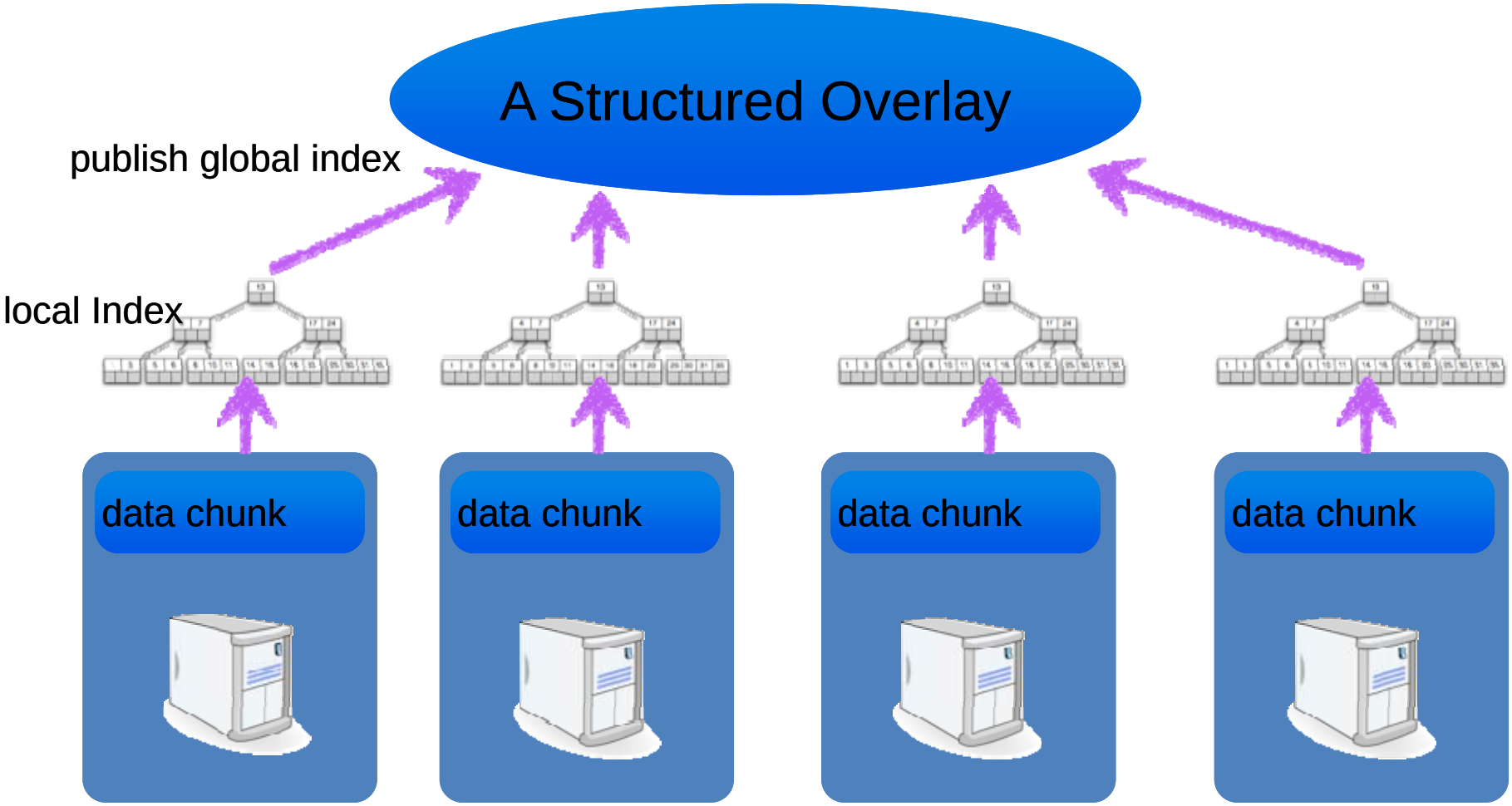
A Naïve Cloud Solution (for Search)



How to Build Indexes for Cloud Storage System

- Google applies MapReduce to compute inverted indexes
  - Good for web documents, which are updated infrequently
  - Not applicable to database applications, where data are continuously updated
- Our approach: a two layer indexing framework

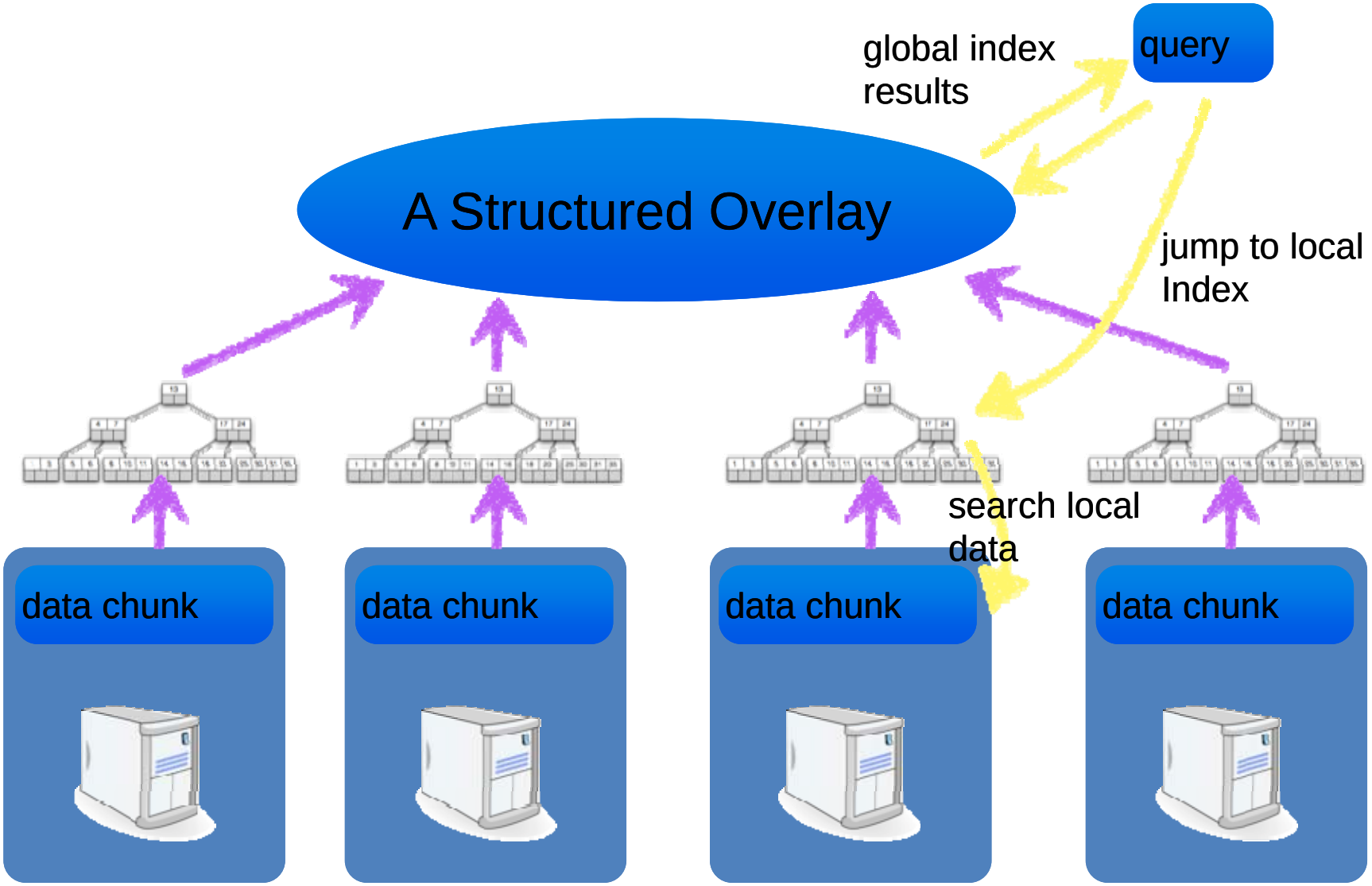
A Two-Layer Indexing Framework



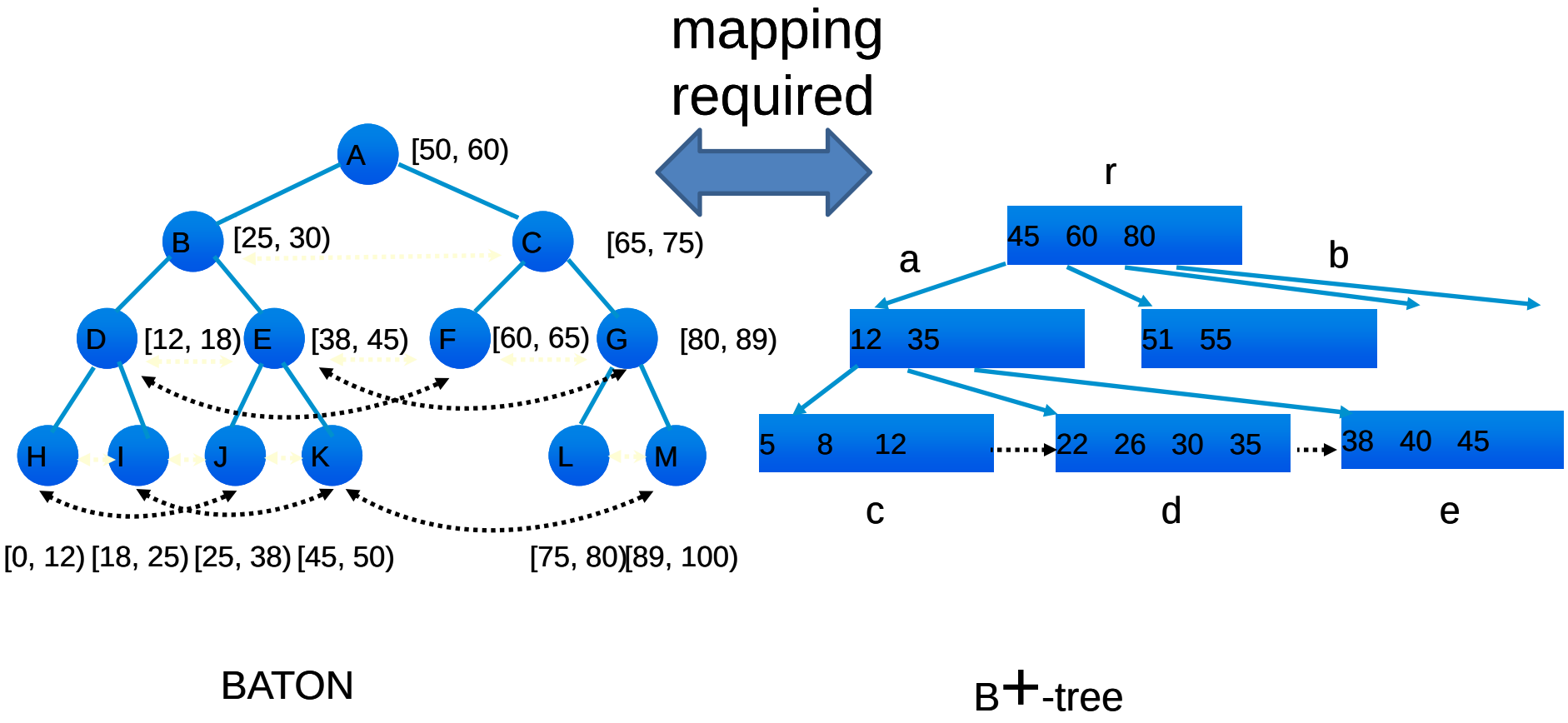
Index Format

- Local Index: A B+-tree
- Global Index: (BLK, Range, Keys, IP)
  - BLK: The Pointer to the local index block
  - Range: The range of the corresponding local index node
  - Keys: The keys maintained by the local index node
  - IP: IP addresses of the cluster node hosting the index

