Data-Centric Publish/Subscribe and Cloud Computing Enablers for Industrial Internet

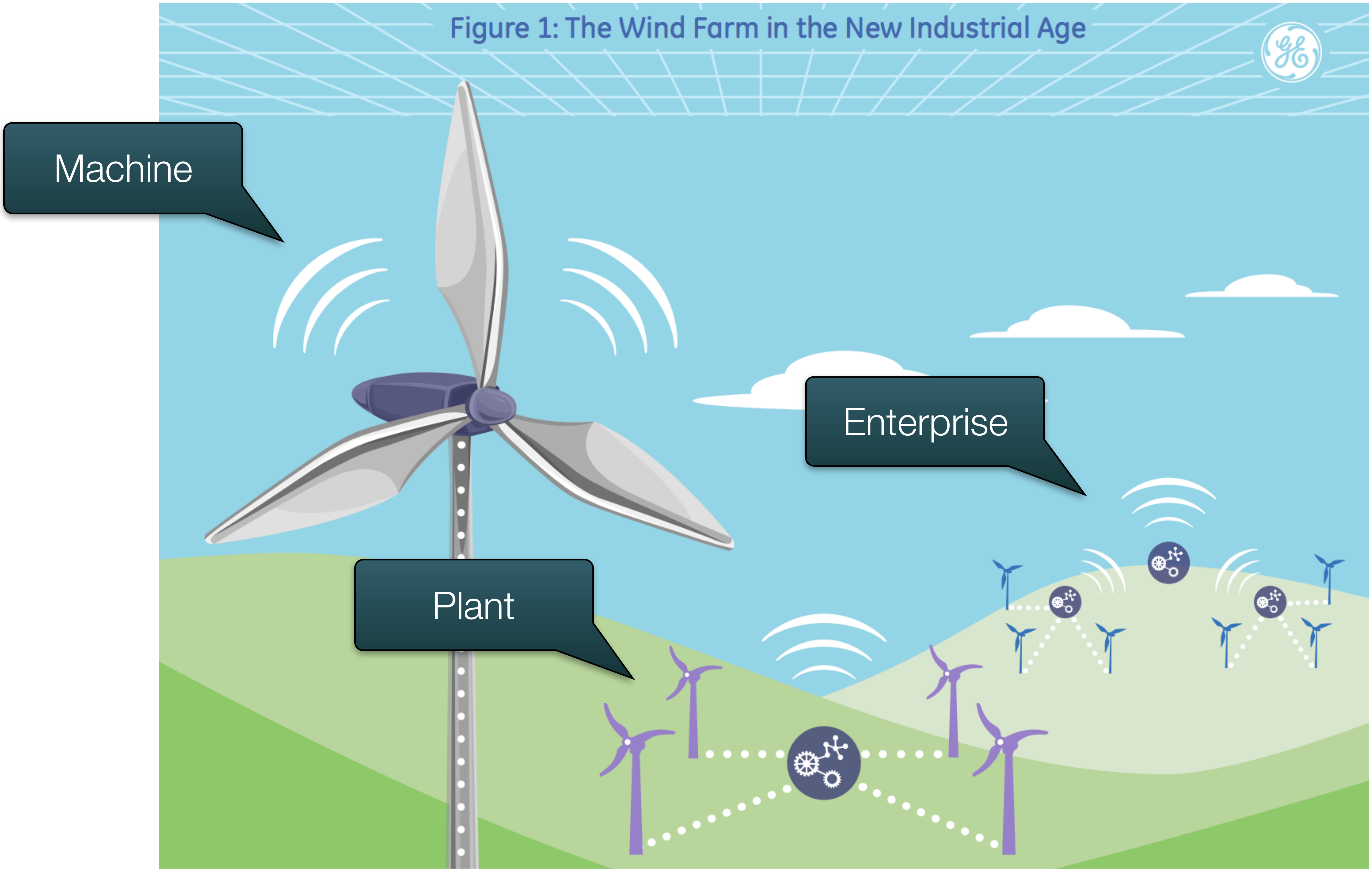
Qualifying Exam
October 2, 2014

Kyoungho An
Institute for Software Integrated Systems (ISIS)
Department of Electrical Engineering and Computer Science
Vanderbilt University
Nashville, Tennessee

Technology Trends: Industrial Internet

- Internet of Things (IoT) - Things hyper-connected over Internet realized by advances of networking, sensors, and embedded devices
- Collecting, sharing and analyzing data from connected things to provide intelligent services
- Industrial Internet - Focus on industry oriented and mission-critical applications such as Healthcare, Transportation, Manufacturing, Energy

3-Level Analysis of Industrial Internet



Reference from https://www.gesoftware.com/Industrial_Big_Data_Platform.pdf

3-Level Analysis of Industrial Internet

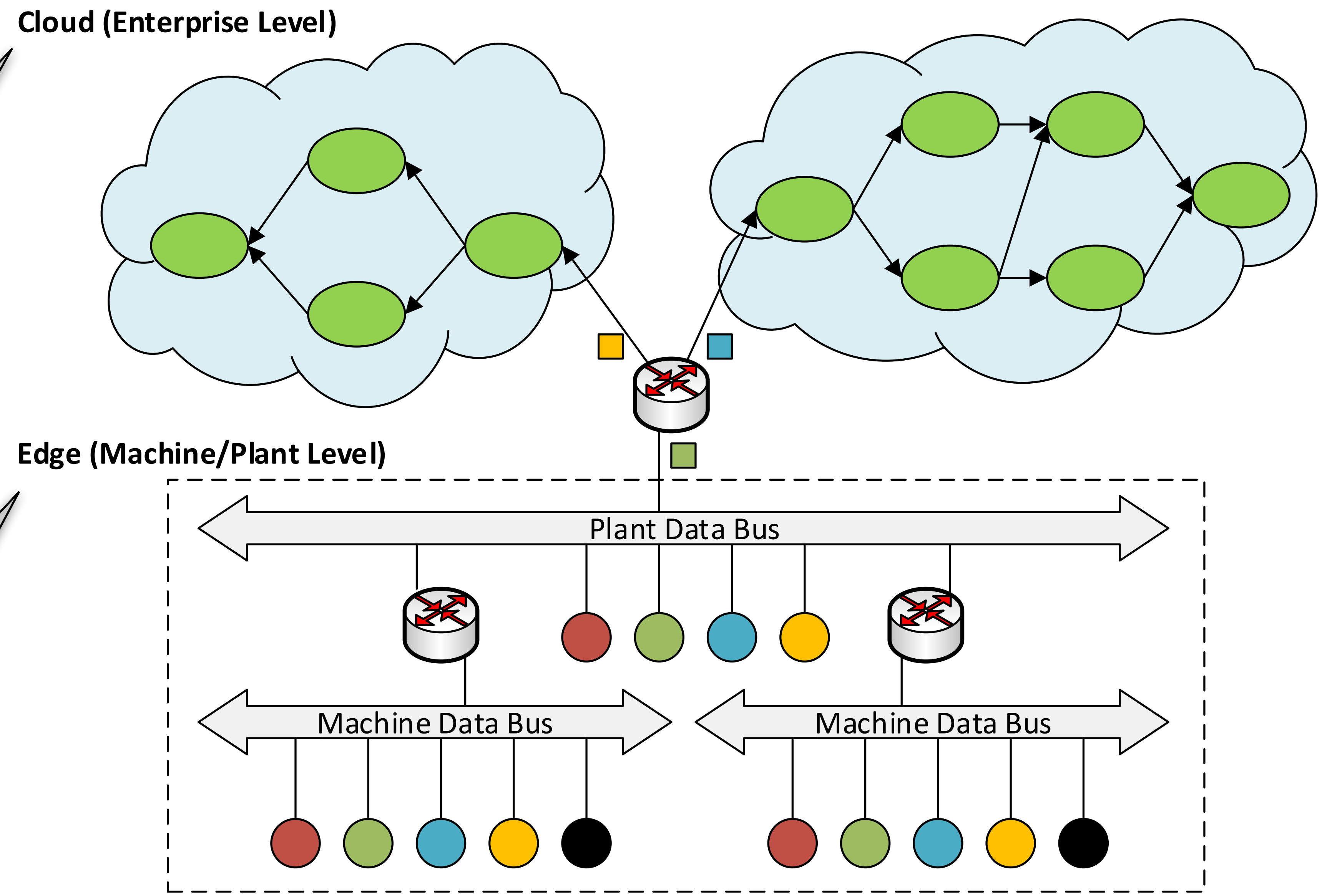
	Turbine (Machine)	Wind farm (Plant)	Power producer (Enterprise)
Analytics	Asset optimization	Operations optimization	Business optimization
Data Quantity	>100 tags	>6,000	>1,000,000 tags
Data Frequency	40 milliseconds	1 second	1 second - 10 minutes

