A Middleware for Data-centric and Dynamic Distributed Complex Event Processing for IoT Real-time Analytics in the Cloud

Gustavo B. Baptista, Felipe Carvalho, Sergio Colcher and Markus Endler

Department of Informatics – Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rio de Janeiro, Brazil.
{gbaptista,fcarvalho,colcher,endler}@inf.puc-rio.br

LAC
Laboratory for Advanced Collaboration

DEPARTAMENTO DE INFORMÁTICA
PUC·RIO

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

Introduction

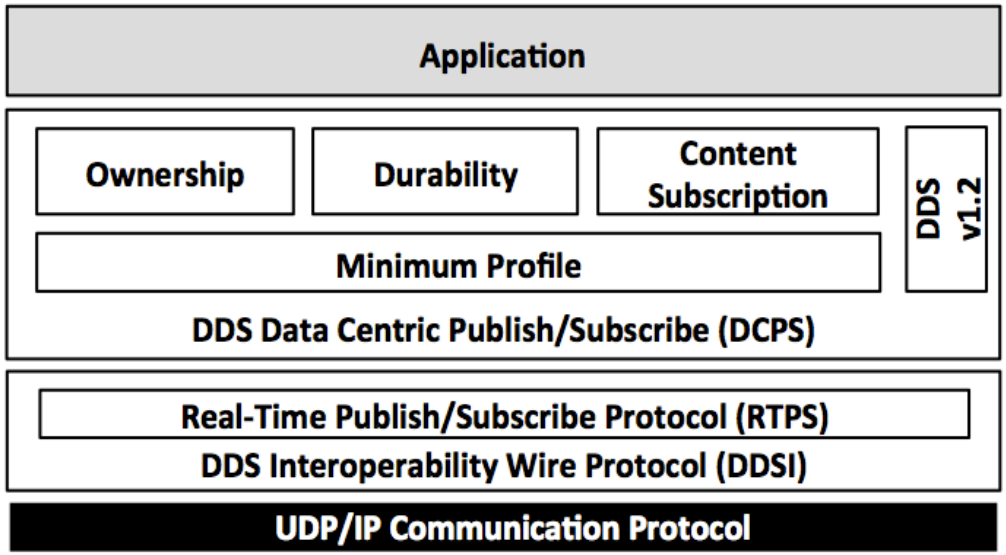
- IoT big data real-time analytics systems
  - Massive amounts of data
  - Streams produced by distributed data sources
- Reactive Paradigm
  - Distributed CEP Systems are very suitable
  - Challenges in deploying and managing processing logic at execution time
  - 24x7 availability

Data-centric Paradigm

- ***Data-centric Paradigm***
  - *T*he mean of interaction is data. (vs. message-centric, is the *message*)
  - The middleware
    - Has the definition of structure and data
    - Aware of contents (i.e. instances) of structures
    - Imposes rules on structures, changes and access
    - Manages distributed state
  - **Data-centric Publish-Subscribe (DCPS)**
    - Global Shared Data Space
    - Logical decentralized space maintained by all peers
    - Contains the structure and instances of data
    - Nodes read/write data
    - Infrastructure ensures all participants a consistent and up-to-date view

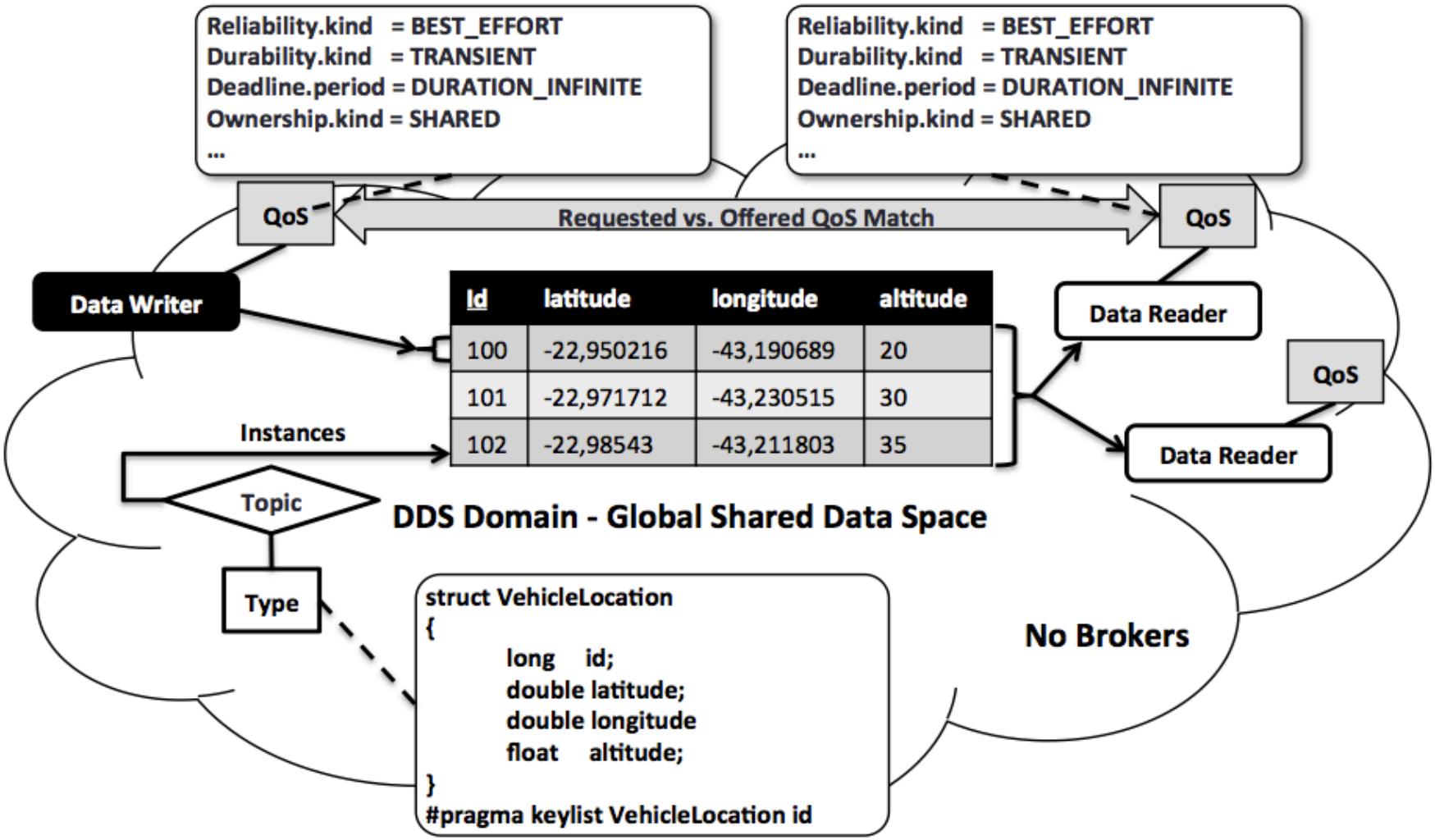
Data-Centric Paradigm

- **Data Distribution Service for Real-Time Systems (OMG-DDS)**
  - Fully distributed peer-to-peer (i.e. broker-less)
  - Real-time data-centric publish/subscribe
  - High performance communication, scalability and availability
  - Specification of Quality of Service (QoS) contracts
  - Mechanisms for dealing with real-time aspects
  - Priority and other specific QoS policies
  - Interoperability across
    - DDS implementations
    - Programming languages
    - Platforms
  - Automatic discovery