Search Process



Challenge 1: How to Publish Local Index in Overlay Network

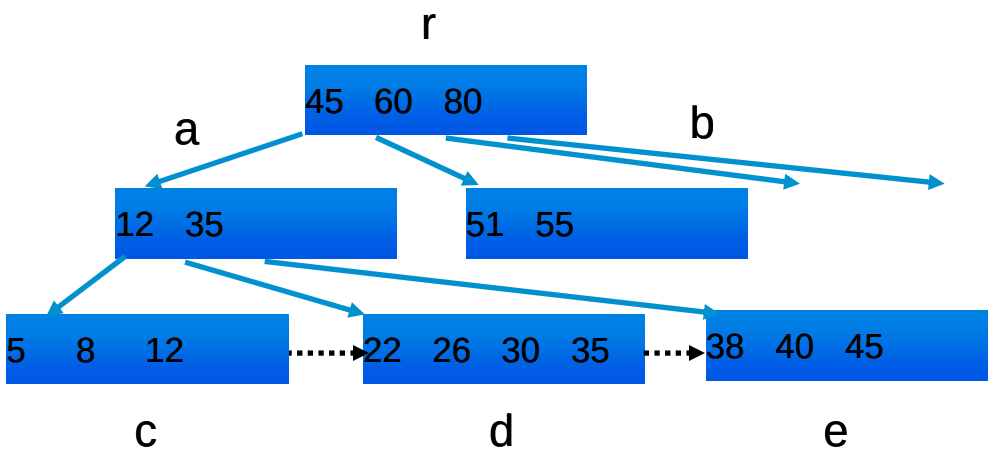


BATON

B+-tree

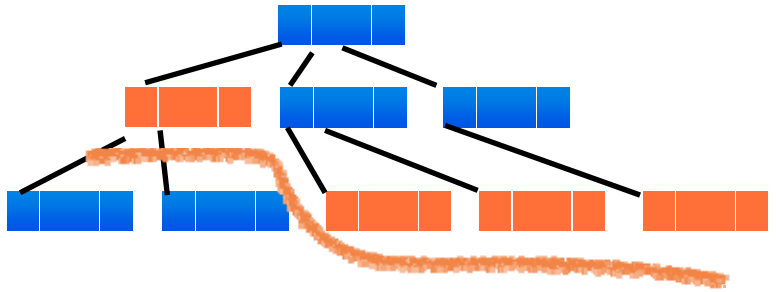
Mapping Function

- Each BATON node responds for a range
- Each B-tree node also handles a range
  - root node's range is [min, max], r=[5, 100]
  - a node inherits its range from the parent, a=[5, 45]
- Publish a B-tree node to the BATON node, whose range covers the range of the B-tree node

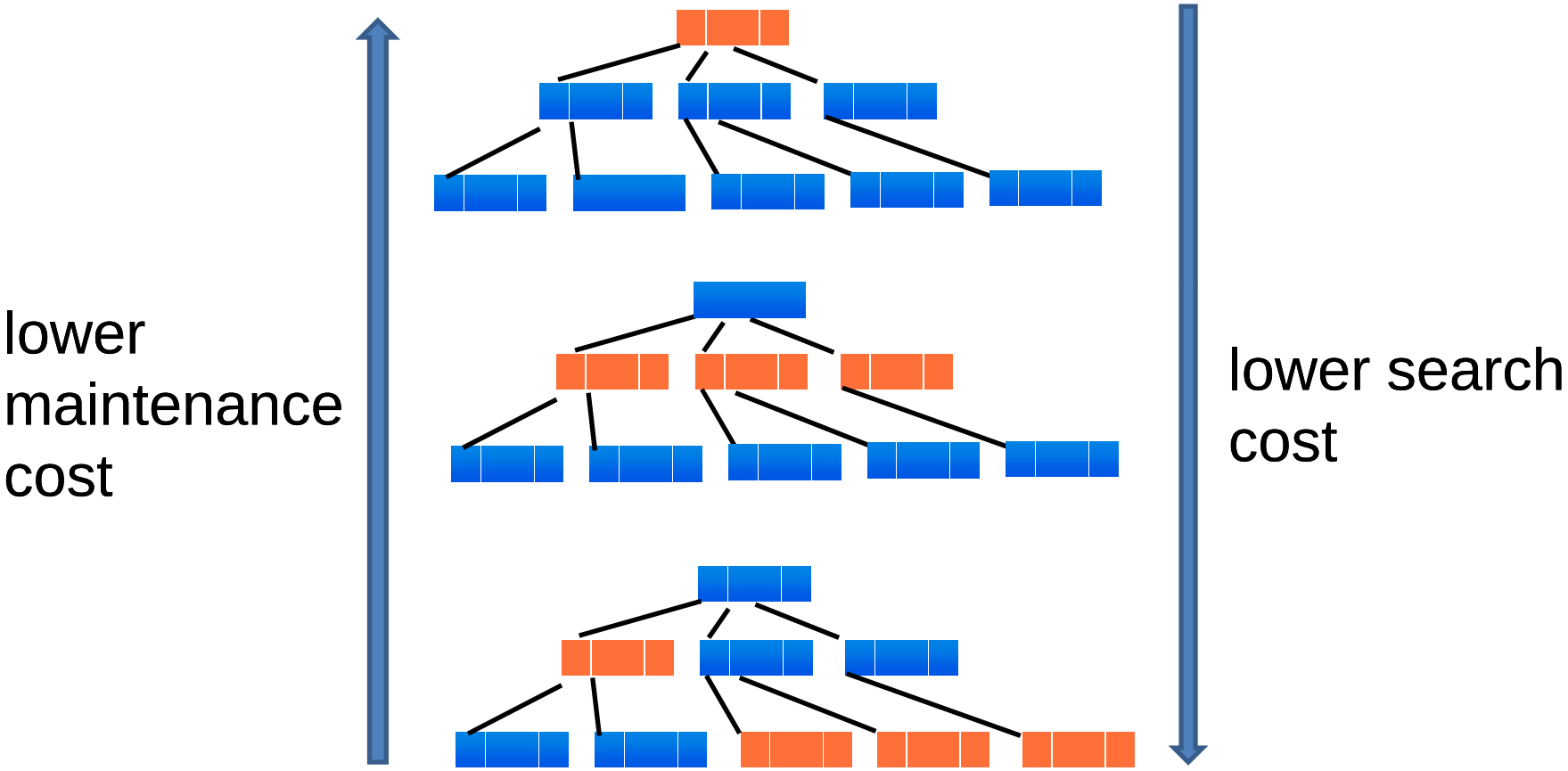


Challenge 2: Which B-tree node should be published

- Publishing all tree nodes are costly
- The published index must be complete
  - A slice of the B-tree
  - No false negative for the queries



An Index Expansion Scheme



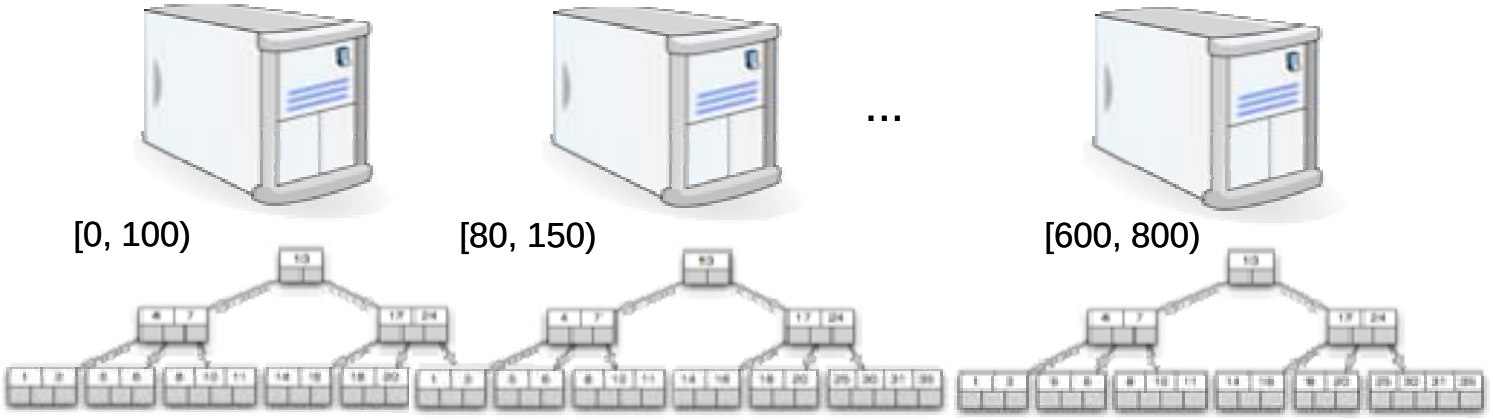
A cost-model is applied to adjust the indexing strategy based on query patterns