Reference from https://www.gesoftware.com/Industrial_Big_Data_Platform.pdf

Motivational Architecture

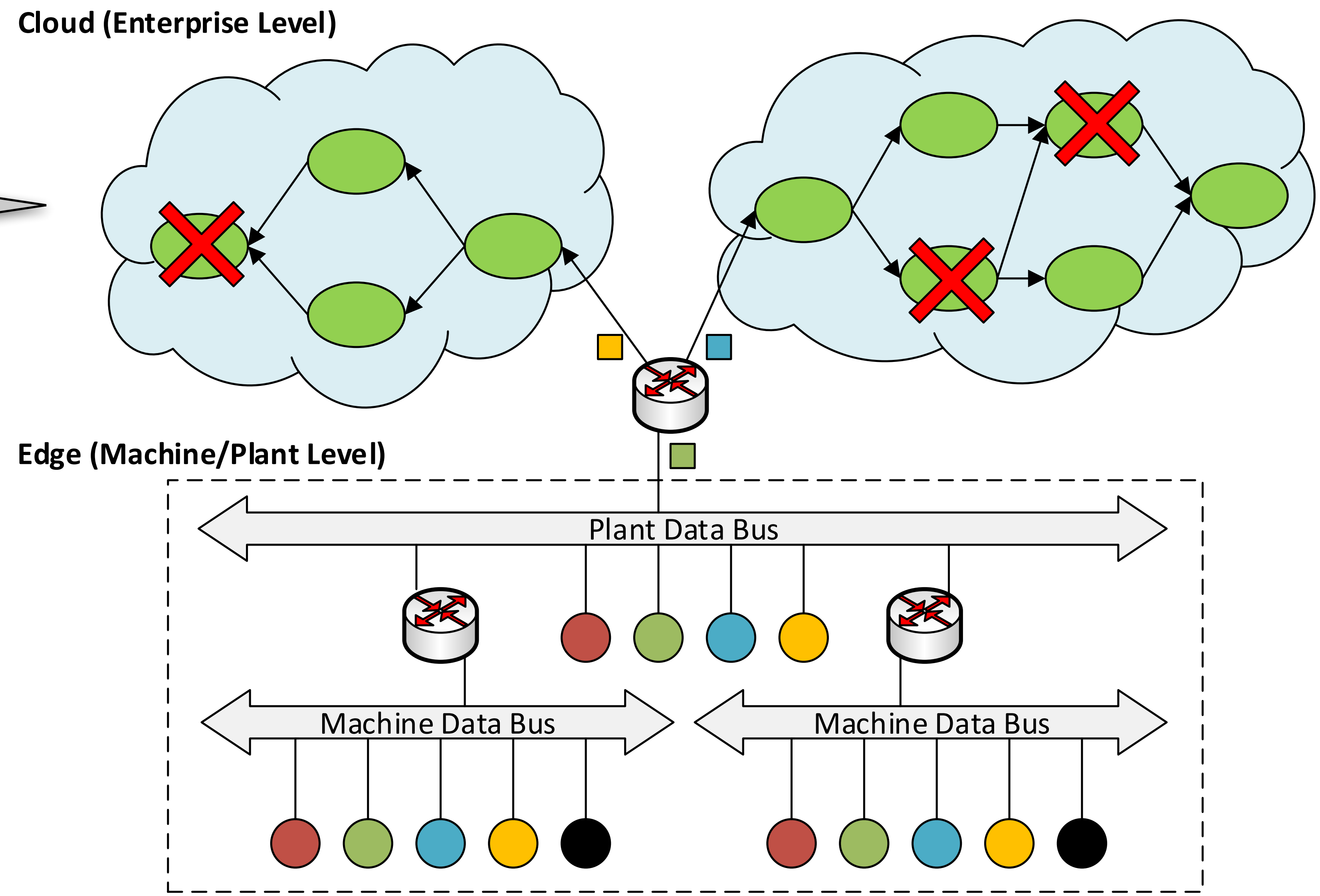
Data processing to optimize operations at edge

Data collection and sharing from edge devices to edge devices or cloud



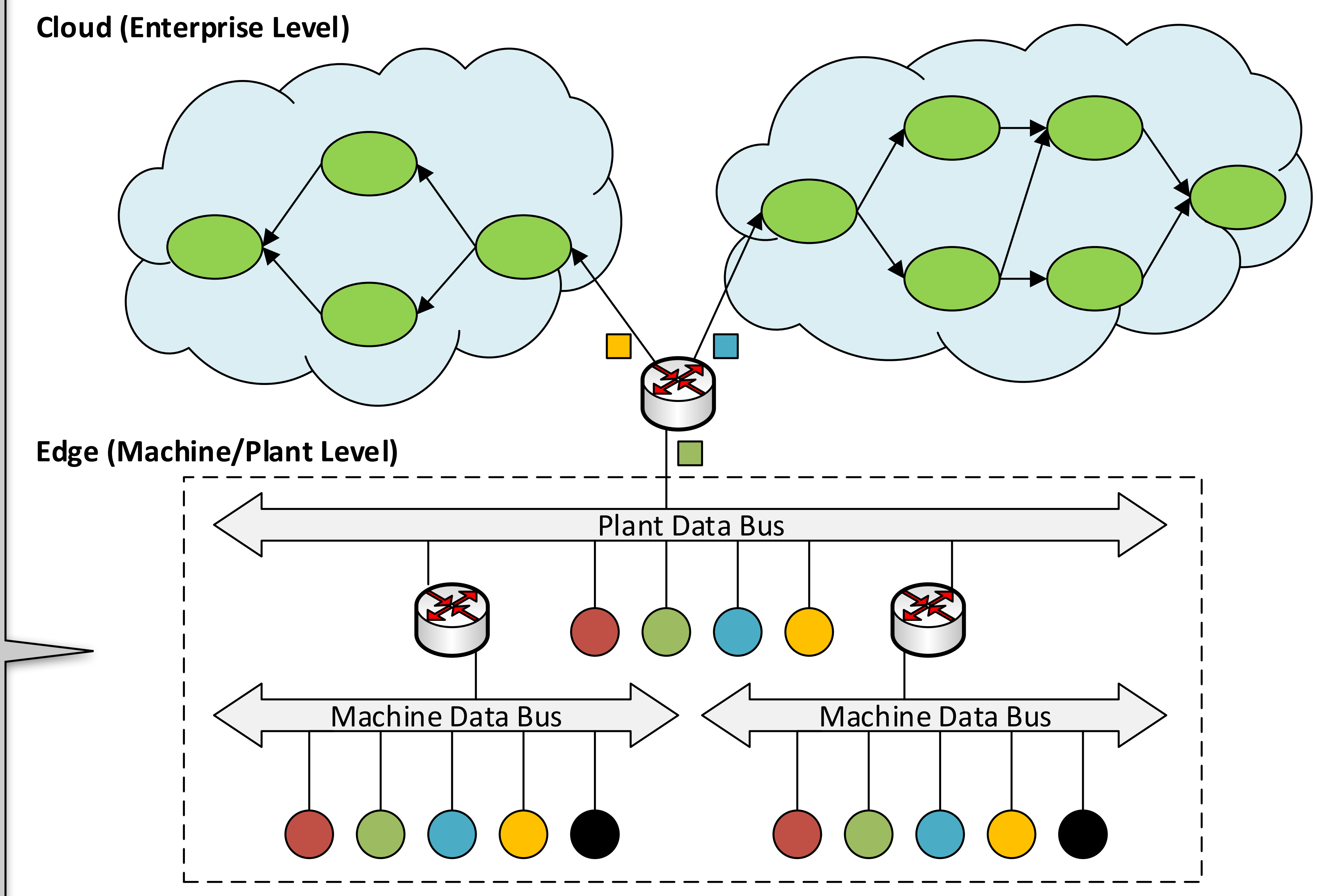
Challenge 1:
High Availability and Timeliness at Enterprise-level