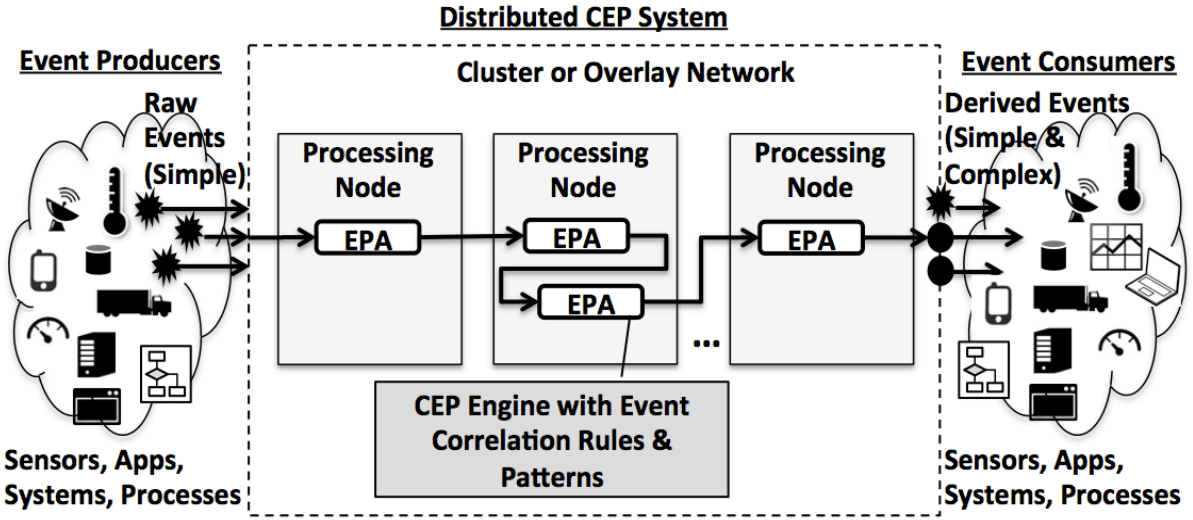
Data-Centric Paradigm

- Data Distribution Service for Real-Time Systems (OMG-DDS)



D3CEP Middleware

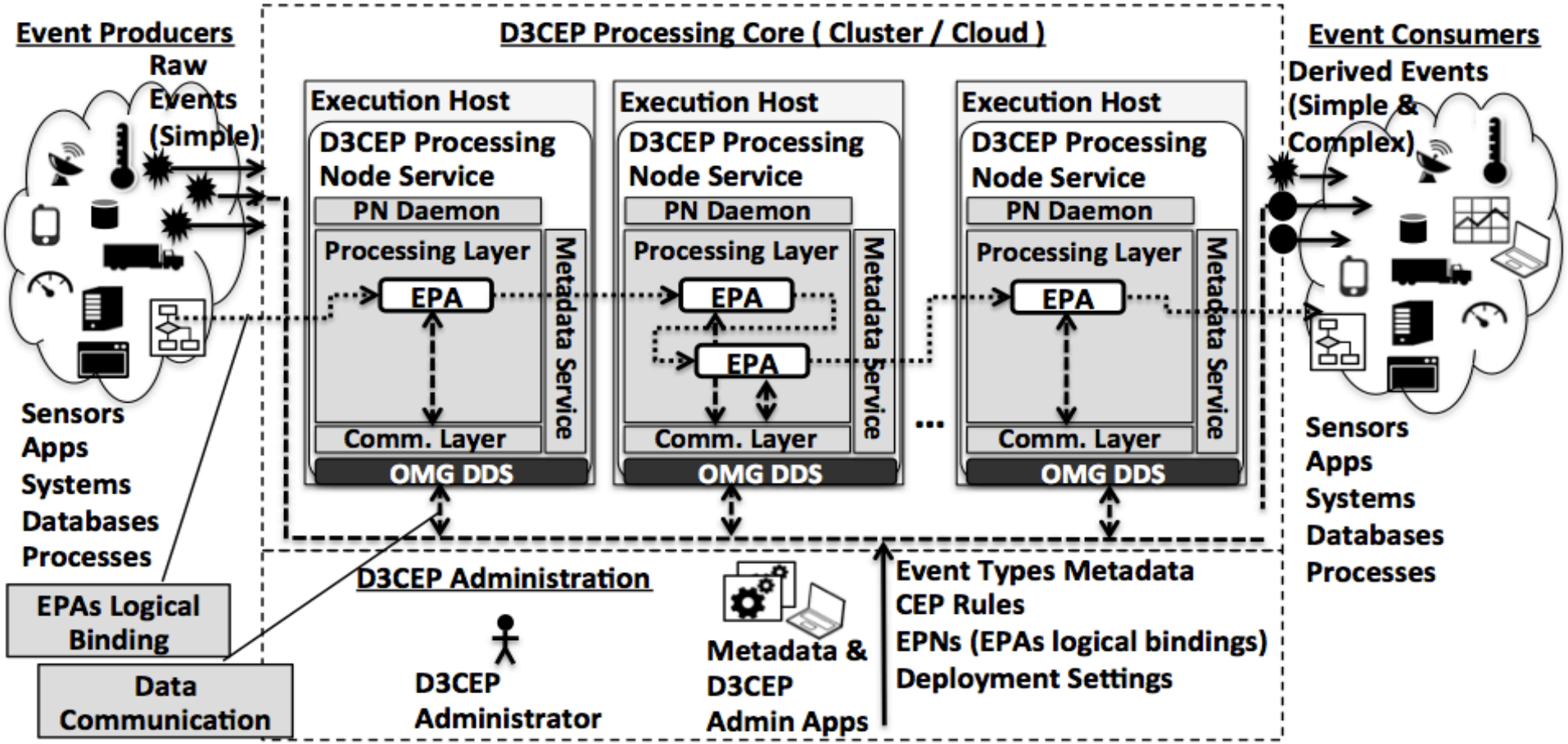
- We present a middleware for Distributed CEP
  - Benefits of Data-centric and dynamic design approach
    - Dynamic definition and deployment of CEP rules
    - Peer-to-peer routing of events among CEP rules
    - Reduced coupling of producers, consumers and CEP rules
    - Availability provided by peer-to-peer model
    - High throughput and low latency in communication & detection
    - Esper as a CEP engine at each node
  - Architecture and tests regarding performance and scalability