Update Mode

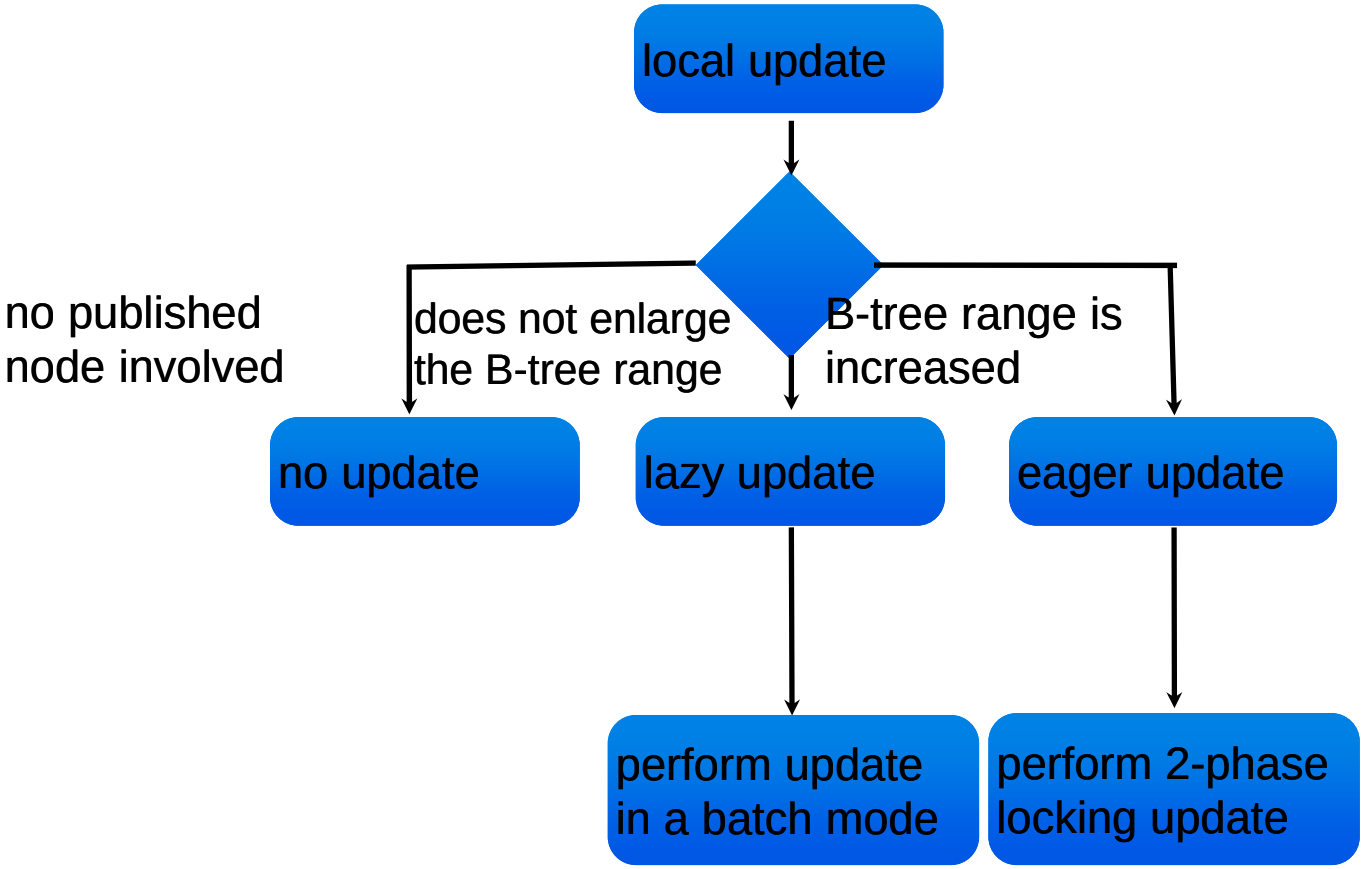
- An update in local B-tree node ni
  - if ni is not published, no update for global index
  - otherwise, we need to synchronize ni with its index
  - as only a portion of nodes are published, the update cost is reduced
  - moreover, if we only publish the inner-node, the number of remote updates is quite small

Lazy Update

- To further reduce the update cost, we adopt the lazy update strategy
- Each B-tree represents a range of data
- If its range does not enlarge, we can still forward query to the proper nodes
- If lazy update leads to false positive, we perform the search from the root of local B-tree

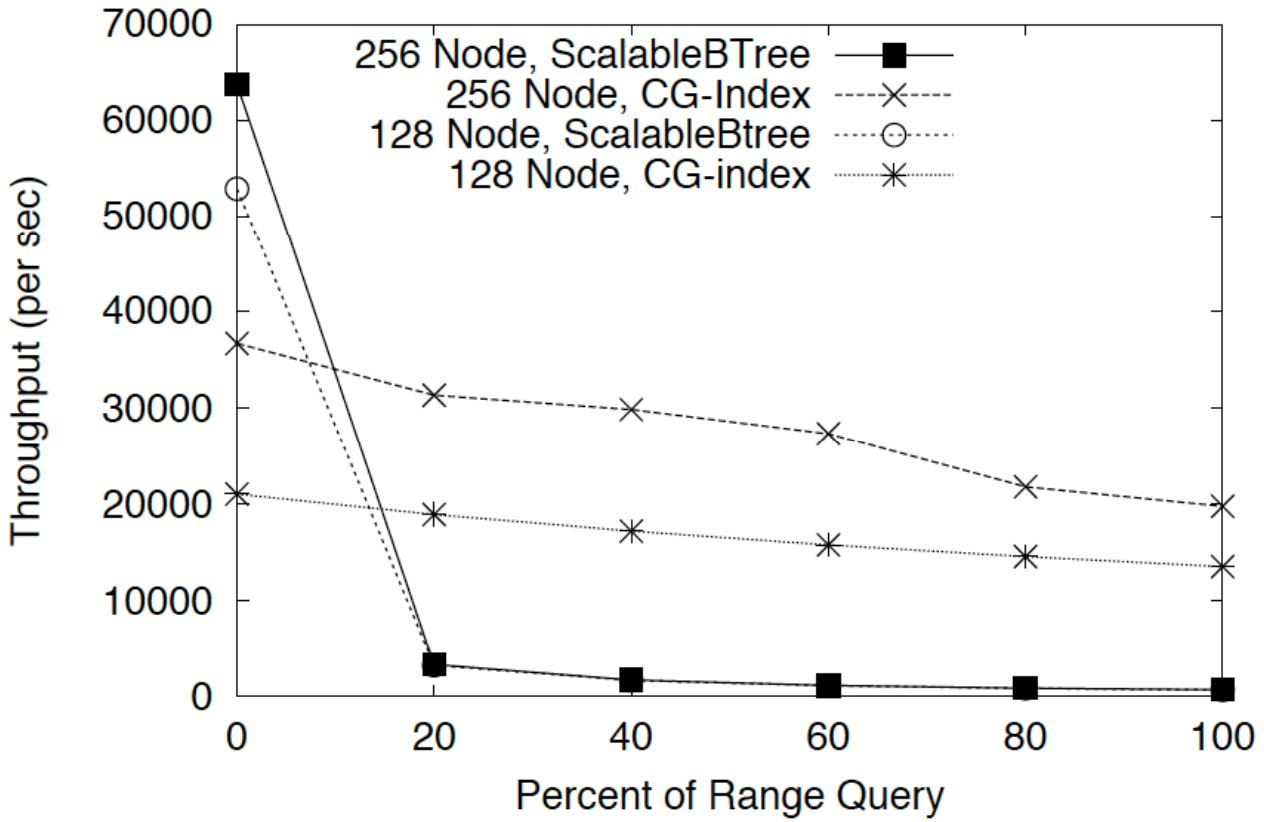


Lazy Update (cont.)



Comparison with Scalable B-tree

X-axis : number of range query / (number of range query+number of exact query)



Outline

- Emerging IT trends
  - Sensors, Web
- Cloud, SensorCloud & Challenges
- Preliminary work done at NUS
  - epiC
  - HipCloudS
- The NExt Center
- Conclusion

HipCloudS

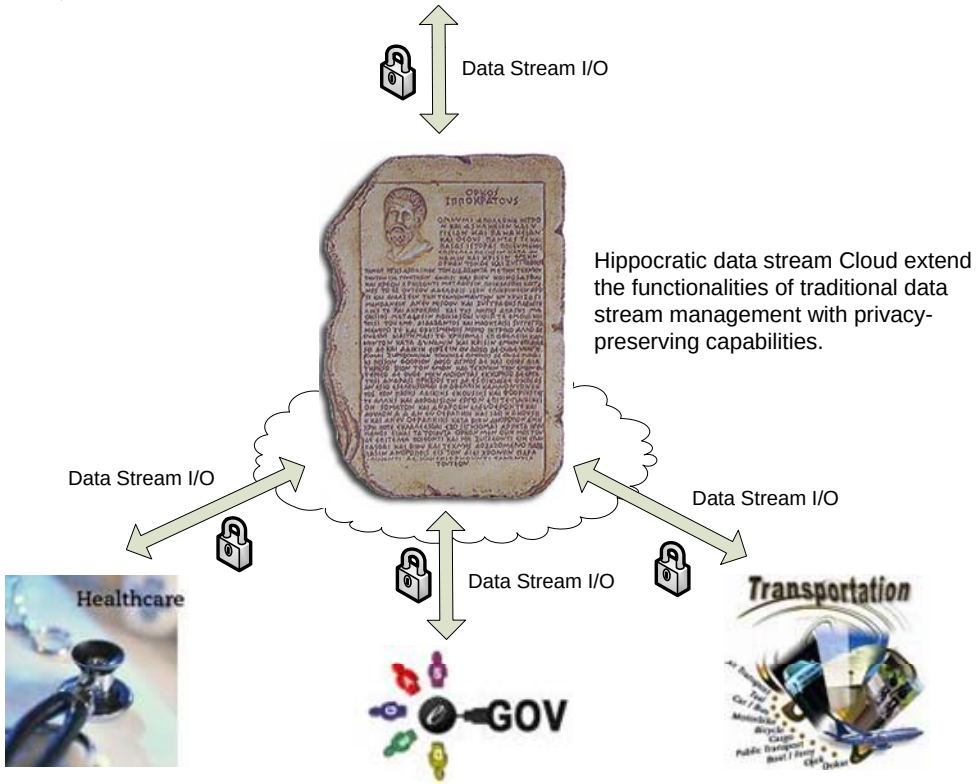
Hippocratic Data Stream Cloud for Secure, Privacy-preserving Data Analytics Services (Ref [1])

Institute of Infocomm Research

Center for Maritime Studies, NUS

Department of Computer Science, NUS

"And about whatever I may see or hear in treatment, or even without treatment, in the life of human beings – things that should not ever be blurted out outside – I will remain silent, holding such things to be unutterable" – Hippocratic Oath



Principles:

- Purpose-specification
- Consent
- Limited collection
- Limited use
- Limited disclosure
- Limited retention
- Accuracy
- Safety
- Openness
- Compliance
- Faithful representation
- Guarantee QoS
- Seamless integration of data & summaries
- Fault tolerance & high availability
- Agnostic to heterogeneous environment

Limited Collection

- Data collection should be limited to the minimum amount of data that satisfies the user's specified purposes.
  - @ Stream-level, @ Attribute level, @Tuple level
- Dynamism
  - Existing queries expire and new queries arrive, rules and filters that the system uses to limit the data collection change dynamically.