- Faults can happen by failures physical machines (PMs) or virtual machines (VMs) in the cloud
- Failover using periodic snapshots of VMs
- How to guarantee the same service level even after failover? Optimal placement of backup VMs?

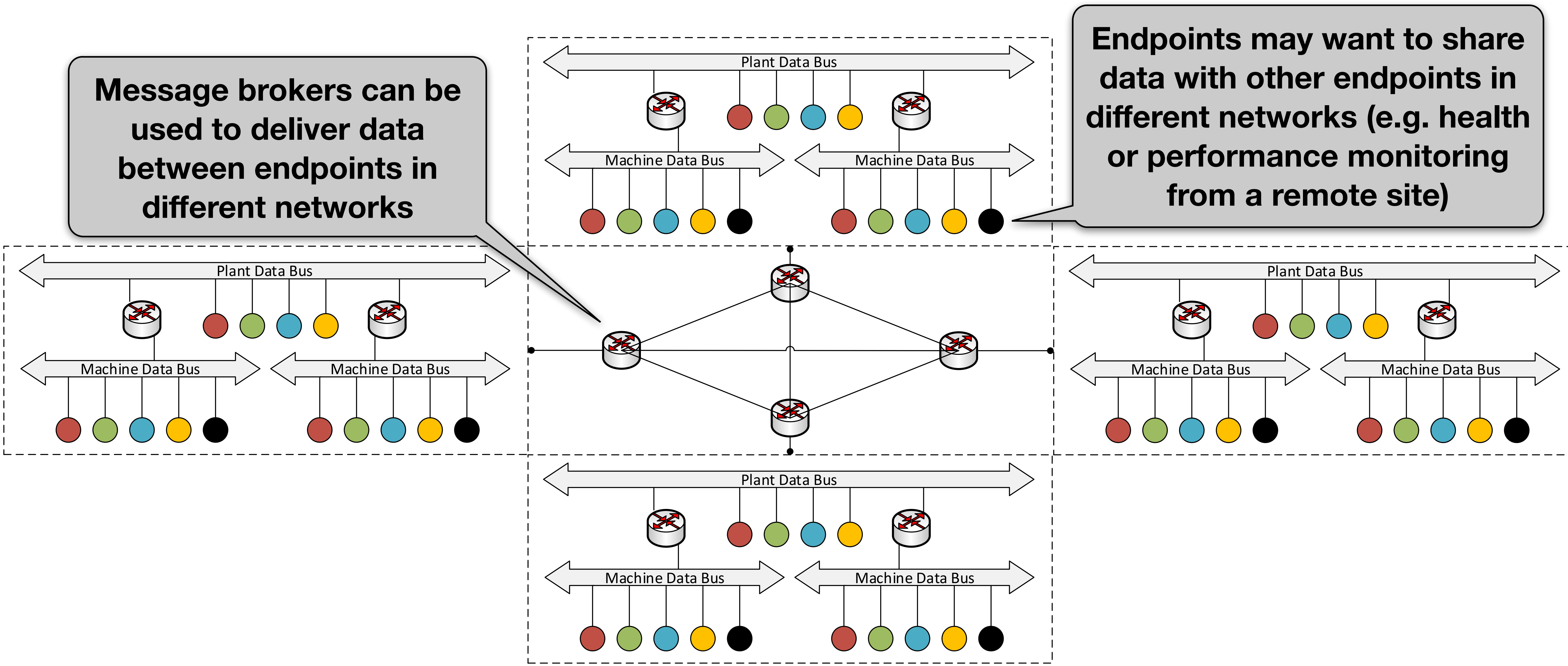


Challenge 2:
Scalability of Discovering Devices and Endpoints at Edge

- Data distribution from a turbine to other turbines or operation centers
- A turbine contains more than 50 sensors
- A modern wind farm contains more than 200 turbines
- At least 10,000 endpoints exist