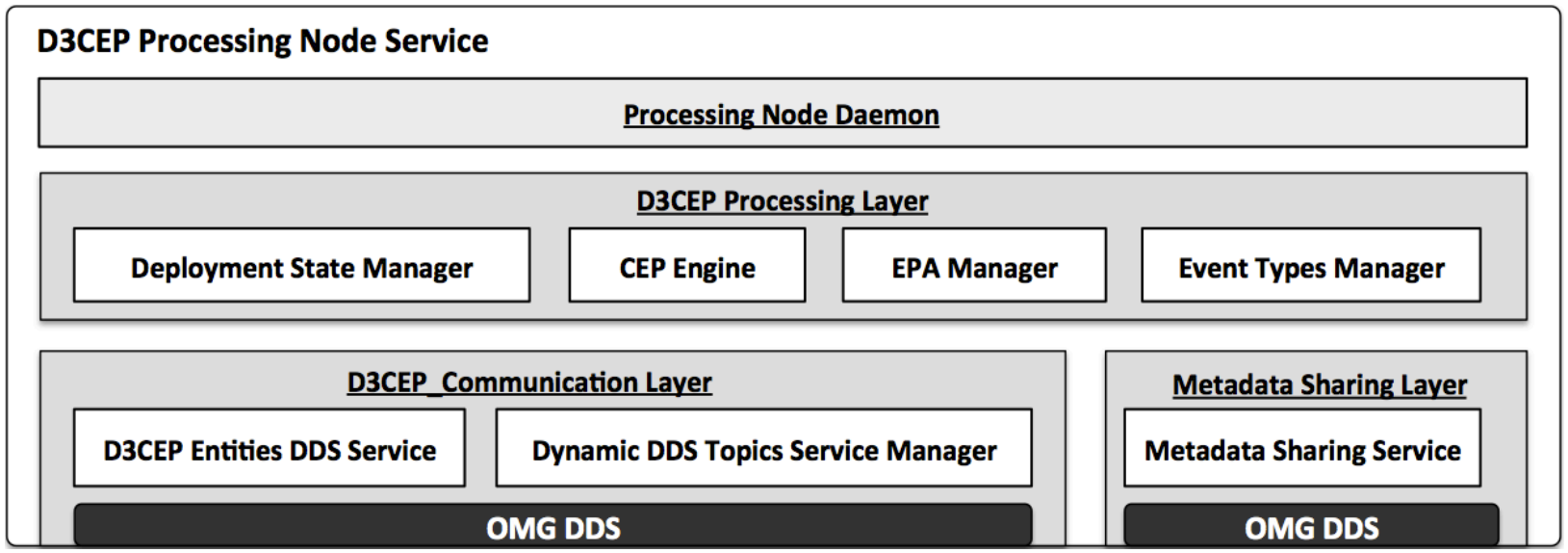
D3CEP Middleware

- Global Shared ***Reactive*** Data Space
  - Data dissemination and reactive behavior modeled together
    - Description of events at processing and communication layers
  - Consumers, producers and CEP rules
    - Dynamically defined and deployed seamlessly
  - Additional mechanisms:
    - Global Catalog of event types
    - Global definition of DCEP entities and deployment
    - EPAs, EPNs, CEP Rules
    - CEP services