Limited Retention

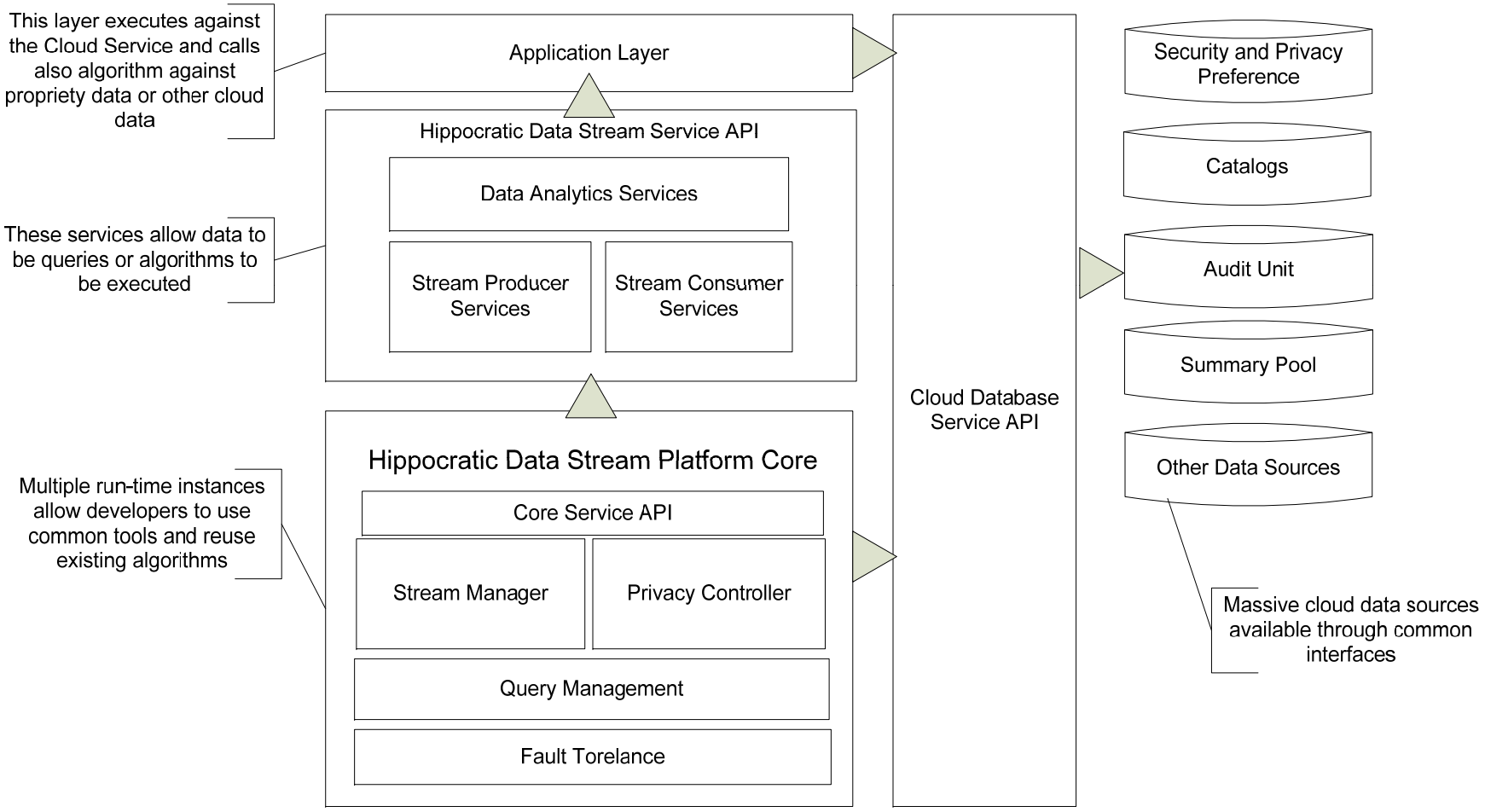
- Data collection should be retained only as long as necessary for the fulfillment of the purposes for which it has been collected.
  - They can be in the system's input buffers, the queries' input buffers, or queued in one or more query plan operators and summaries over the stream.
- Retention time challenge
  - @Queries window size
  - @Fading summaries
- Dynamism

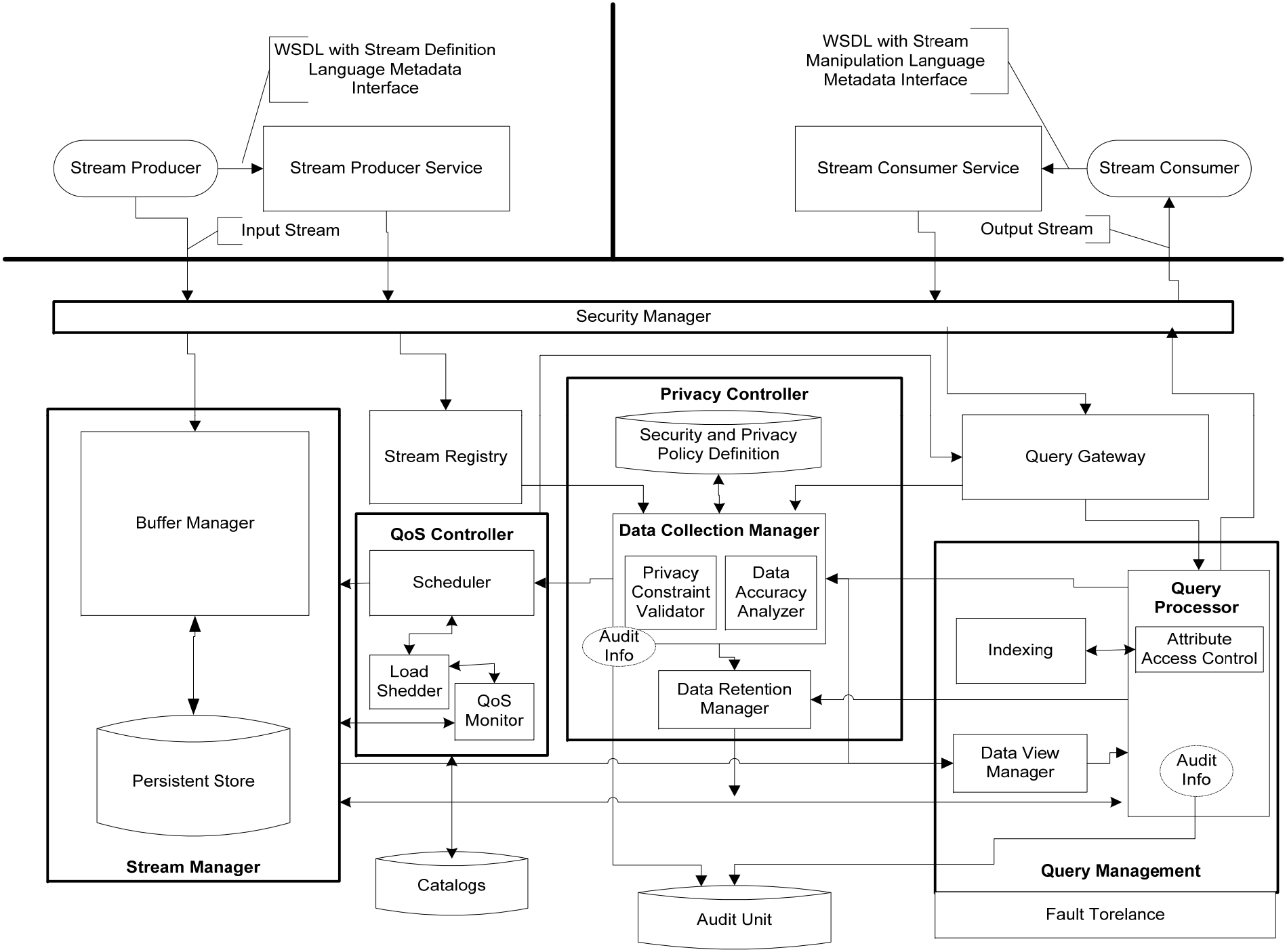
Limited Use/Disclosure

- The data is used by queries that do not violate the purposes of the collected data
- The system is not allowed to release any information to a third party that is outside the system without the owner's approval.
  - "Purpose-recipient" as a central concept of this principle.
  - Manage disclosure at a very fine granularity is important.

Limited Disclosure

- Models of Limited Disclosure
  - Stream semantics
    - Conceptually, defines a ‘view’ of each stream for each purpose-recipient pair, based on the disclosure constraints specified in the privacy policy.
    - Independent of any queries
  - Query semantics
    - Prohibited data is removed from a query’s result set based on the purpose-recipient pair and the query itself.


This layer executes against the Cloud Service and calls also algorithm against propriety data or other cloud data
Application Layer
Hippocratic Data Stream Service API
Data Analytics Services
These services allow data to be queries or algorithms to be executed
Stream Producer Services
Stream Consumer Services
Hippocratic Data Stream Platform Core
Multiple run-time instances allow developers to use common tools and reuse existing algorithms
Core Service API
Stream Manager
Privacy Controller
Query Management
Fault Torelance
Cloud Database Service API
Security and Privacy Preference
Catalogs
Audit Unit
Summary Pool
Other Data Sources
Massive cloud data sources available through common interfaces


WSDL with Stream Definition Language Metadata Interface
WSDL with Stream Manipulation Language Metadata Interface
Stream Producer
Stream Producer Service
Stream Consumer Service
Stream Consumer
Input Stream
Output Stream
Security Manager
Buffer Manager
Stream Registry
Privacy Controller
Security and Privacy Policy Definition
Query Gateway
QoS Controller
Scheduler
Load Shedder
QoS Monitor
Data Collection Manager
Privacy Constraint Validator
Data Accuracy Analyzer
Audit Info
Data Retention Manager
Query Processor
Indexing
Attribute Access Control
Data View Manager
Audit Info
Persistent Store
Stream Manager
Catalogs
Audit Unit
Query Management
Fault Torelance

Access control in data streams (Ref [2])

- Traditional access control is not suitable for data streams:
  - static and bounded databases vs. unbounded and infinite streams;
  - one time and ad-hoc queries vs. continuous and long running queries;