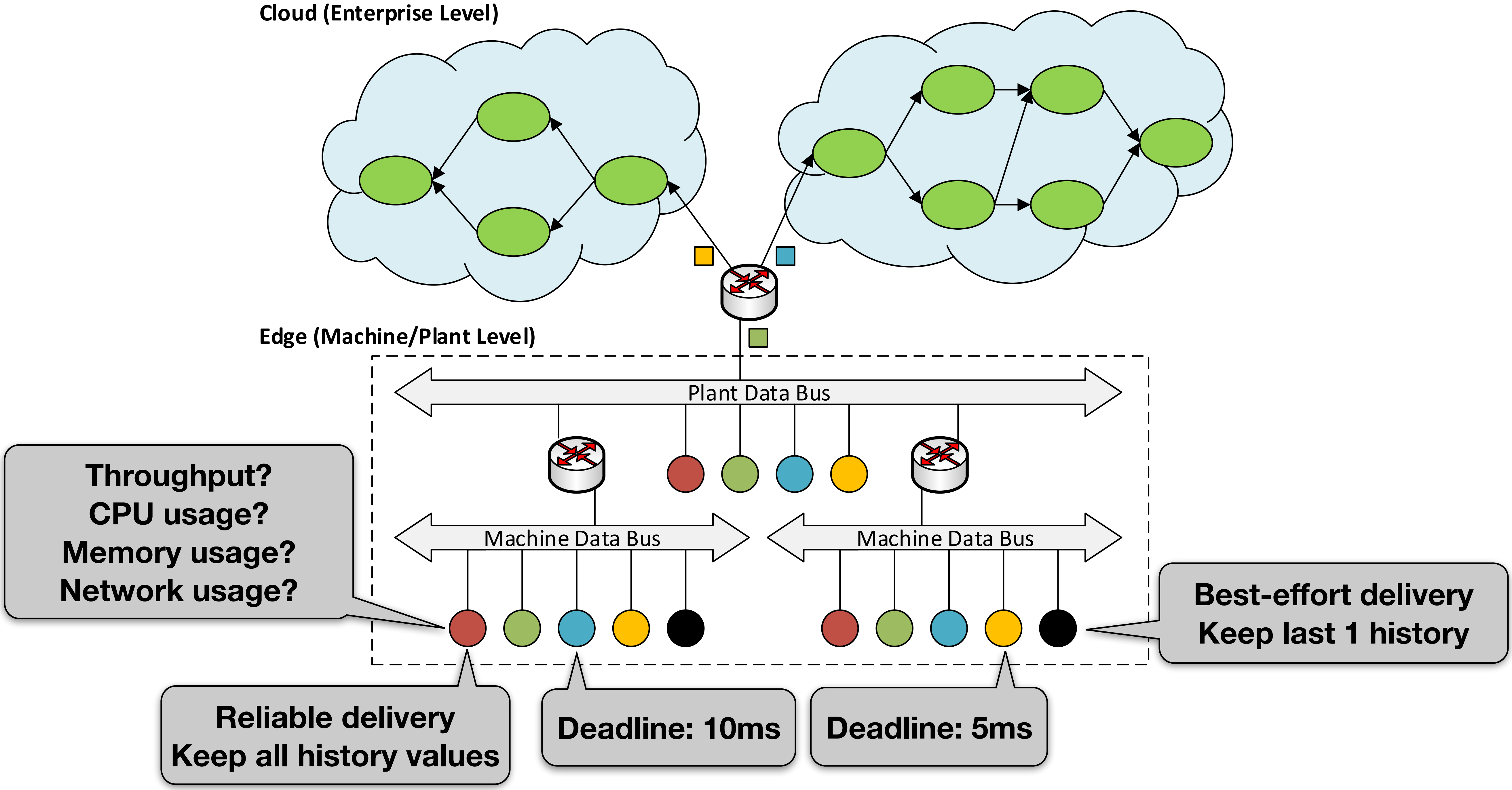


Challenge 3:
Overlay Networks for Data Distribution over WANs



- How to automatically discover brokers over WANs? How to form an optimal overlay network in terms of scalability and low latency? How to guarantee consistency of dissemination paths for dynamic endpoints?

Challenge 4:
Testing Expected Performance by Different QoS settings?



- Requires a technique to validate performance impact by different QoS configurations

Focus Areas in Industrial Internet Systems

	Turbine (Machine)	Wind farm (Plant)	Power producer (Enterprise)
Focus Area 1			Fault-tolerant and real-time cloud infrastructures
Focus Area 2		Scalable discovery protocols for pub/sub middleware	
Focus Area 3		Coordination service for pub/sub middleware in WANs	Coordination service for pub/sub middleware in WANs
Focus Area 4		Performance testing for different combinations of QoS configurations	Performance testing for different combinations of QoS configurations

Focus Areas in Industrial Internet Systems

	Turbine (Machine)	Wind farm (Plant)	Power producer (Enterprise)
Focus Area 1			Fault-tolerant and real-time cloud infrastructures
Focus Area 2		Scalable discovery protocols for pub/sub middleware	
Focus Area 3		Coordination service for pub/sub middleware in WANs	
Fo		Performance testing for different combinations of QoS configurations	

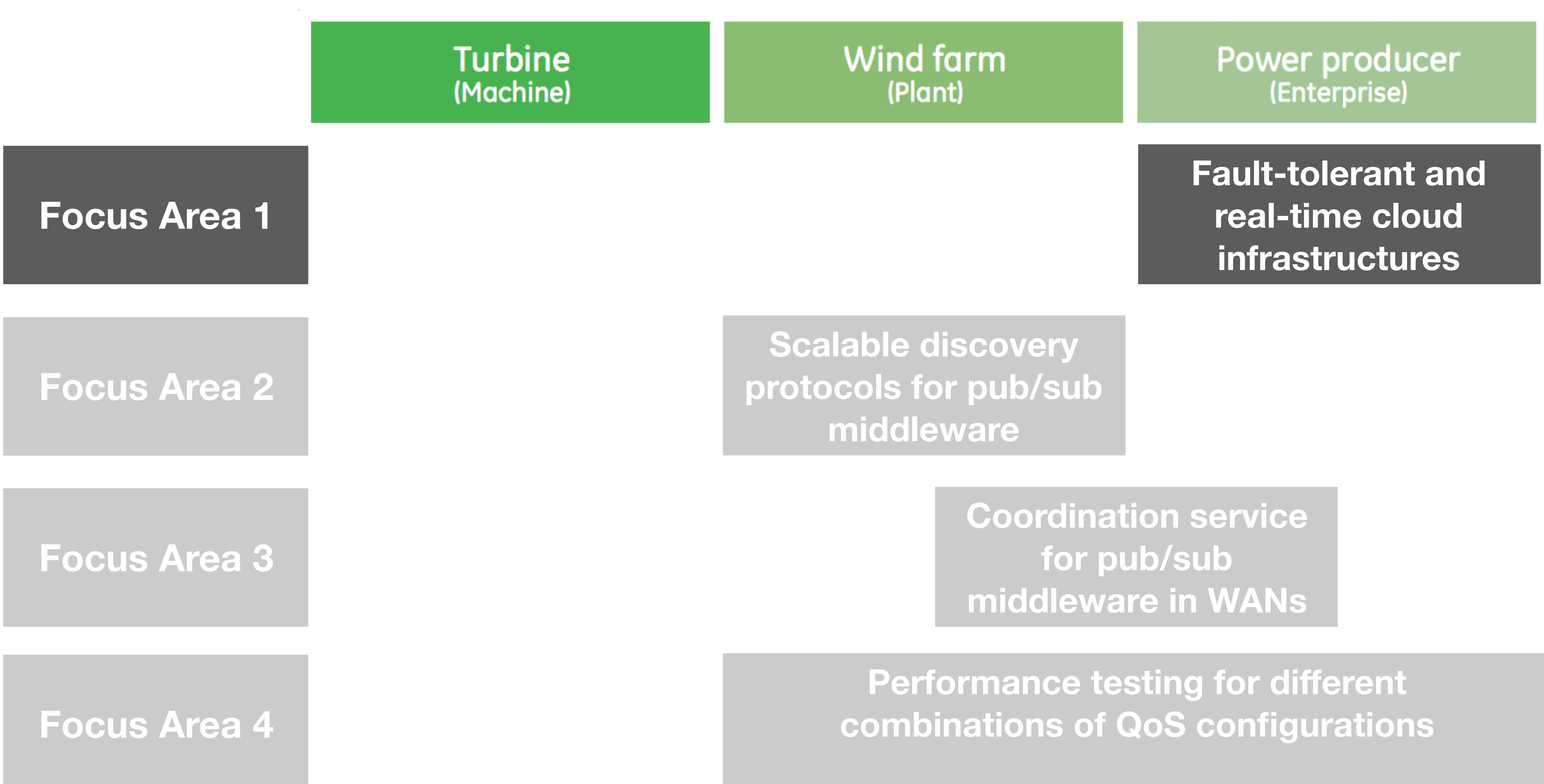
Inherent Complexity

Focus Areas in Industrial Internet Systems

	Turbine (Machine)	Wind farm (Plant)	Power producer (Enterprise)
Focus Area 1			Fault-tolerant and real-time cloud infrastructures
Focus Area 2		Scalable discovery protocols for pub/sub middleware	
F		Coordination service for pub/sub middleware in WANs	
Focus Area 4		Performance testing for different combinations of QoS configurations	