D3CEP Middleware

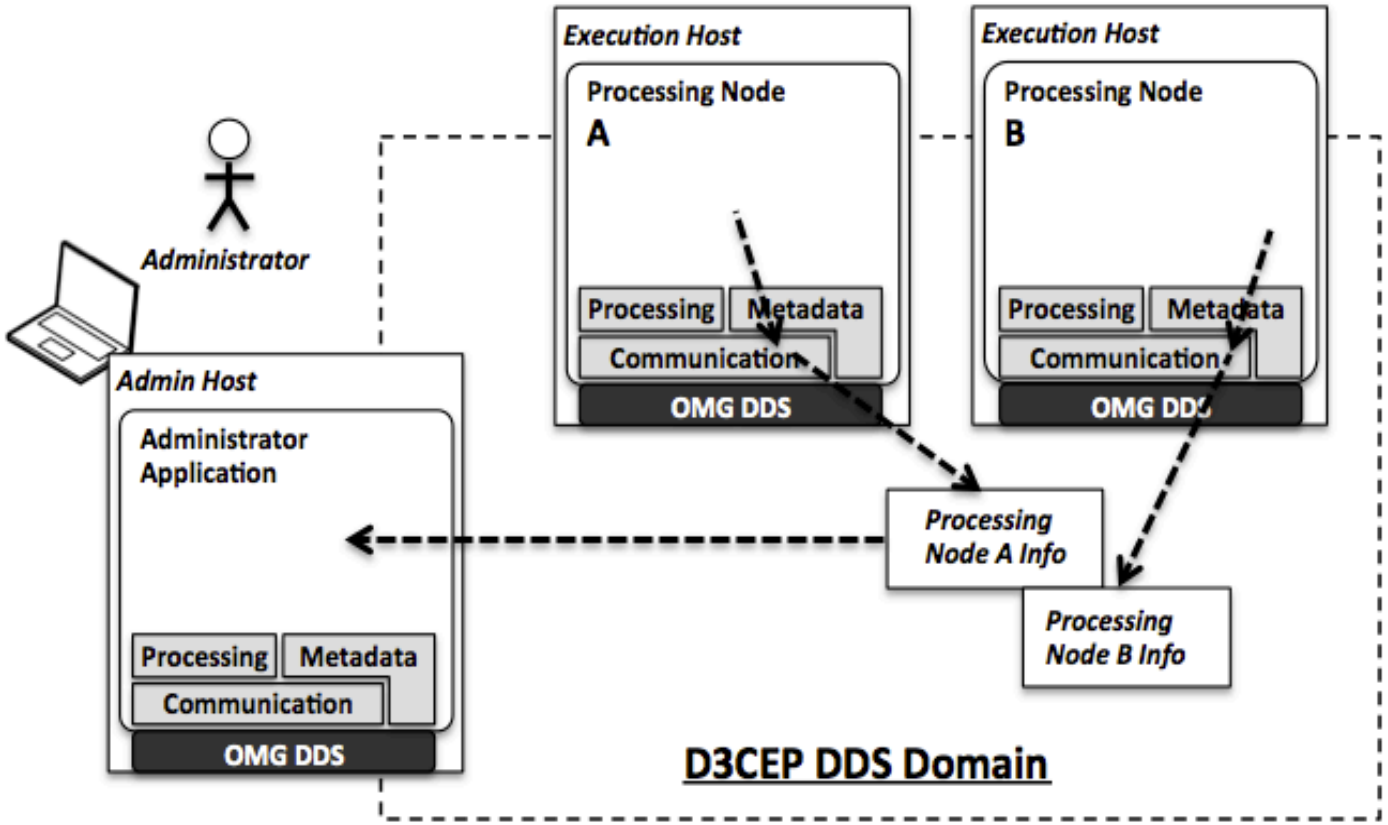


Architecture



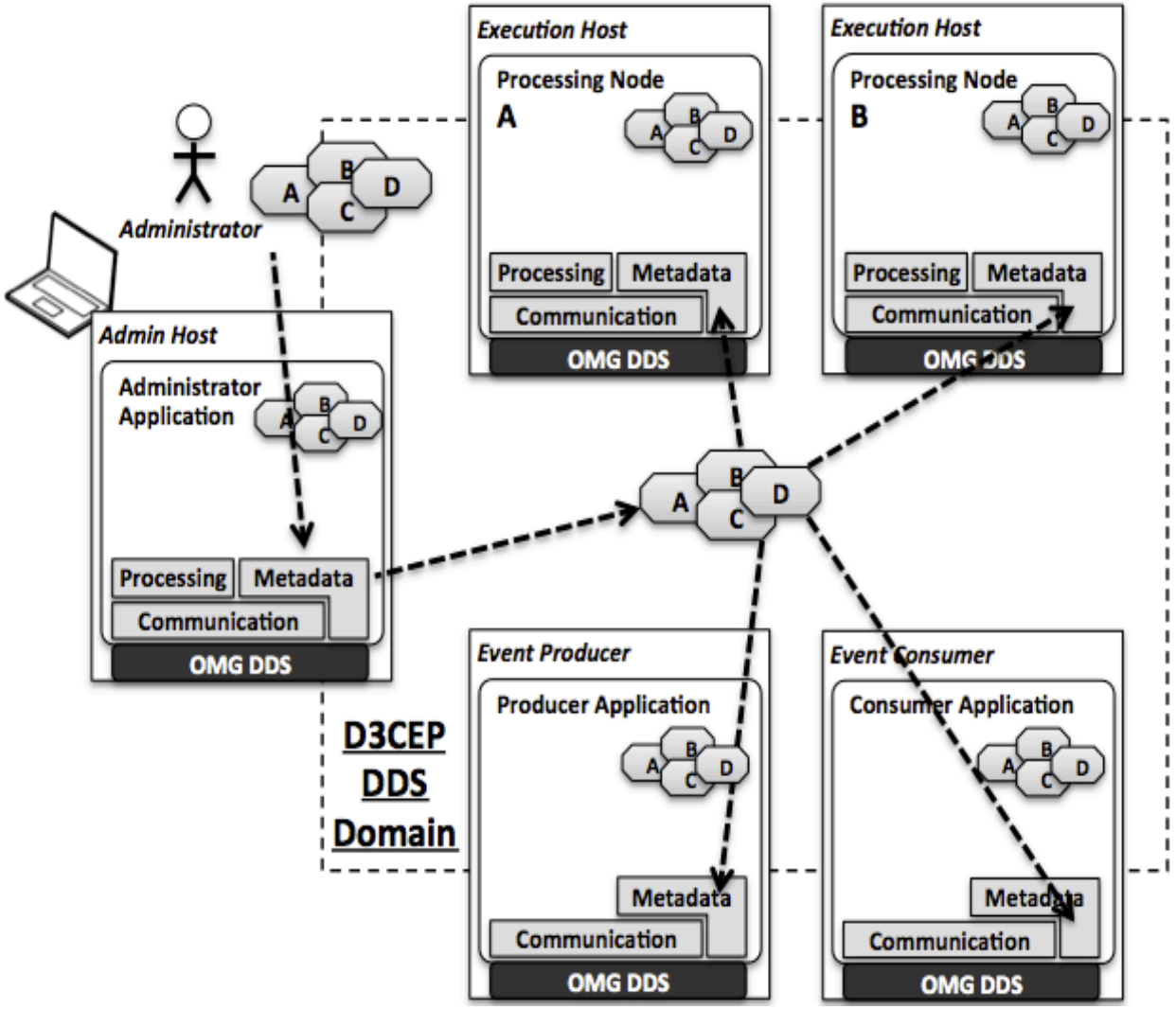
Architecture

- *Processing Node Daemon*



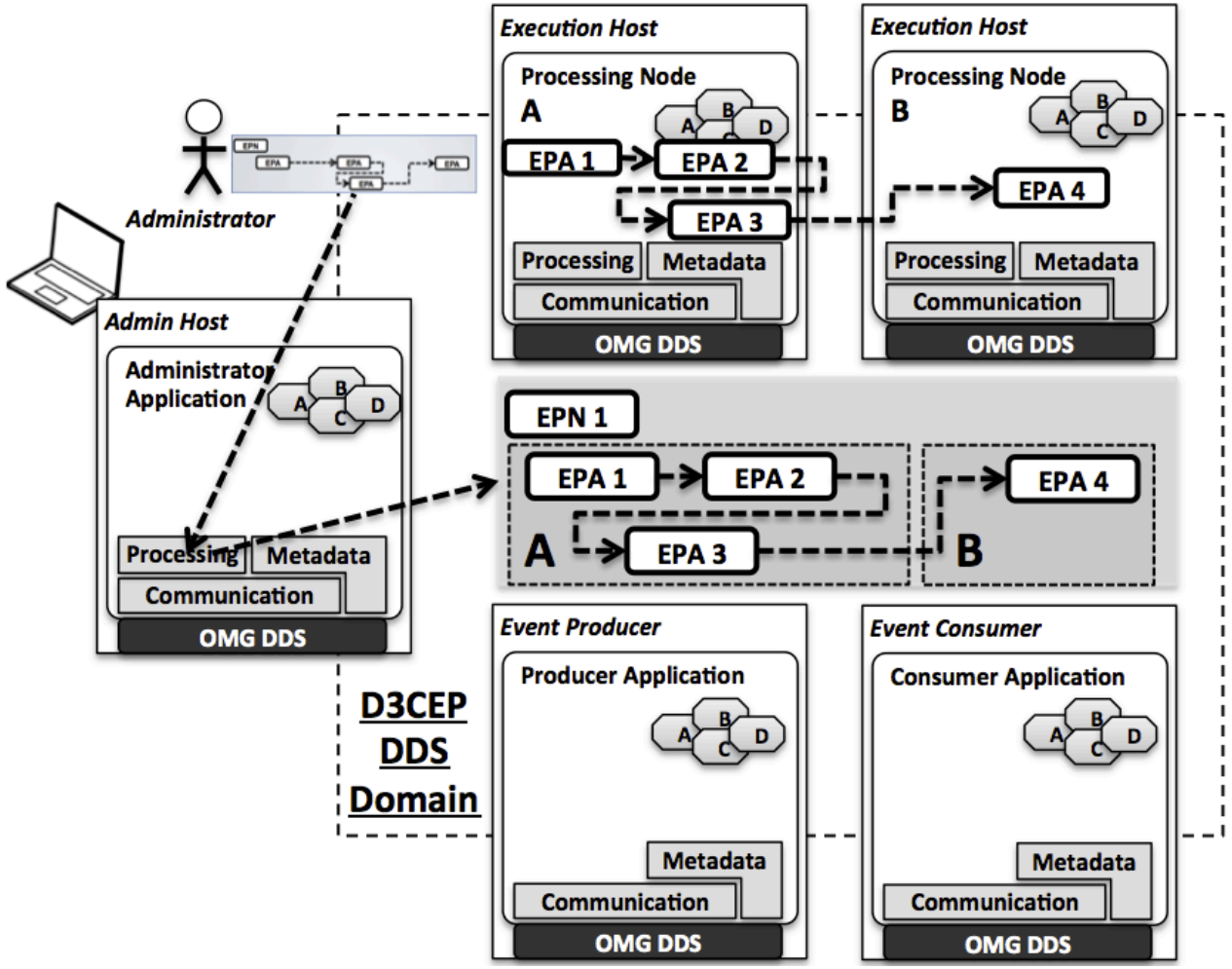
Architecture

- *Metadata Sharing Service*