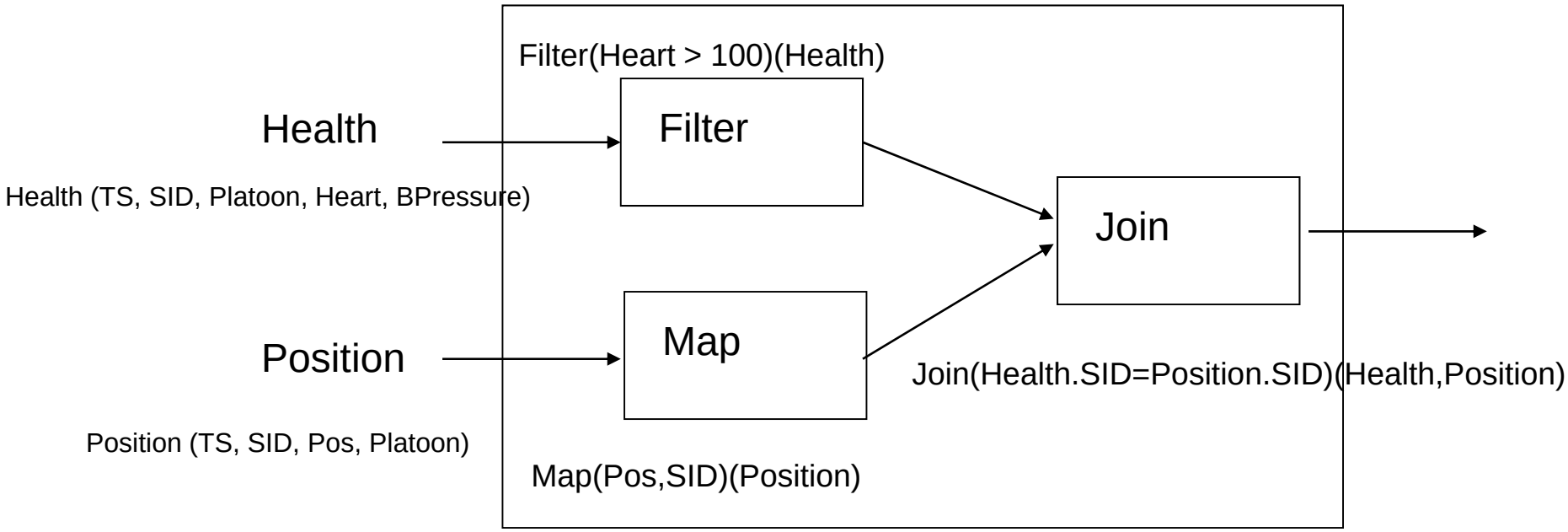
Query-driven access control enforcement

vs.

Data-driven access control enforcement

Background: Aurora

- Aurora data stream model [MIT,Brown Un., Brandeis Un.]:
  - Borealis= Aurora + Medusa
  - **NOW:** StreamBase Product



Aurora: Boxes

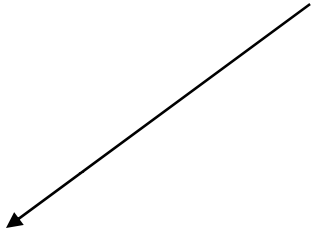
- Aurora Stream Query Algebra (SQual) operators:
  - *Filter*: Filter acts like a case statement and can be used to route input tuples to alternative streams
  - *Map*: Map is a generalized projection operator
  - *Union*: Union is used to merge two or more streams into a single output stream.
  - *Bsort*: BSort performs a buffer-based approximate sort
  - *Aggregate:* aggregate functions on sliding windows
  - *Join:* Combining data from multiple streams
  - *Resample*: Resample is an asymmetric, semijoin-like synchronization operator that can be used to align pairs of streams

Access control policies for data streams

(subject, object, privilege, GT,WT)

Access control policies for data streams

(subject, object, privilege, GT,WT)

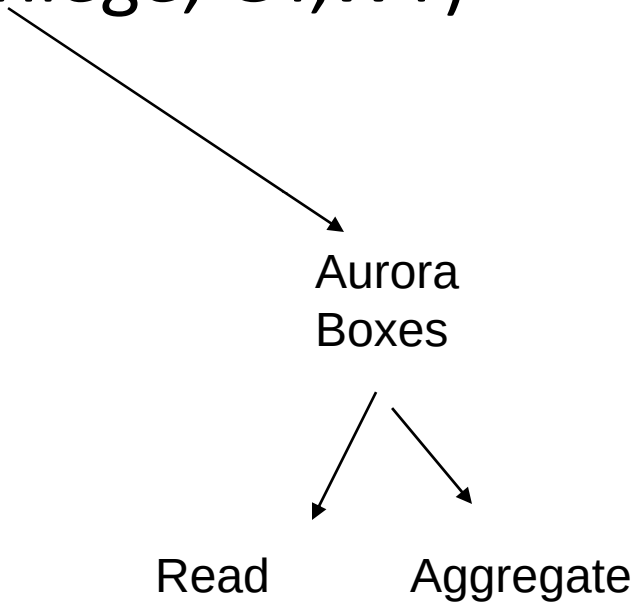


Roles

(Captain, object, privilege, GT,WT)

Access control policies for data streams

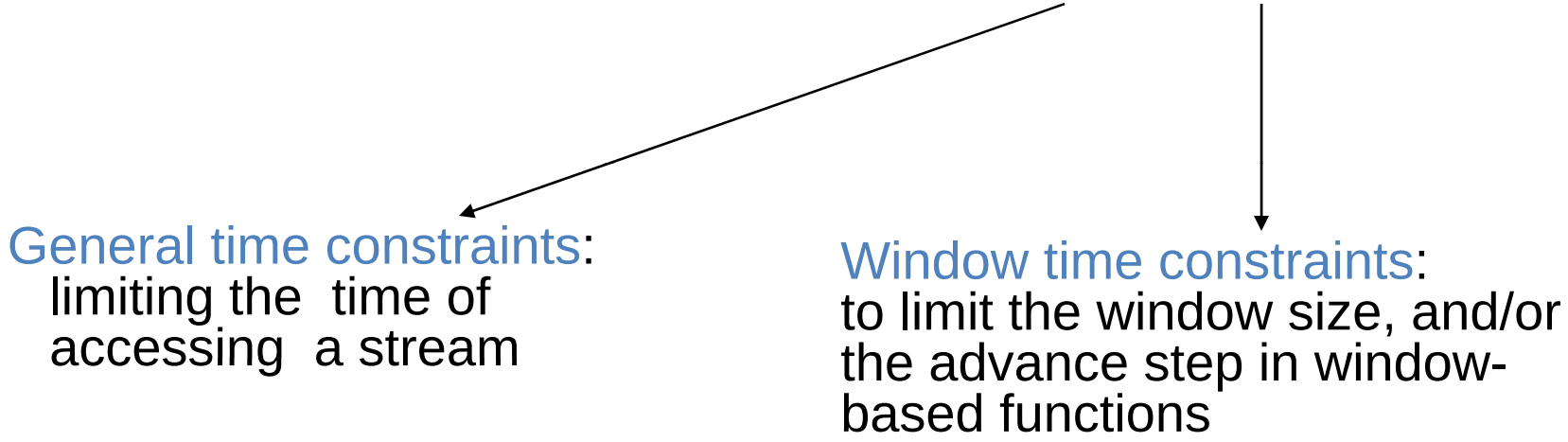
(subject, object, privilege, GT,WT)



(Captain, object, AVG, GT,WT)

Access control policies for data streams

(subject, object, privilege, GT,WT)



(Captain, object, AVG, [Start(a),End(a)],[1,1])

Access control policies for data streams

(subject, object, privilege, GT,WT)

How to model
protection objects?

(Captain, object, AVG, [Start(a),End(a)],[1,1])

Access control policies for data streams

- Protection objects are modelled as a view on streams

```
CREATE VIEW Soldier_heart AS
SELECT Heart,SID
FROM Health
WHERE  Health.Platoon ≠ self.Platoon;
```

- Since a standard query language for data streams has not yet emerged:
  - Protection object specification is a triple:

(STRs, ATTs, EXPs)

```
(Health,  {Heart,SID}, Health.Platoon ≠ self.Platoon)
```

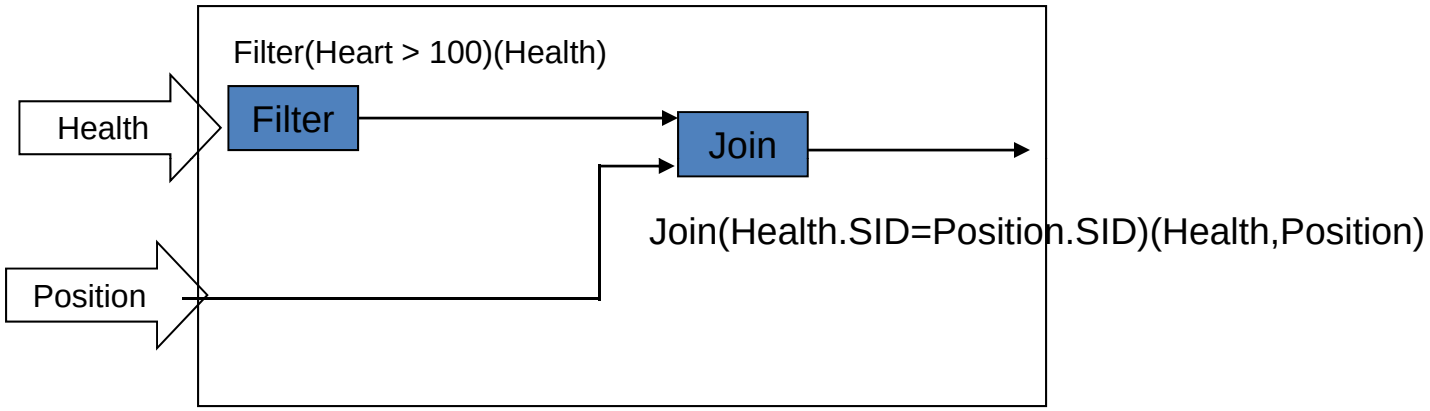
(Captain, {Health, {Heart, SID}, Health.Platoon ≠ self.Platoon}, AVG, [Start(a),End(a)],[1,1])

Examples (SysAuth)

Subj	Protection object spec			Priv	GTC	WTC
	STRs	ATTs	EXPs			
Captain	Position	Pos, SID	Position.Platoon = self.Platoon	Read		
Captain	Position	Pos, SID	Pos > k	Avg	[Start(a), End(a)]	[1, 1]
Doctor	Health	Heart, SID	Health.Platoon = self.Platoon	Read		
Doctor	Health	Heart, SID	Health.Platoon ≠ self.Platoon	Read	[Start(a), End(a)]	
Doctor	Health, Position	Heart, SID	Position.SID = Health.SID AND Pos < k2	Read		