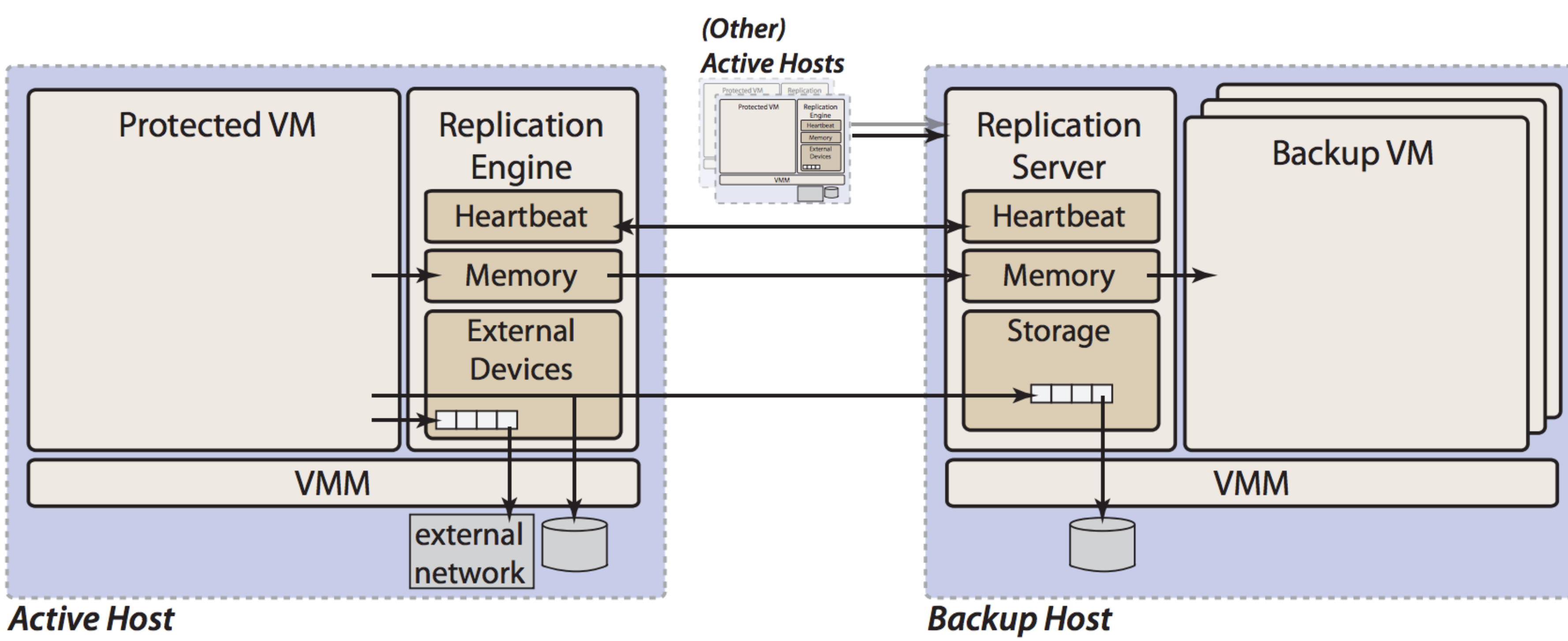
Accidental Complexity

Focus Area 1:
Fault-Tolerant and Real-Time Cloud Infrastructure



Context

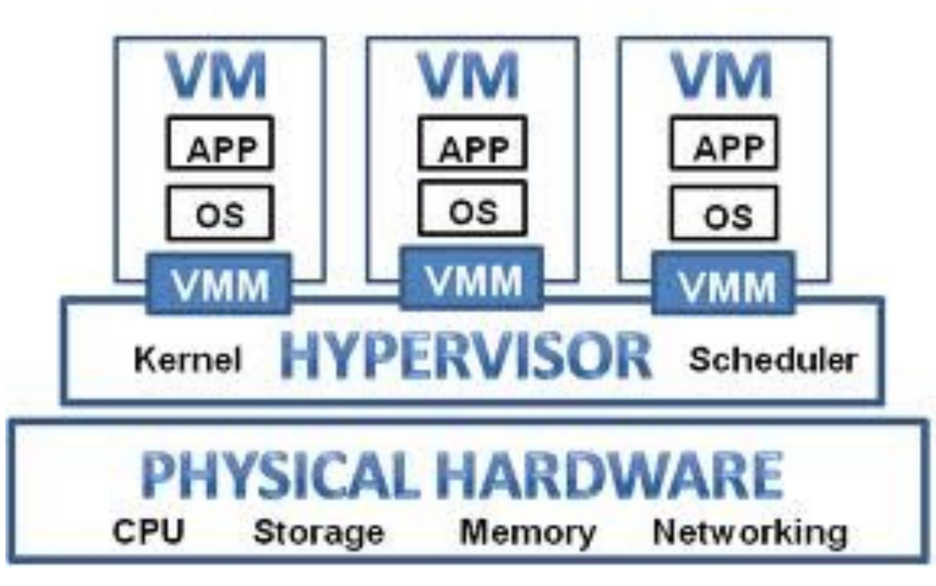
- High Availability (HA) is a key requirement for mission critical systems in the cloud
- HA solutions using VM (e.g. Remus and Kemari)
- Continuous replicate states of VMs to backup PMs

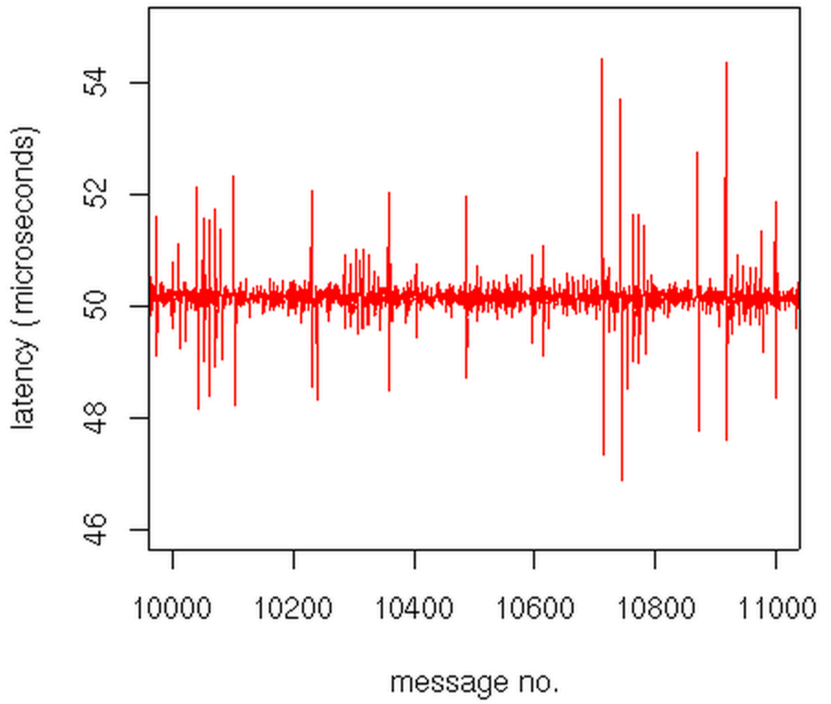


Reference from https://www.usenix.org/legacy/event/nsdi08/tech/full_papers/cully/cully.pdf