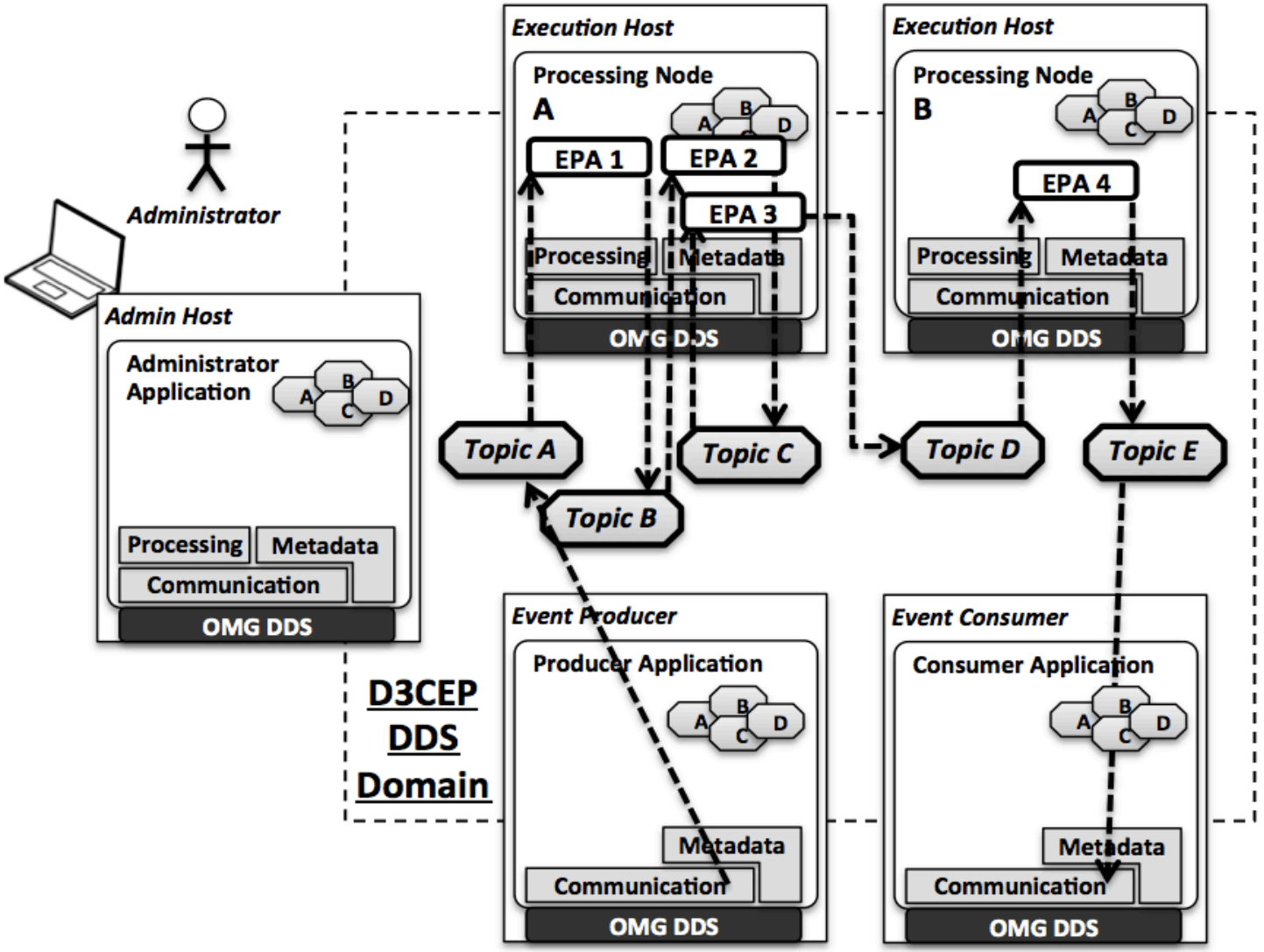
Architecture

- *D3CEP Administration*



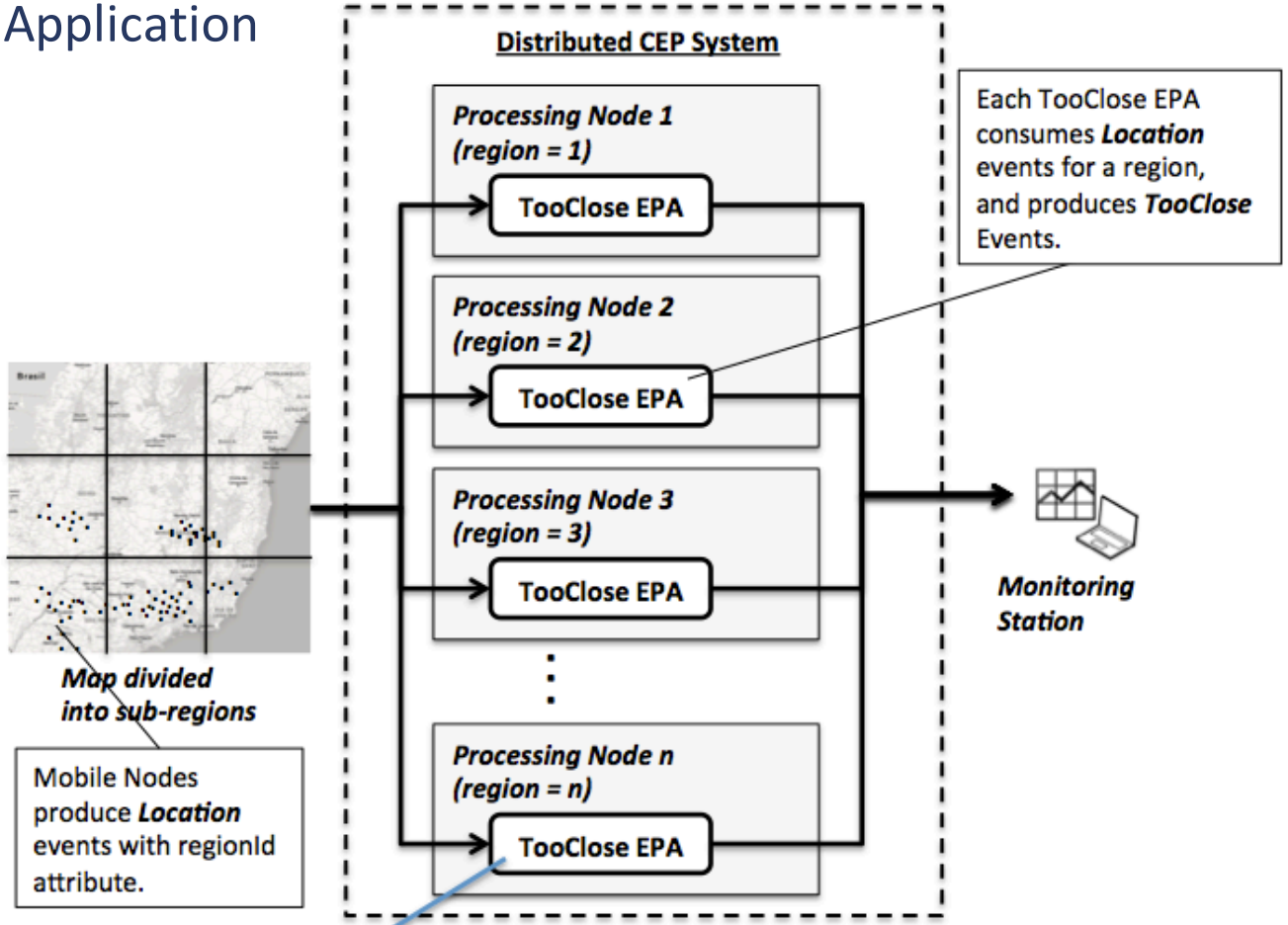
Architecture

- *Dynamic DDS Topics Service*



Use Case

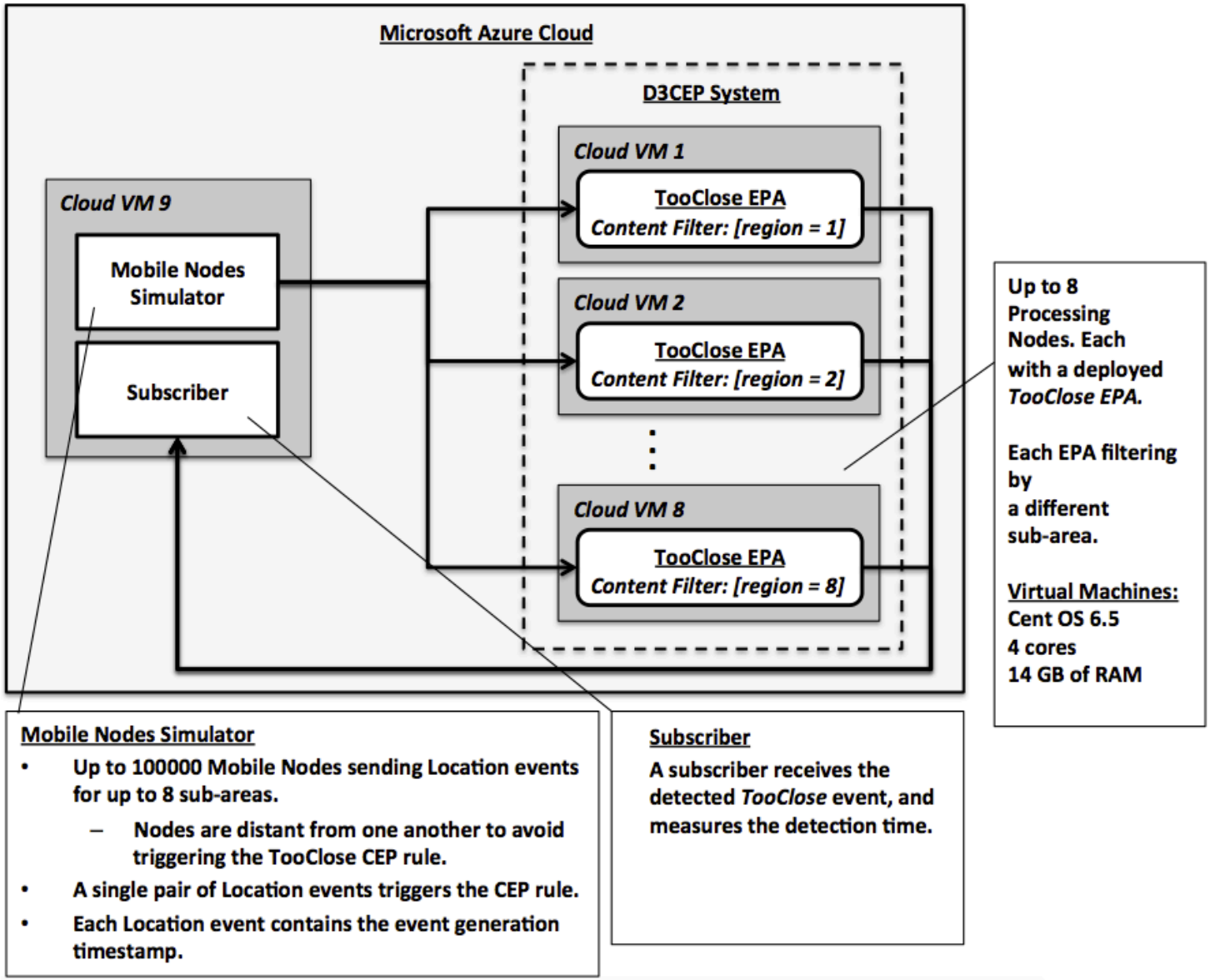
- Telemetry Application




```
Too Close EPA
INSERT INTO TooClose
SELECT A, B
FROM Location.win:time_batch(30 sec) as A, Location.win:time_batch(30 sec) as B
WHERE A.nodeId <> B.nodeId AND
distance(A.latitude, A.longitude, B.latitude, B.longitude) < D + 2ε
```