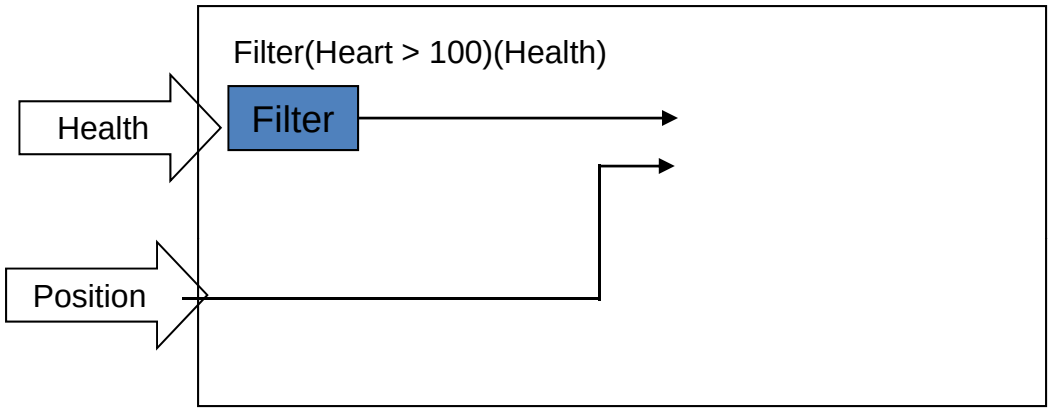
Access control enforcement

- Access control enforcement at query definition time

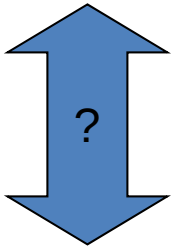


- Whenever a user tries to insert a box in a graph, the reference monitor:
  - (A) Checks the SystAuth catalog
  - (B) Decides whether
    - To deny the operation (the box cannot be inserted)
    - To grant/authorize the operation (the box can be inserted)
    - To partially authorize the operation (the box must be modified)

Access control enforcement



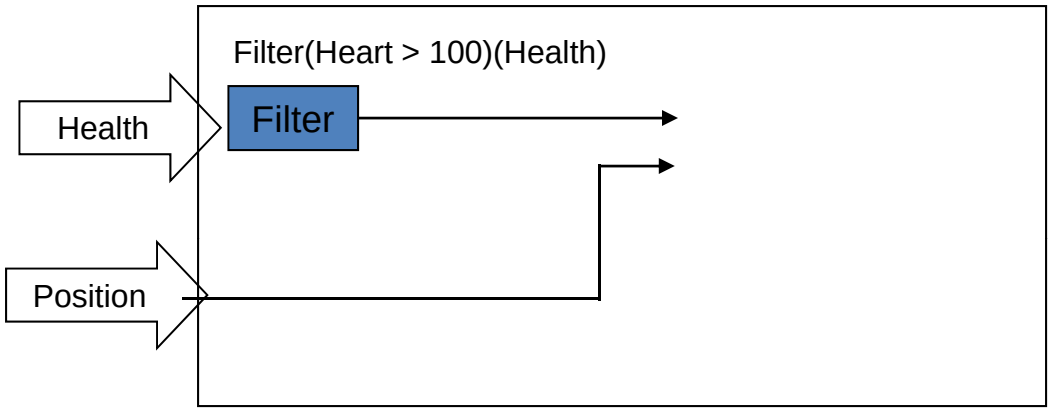
Access request



AC policies

Sub	Protection object spec			Priv	GTC	WTC
	STRs	ATTs	EXPs			
Captain, Doctor	Position	Pos, SID	Position.Platoon = self.Platoon	Read		
Captain	Position	Pos, SID	Pos > k	Avg	[Start(a), End(a)]	[1, 1]
Doctor	Health	Heart, SID	Health.Platoon = self.Platoon	Read		
Doctor	Health	Heart, SID	Health.Platoon ≠ self.Platoon	Read	[Start(a), End(a)]	
Doctor	Health, Position	Heart, SID	Position.SID = Health.SID AND Pos < k2	Read		

Access control enforcement



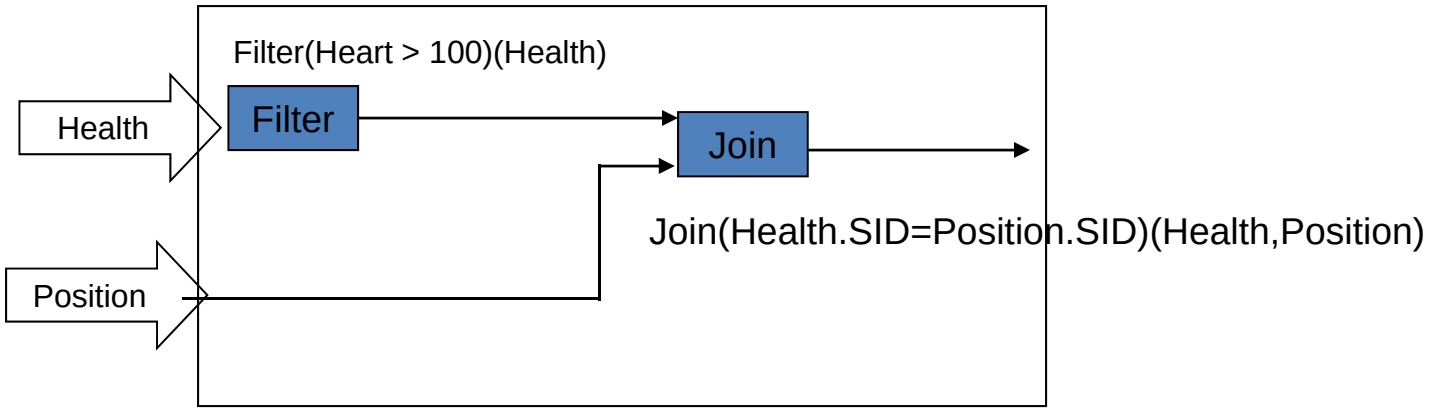
Obj1=(Health, *, {Heart>100})

Obj2=(Position, *, null)

- Solution: algo to generate the protection object like representation of a data stream

Access control enforcement

- Access control enforcement at query definition time



- Whenever a user tries to insert a box in the graph, the reference monitor:
  - (A) Checks the SystAuth catalog
  - (B) Decides whether
    - the box cannot be inserted
    - the box can be inserted
    - the box must be modified: **SECURE OPERATORS**

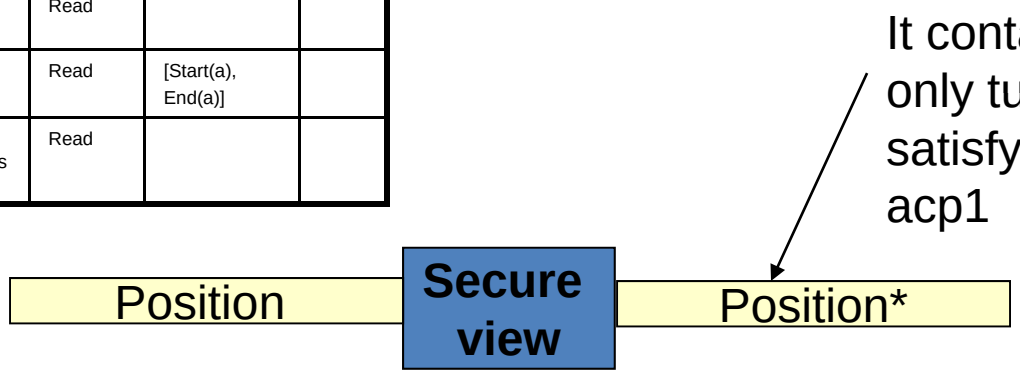
Secure operators

- Secure view operator:

  `Sec_View(S,acp) = Map(att)(Filter(P)(S))`

Sub	Protection object spec			Priv	GTC	WTC
	STRs	ATTs	EXPs			
Captain	Position	Pos, SID	Position.Platoon = self.Platoon	Read		
Captain	Position	Pos, SID	Pos > k	Avg	[Start(a), End(a)]	[1, 1]
Doctor	Health	Heart, SID	Health.Platoon = self.Platoon	Read		
Doctor	Health	Heart, SID	Health.Platoon ≠ self.Platoon	Read	[Start(a), End(a)]	
Doctor	Health, Position	Heart, SID	Position.SID = Health.SID AND Pos < k2	Read		

`Sec_View(Position,acp1)`

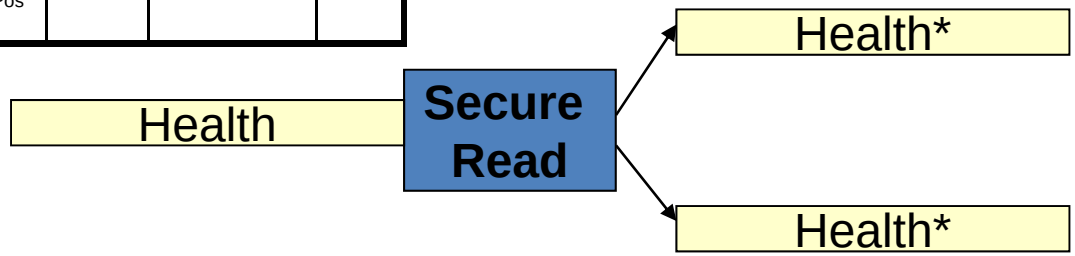


Secure operators

- Secure Read operator
  - $\texttt{Sec\_Read(S,u)} = \bigcup_{acp_j \in Pol(S,u)} \texttt{Sec\_View(S,}acp_j\texttt{)}$

Sub	Protection object spec			Priv	GTC	WTC
	STRs	ATTs	EXPs			
Captain	Position	Pos, SID	Position.Platoon = self.Platoon	Read		
Captain	Position	Pos, SID	Pos > k	Avg	[Start(a), End(a)]	[1, 1]
Doctor	Health	Heart, SID	Health.Platoon = self.Platoon	Read		
Doctor	Health	Heart, SID	Health.Platoon ≠ self.Platoon	Read	[Start(a), End(a)]	
Doctor	Health, Position	Heart, SID	Position.SID = Health.SID AND Pos < k2	Read		

- `Bob is a doctor`
- `Sec_Read(Health,bob)`



Secure operators

- Secure Aggregate operator
  - $\texttt{Sec\_Aggr(S,Op,s,i,u)} = \bigcup_{acp_j \in Pol_{agg}(S,u)} \texttt{Aggregate(Op,}max_{size}\texttt{,}max_{step}\texttt{)(Map(Op.A)(Filter(P)(S))}$

Sub	Protection object spec			Priv	GTC	WTC
	STRs	ATTs	EXPs			
Captain	Position	Pos, SID	Position.Platoon = self.Platoon	Read		
Captain	Position	Pos, SID	Pos > k	Avg	[Start(a), End(a)]	[1, 1]
Doctor	Health	Heart, SID	Health.Platoon = self.Platoon	Read		
Doctor	Health	Heart, SID	Health.Platoon ≠ self.Platoon	Read	[Start(a), End(a)]	
Doctor	Health, Position	Heart, SID	Position.SID = Health.SID AND Pos < k2	Read		

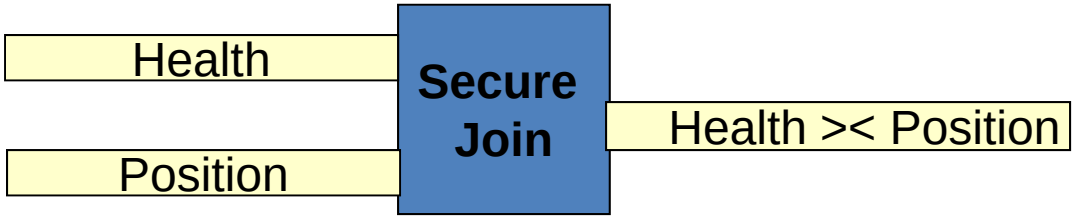
- `Alice is a captain`
- `Sec_Aggr(Position,(AVG,Pos), 1,1, Alice)`