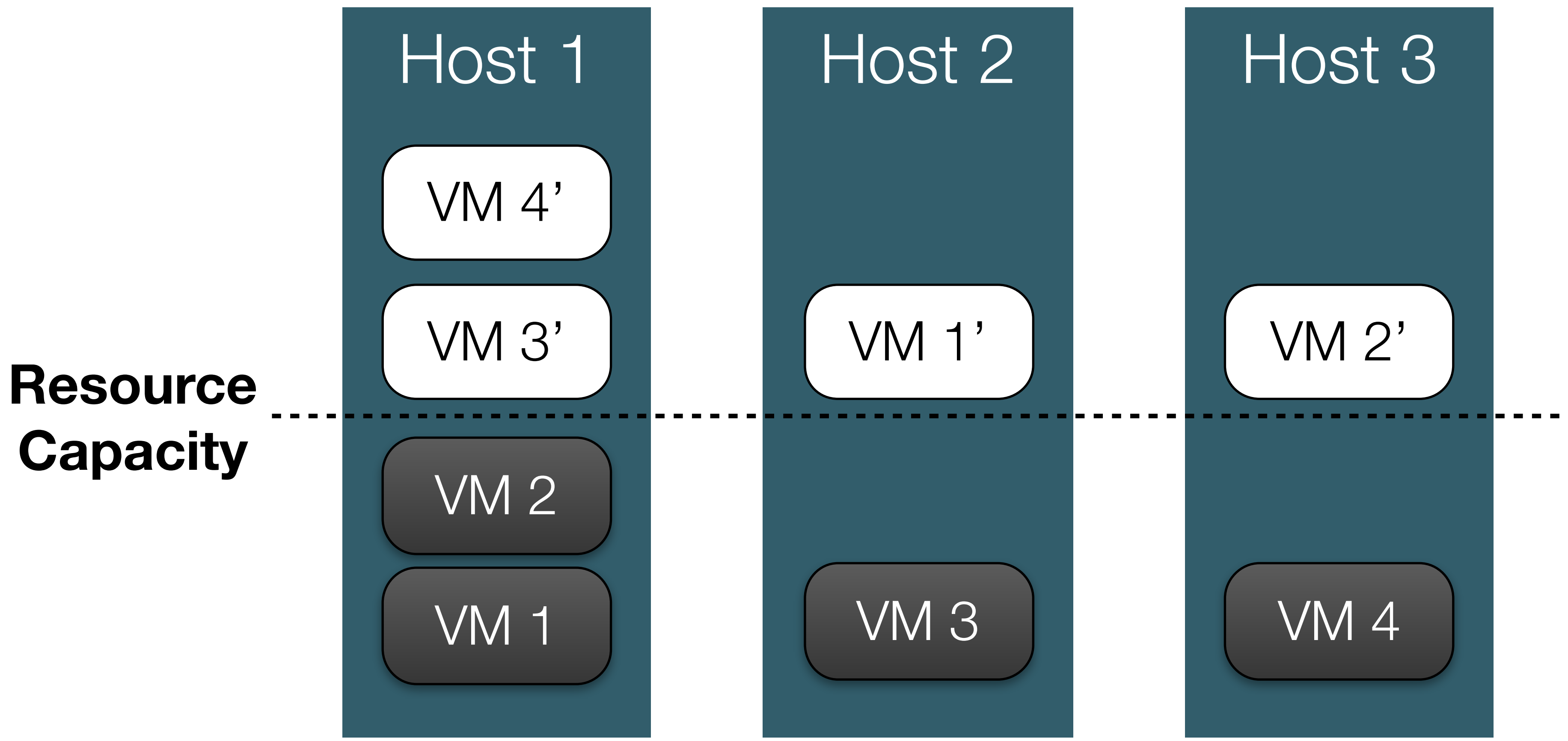
Challenges

- VM placement is critical to avoid resource contention because VM shares underlying PM's resources
- Lack of middleware for automated and effective placement of backup VMs
    - Makes systems available despite failures
    - Needs to guarantee low latency for real-time applications