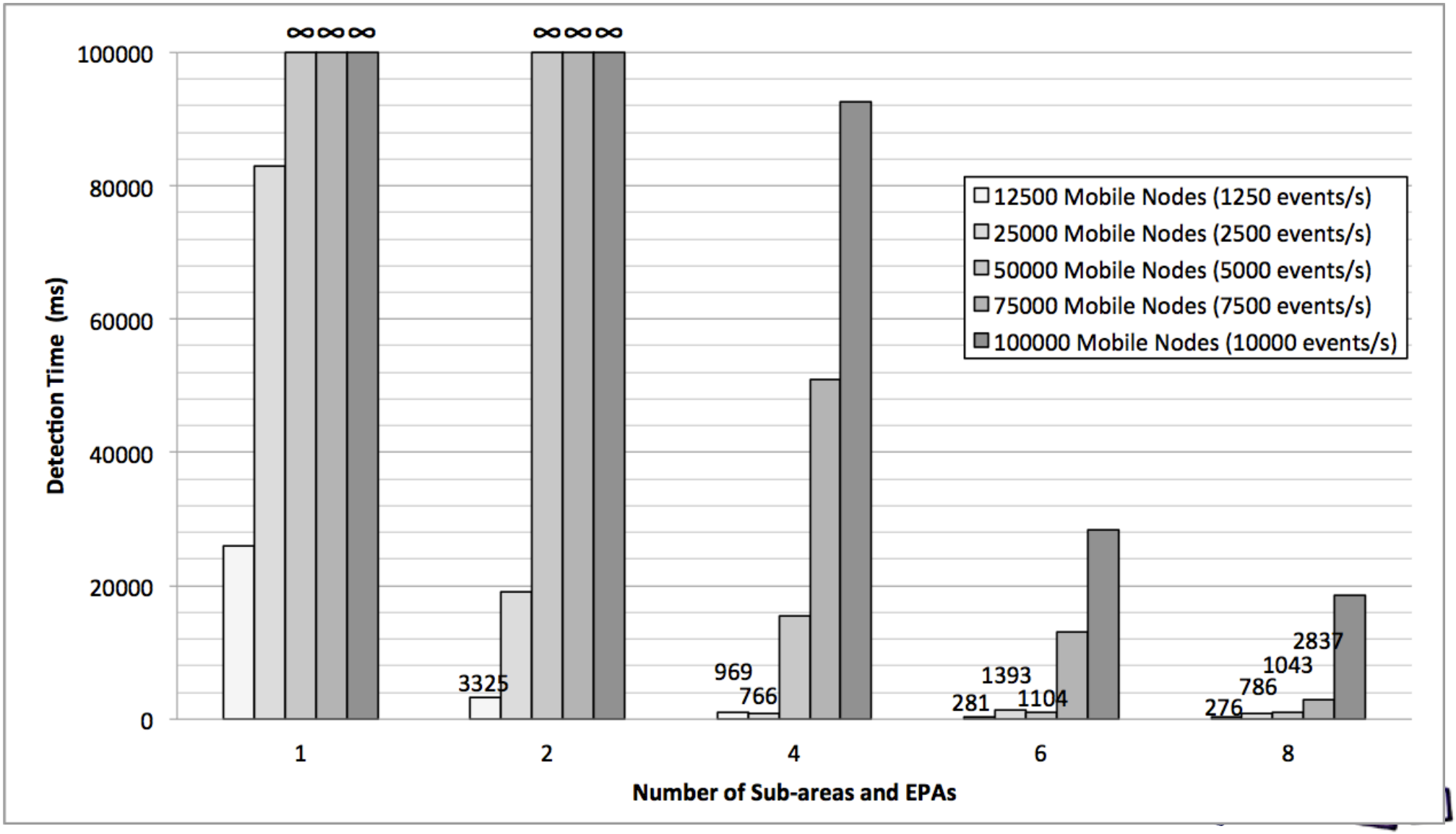
Evaluation

- Detection Time



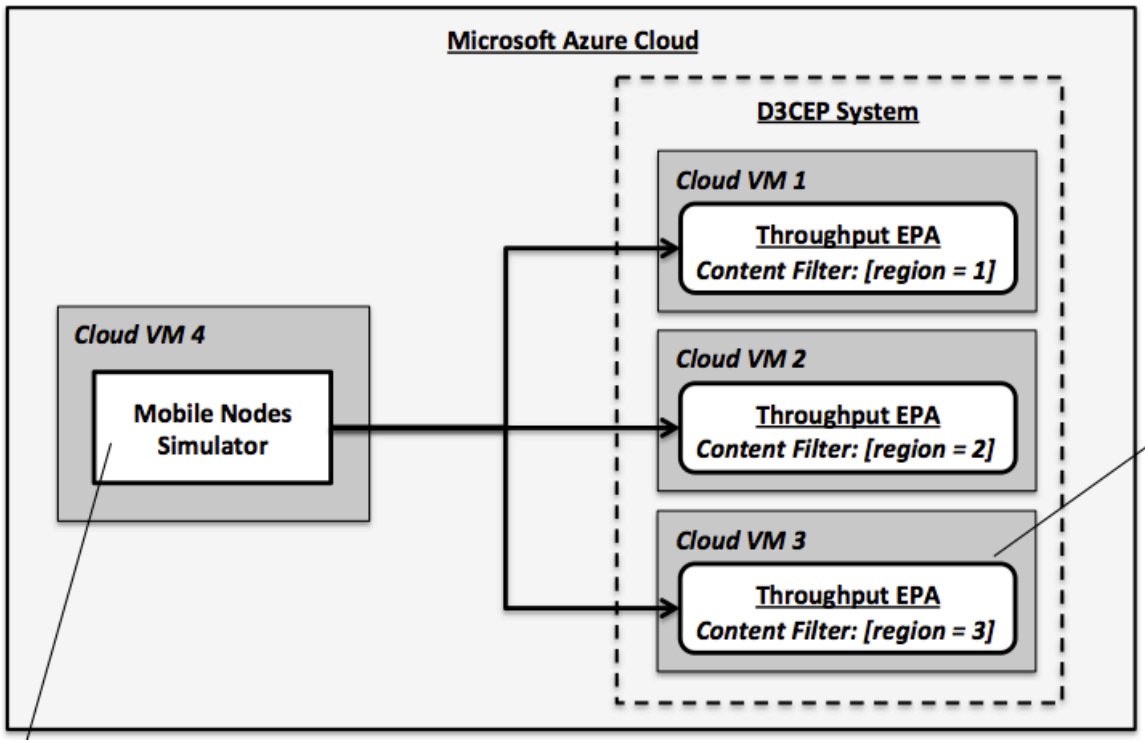
Evaluation

- Detection Time



Evaluation

- Throughput



3 Processing Nodes. Each Throughput EPA measuring the receiving rate of events.

Each EPA filtering by a different sub-area.