Secure operators

- Secure Join operator
  - `Sec_Join(S1,S2,P,u) = Sec_Read(Join(P)(S1,S2),u)`

Sub	Protection object spec			Priv	GTC	WTC
	STRs	ATTs	EXPs			
Captain	Position	Pos, SID	Position.Platoon = self.Platoon	Read		
Captain	Position	Pos, SID	Pos > k	Avg	[Start(a), End(a)]	[1, 1]
Doctor	Health	Heart, SID	Health.Platoon = self.Platoon	Read		
Doctor	Health	Heart, SID	Health.Platoon ≠ self.Platoon	Read	[Start(a), End(a)]	
Doctor	Health, Position	Heart, SID	Position.SID = Health.SID AND Pos < k2	Read		

- `Bob is a doctor`
- `Sec_Join(Health,Position, Position.SID=Health.SID,Bob)`

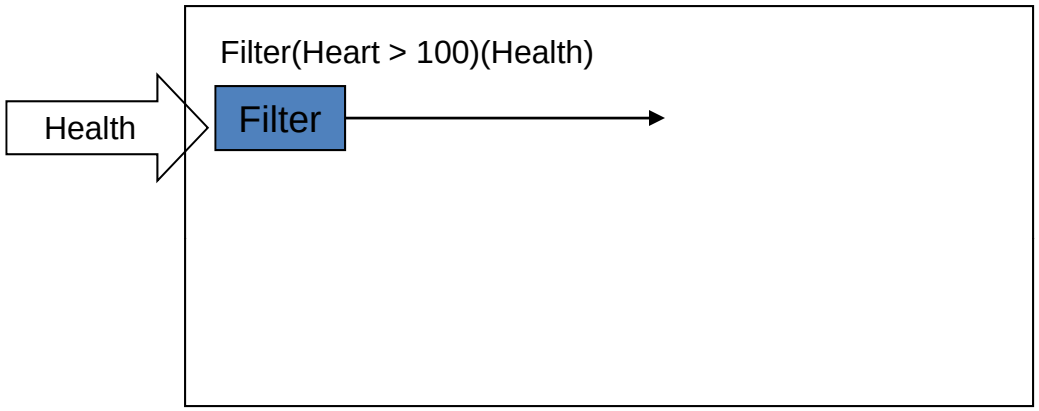


Access control enforcement

- Access control algorithm:
  - Input: Access request=(u, Objs, p)
  - Output: set of Aurora expressions generating the authorized stream

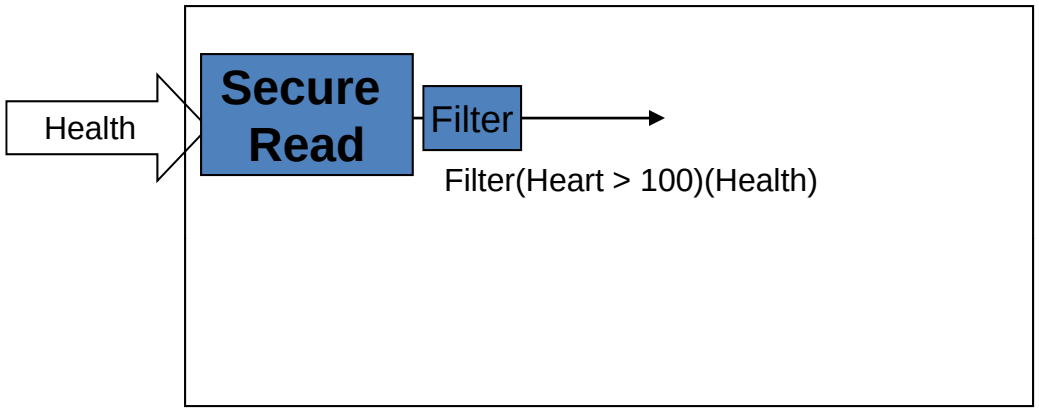
- Formal proof: algorithm correctness

Access control enforcement



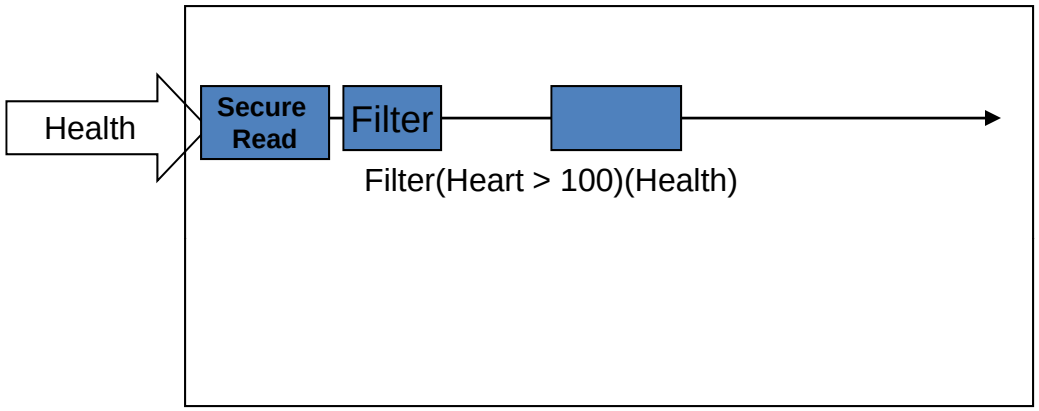
- When the box is applied on input stream:

Access control enforcement:



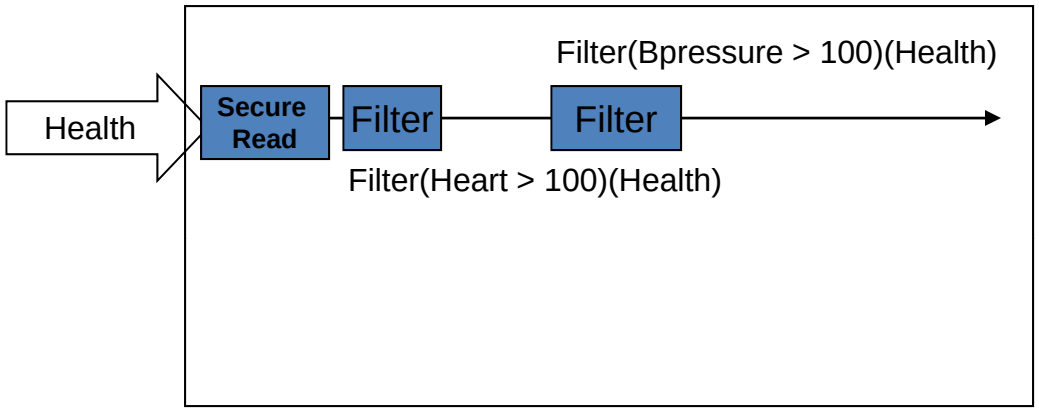
- When a box is applied on input stream(s):
  - secure operators:
    - if box ≠ Aggregate or Join: secure read
    - otherwise: secure aggregate or secure join

Access control enforcement



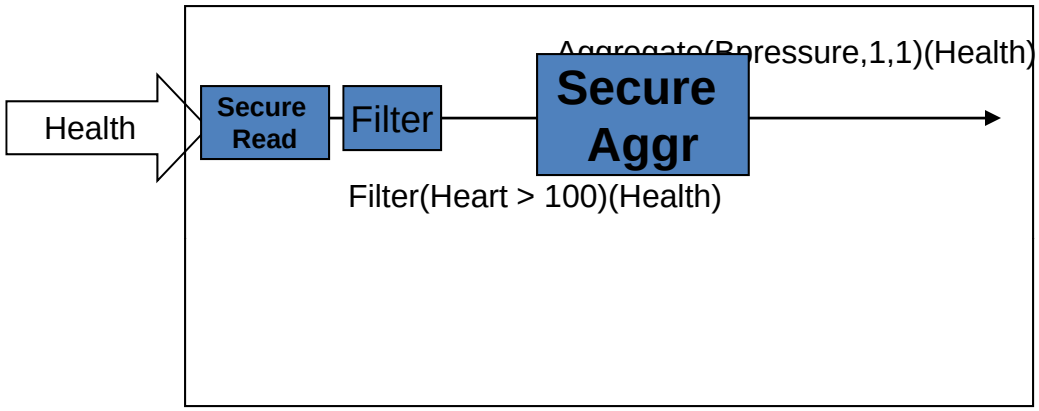
- When a box is applied on internal stream(s):

Access control enforcement



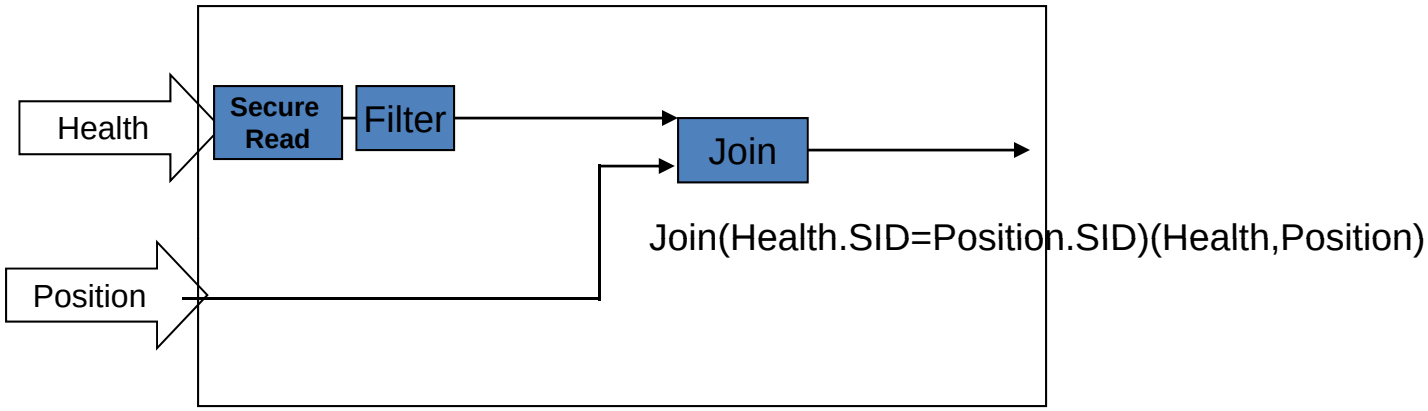
- When the box is applied on internal stream:
  - If box ≠ Aggregate or Join: leaves as it is