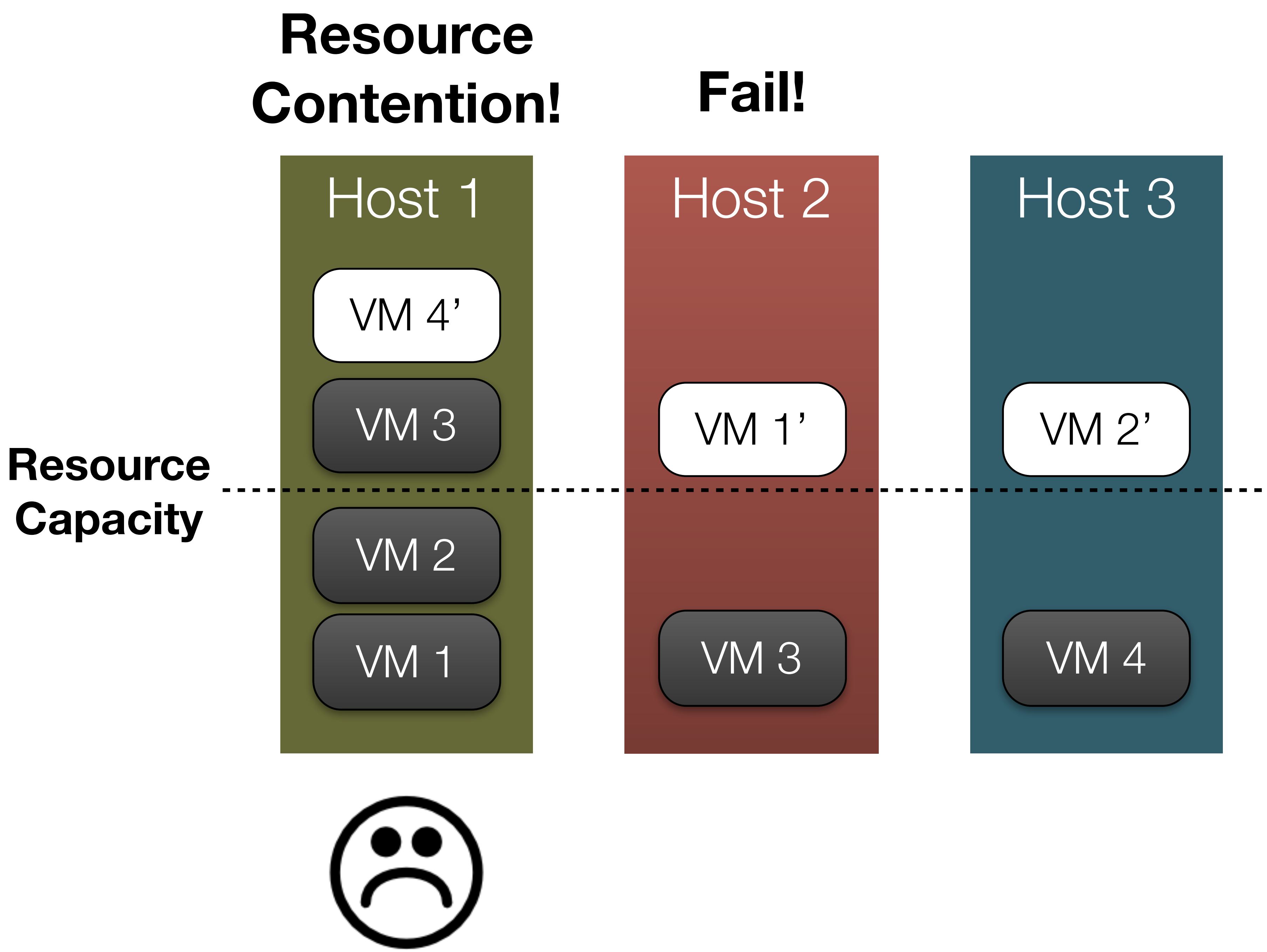




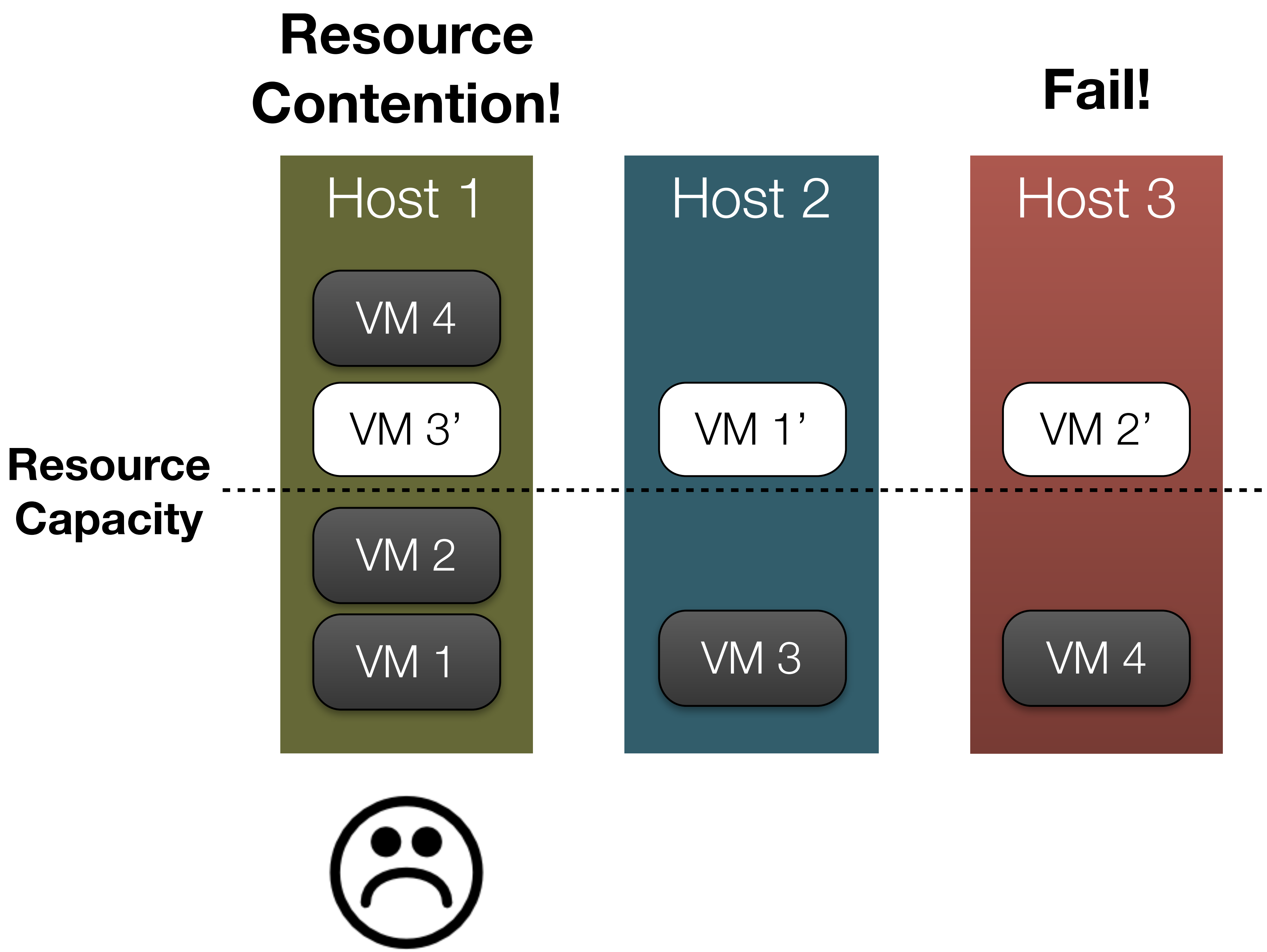
VM Placement
<u>Without</u> Considering Resource Constraints



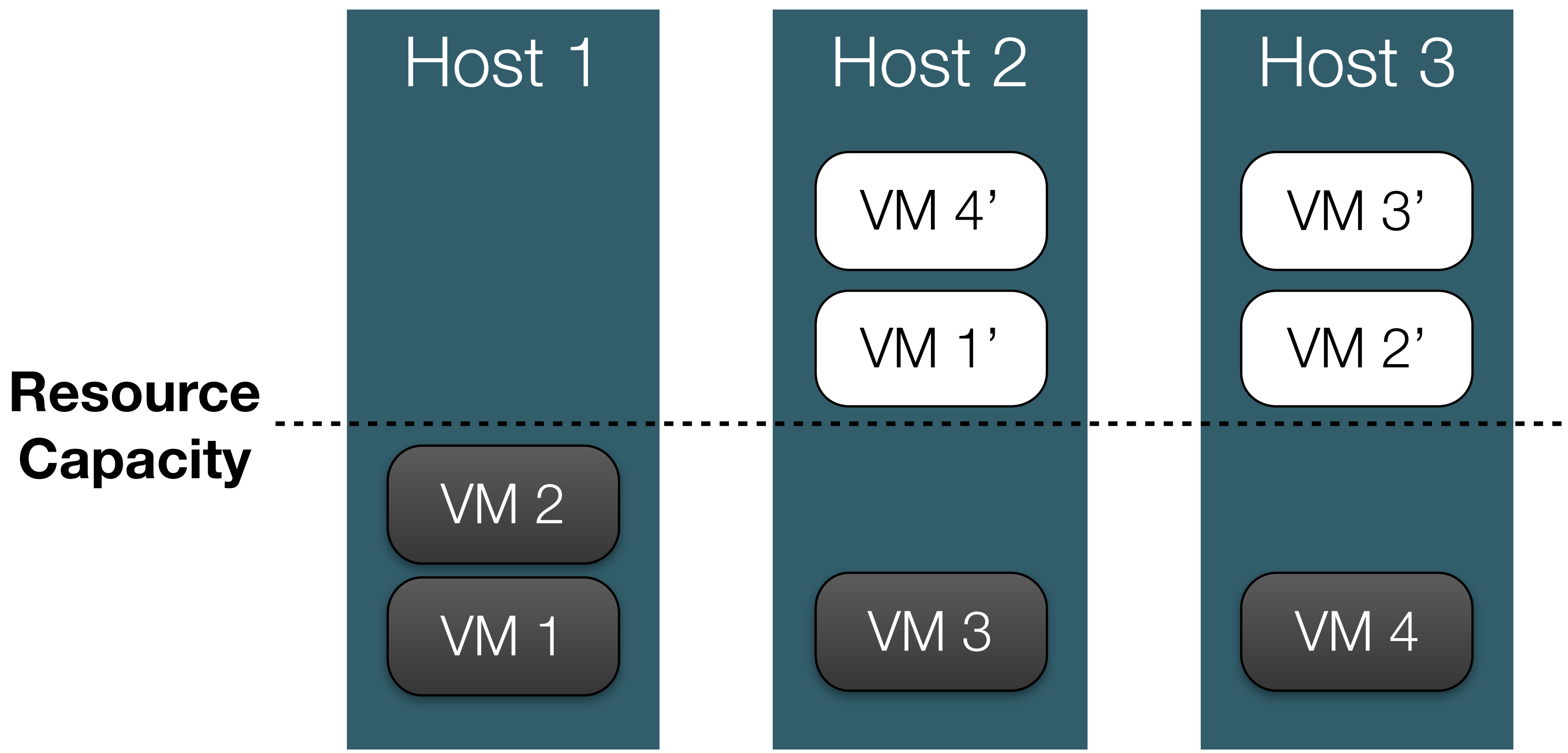
VM Placement
<u>Without</u> Considering Resource Constraints