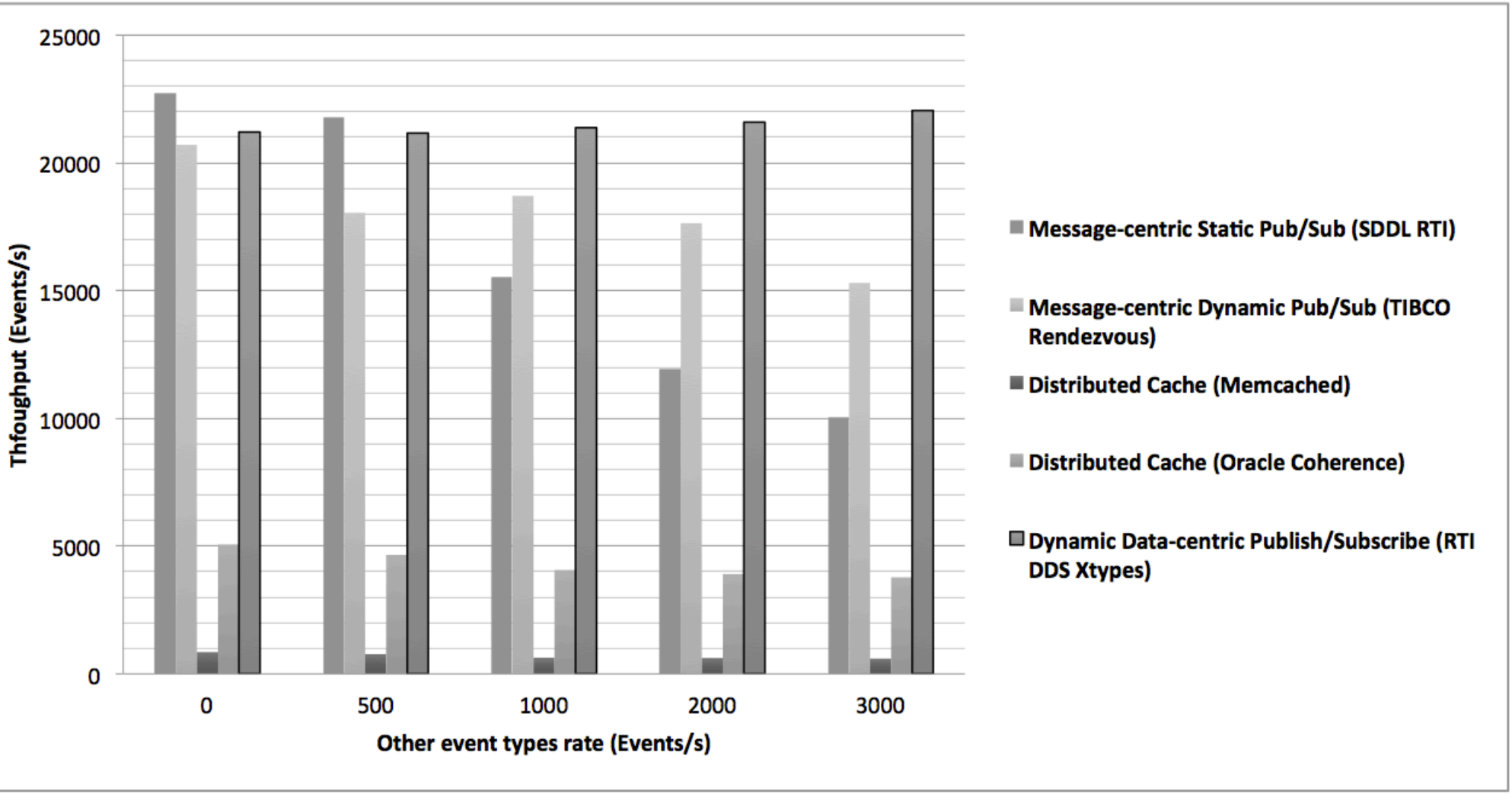
Virtual Machines:
Cent OS 6.5
4 cores
14 GB of RAM

Mobile Nodes Simulator

- Up to 240000 Mobile Nodes sending Location events for 3 sub-areas.
- Sending of alternative event types, at a rate of up to 3000 events/s
- Communication paradigms/technologies tested:
  - Message-centric (SDDL RTI Connext)
  - Dynamic Data-Centric Pub/Sub (RTI Connext DDS Xtypes)
  - Distributed Cache (Memcached)
  - Distributed Cache (Oracle Coherence)

Evaluation

- Throughput

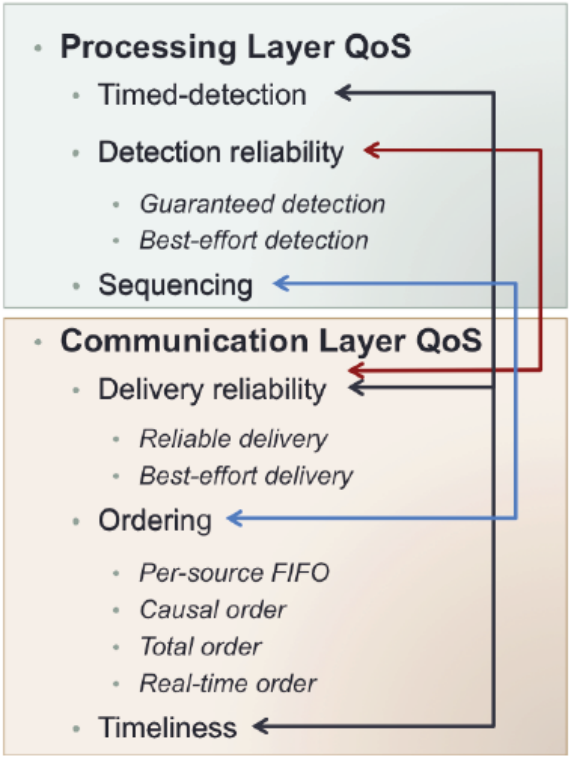


Related Work

Related Work	Integration Paradigm	Communication Technology	Dynamic
OpenSplice+Esper	Static Data-centric	OMG-DDS (OpenSplice DDS)	No
RTI+OracleCEP	Static Data-centric	OMG-DDS (RTI Connext)	No
RTI+Coral 8	Static Data-centric	OMG-DDS (RTI Connext)	No
Oracle CEP	**Dynamic Integrated**	**Dist. Cache (Oracle Coherence)**	**Yes**
Solar	**Dynamic Integrated**	**Dist. Hash Table (Pastry)**	**Yes**
D3CEP	**Dynamic Data-centric**	**OMG-DDS + XTypes (RTI Connext)**	**Yes**
TIBCO Business Events	**Dynamic Integrated**	**Prop. pub/sub (TIBCO Rendezvous)**	**Yes**

Conclusions and Future Work

- Main Contributions
  - Data-centric design approach to DCEP
    - Use of DDS for peer-to-peer routing of events
    - Dynamic Deployment and Automatic Discovery
      - Facilitates deployment of CEP rules
    - Distributed State Management
      - CEP rules seamlessly read/write Global Shared Data Space
- Future work
  - QoS contracts at the detection level [Appel et al.2010]
    - IoT and mission critical applications



Questions

Thank you