Access control enforcement



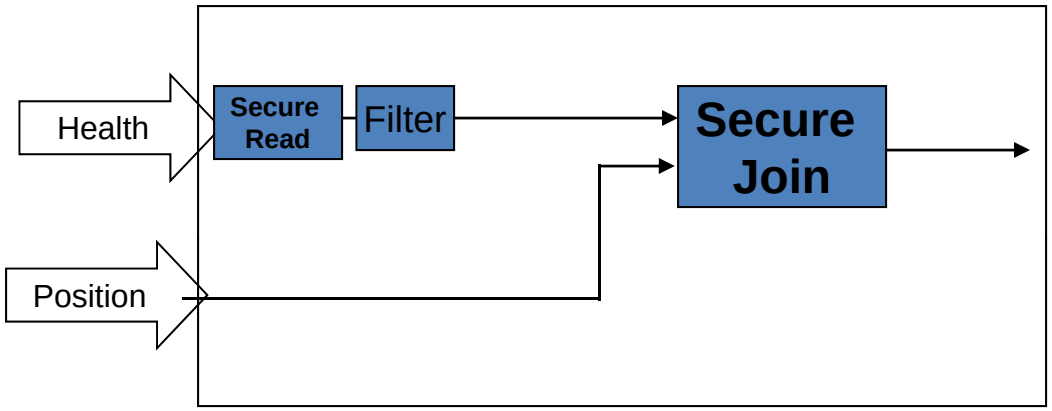
- When the box is applied on internal stream:
  - Otherwise:
    - Secure aggregate operator
    - Secure join operator

Access control enforcement



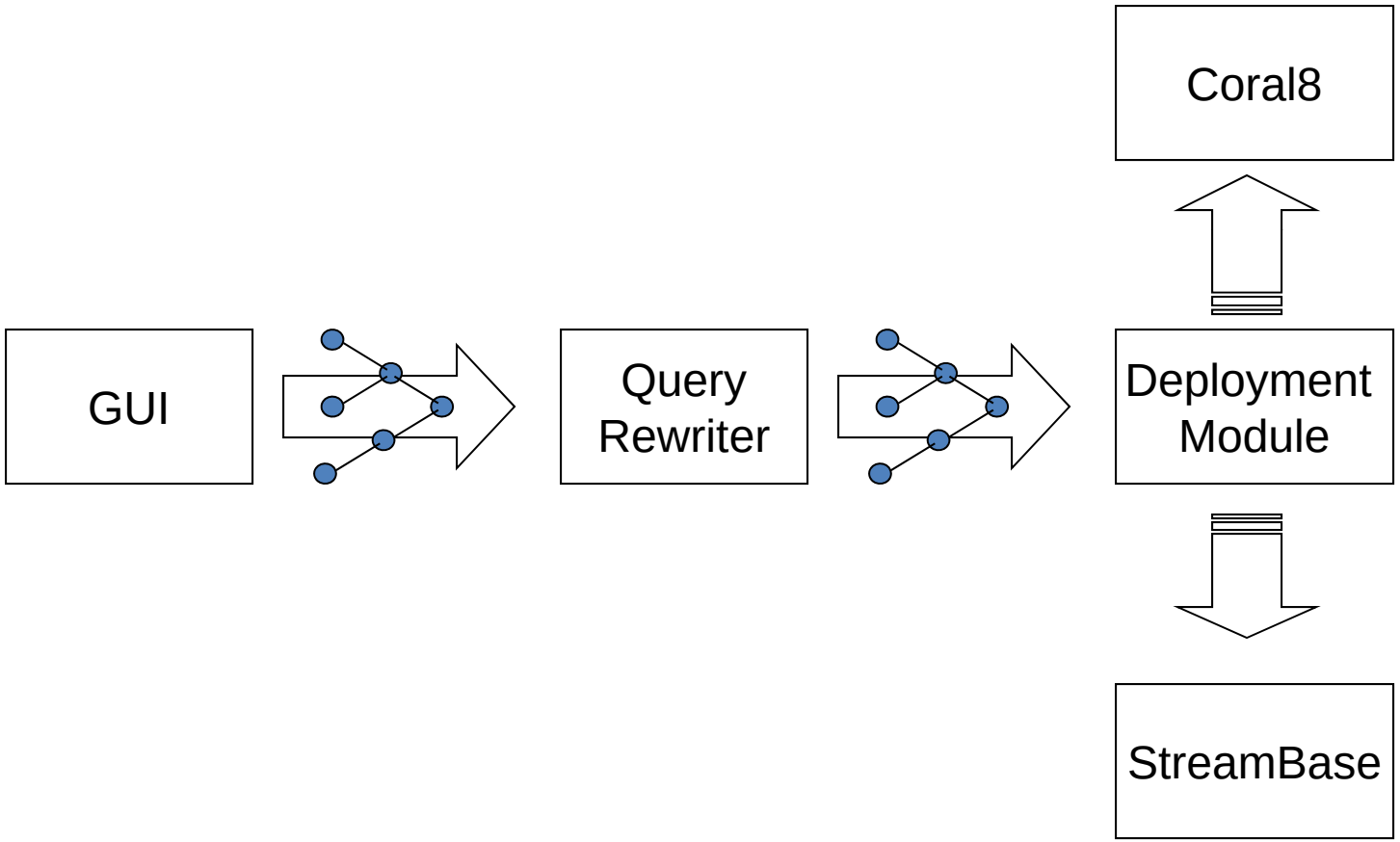
- When the box is applied on internal stream:
  - Otherwise:
    - Secure aggregate operator
    - Secure join operator

Access control enforcement



- When the box is applied on internal stream:
  - Otherwise:
    - Secure aggregate operator
    - Secure join operator

Prototype Implementation



Outline

- Emerging IT trends
  - Sensors, Web
- Cloud, SensorCloud & Challenges
- Preliminary work done at NUS
  - epiC
  - HipCloudS
- The NExt Center
- Conclusion

NExT Search Center

NUS-Tsinghua

Extreme Search Center

NExT Center

- Joint research efforts between NUS & Tsinghua University
  - A 5-year, multi-million collaborative project supported by NRF in Singapore
  - With equivalent contributions from both Universities
  - Centers in both Singapore and Beijing
  - Involve over 50 professors/ researchers/ PhD students

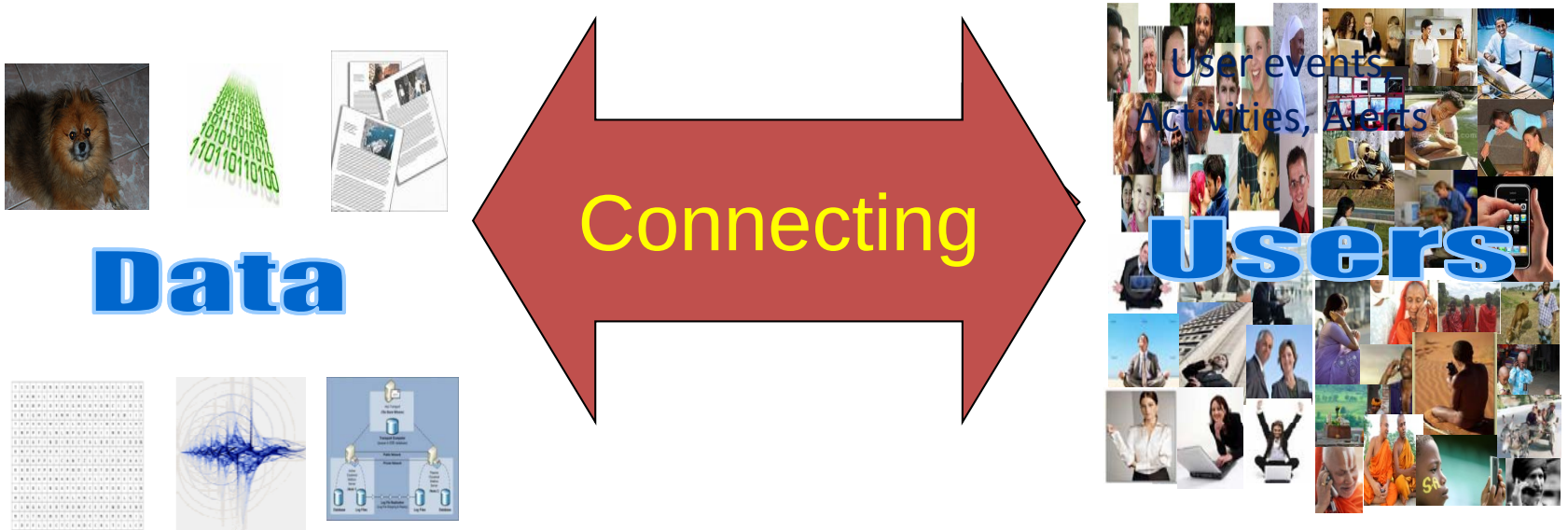
Aims of NExT Center

- To find and extract meanings from millions of real-time data streams

  data → Information

- To aggregate info to realize a SMART environment

  Information → Situational Info → Users



NExT Center

- Research into Internet-Scale Extreme Search for millions of Real-time Data and Sensors to help realize Smart Living for All
  - Aggregate, track and predict events in a location
  - Inform users of latest happenings at all time
  - Help users in their daily activities

- Activities.. Conduct research on:

EXTREME DATABASES

Outline

- Emerging IT trends
  - Sensors, Web
- Cloud, SensorCloud & Challenges
- Preliminary work done at NUS
  - epiC
  - HipCloudS
- The NExt Center
- Conclusion

Conclusion

- Sensors are here to stay
- The past, present, future
  - Descriptive, predictive, prescriptive
- Cyber-Physical systems are becoming increasing important
- What's the future like?
  - Attend the panel session this afternoon …

References

[1] Towards a Privacy-aware Stream Data Management System for Cloud Applications. W.S. Ng, M. Kirchberg, S. Bressan, K.L. Tan, *The Supercomputing Journal*, Springer, 2010.

[2] A Framework to Enforce Access Control over Data Steams. B. Carminati, E. Ferrari, J. Cao, K.L. Tan, *ACM Transactions on Information and System Security* . Vol 13, No. 3, 2010.

[3] Web Squared: Web 2.0 Five Years On. T.O'Reilly, J. Battelle, Web2.0 Summit, 2010.

[4] SensorCloud: Towards Sensor-Enabled Cloud Services. H.B. Lim, 2009

[5] Efficient B-Tree Based Indexing for Cloud Data Processing. S. Wu, D. Jiang, B.C. Ooi, K.L. Wu, PVLDB 2010.

[6] A Cloud Data Storage System for Supporting Both OLTP and OLAP
Yu Cao, Chun Chen, Fei Guo, Dawei Jiang, Yuting Lin, Beng Chin Ooi, Hoang Tam Vo, Sai Wu and Quanqing X. Technical Report, National University of Singapore, School of Computing. TRA8/10, 2010

Thank you for your attention!