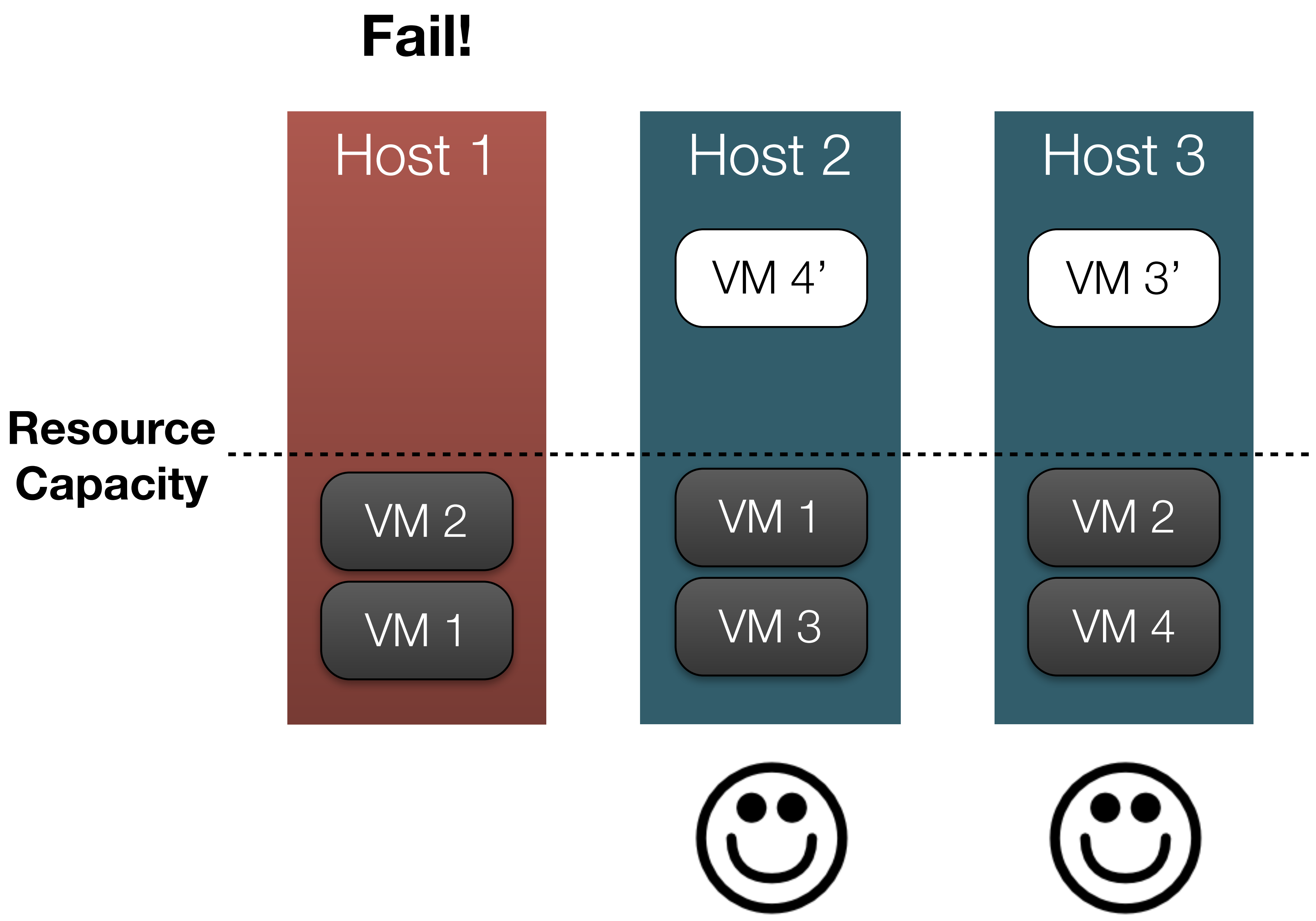
VM Placement
Without Considering Resource Constraints



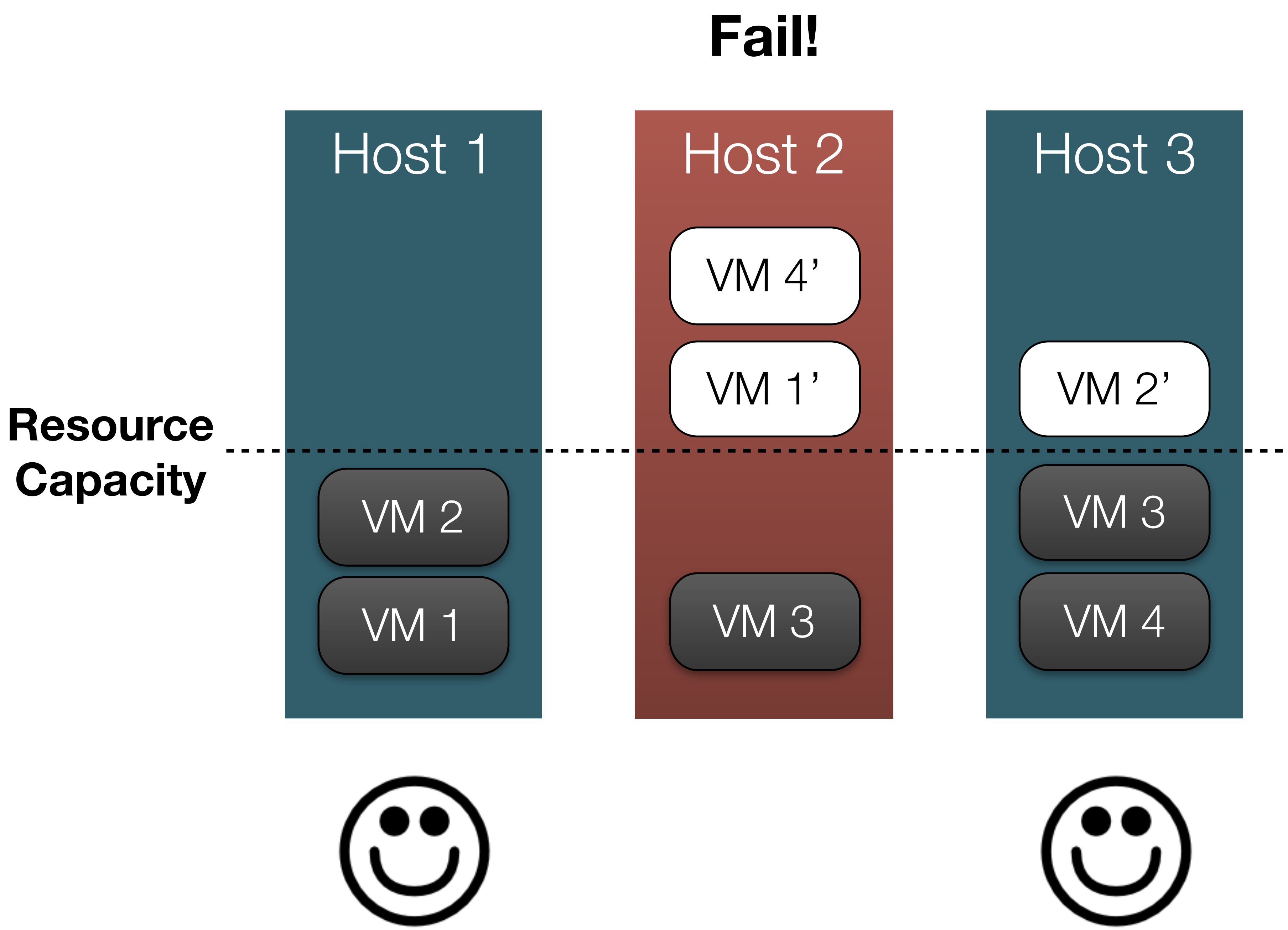
VM Placement
<u>With</u> Considering Resource Constraints



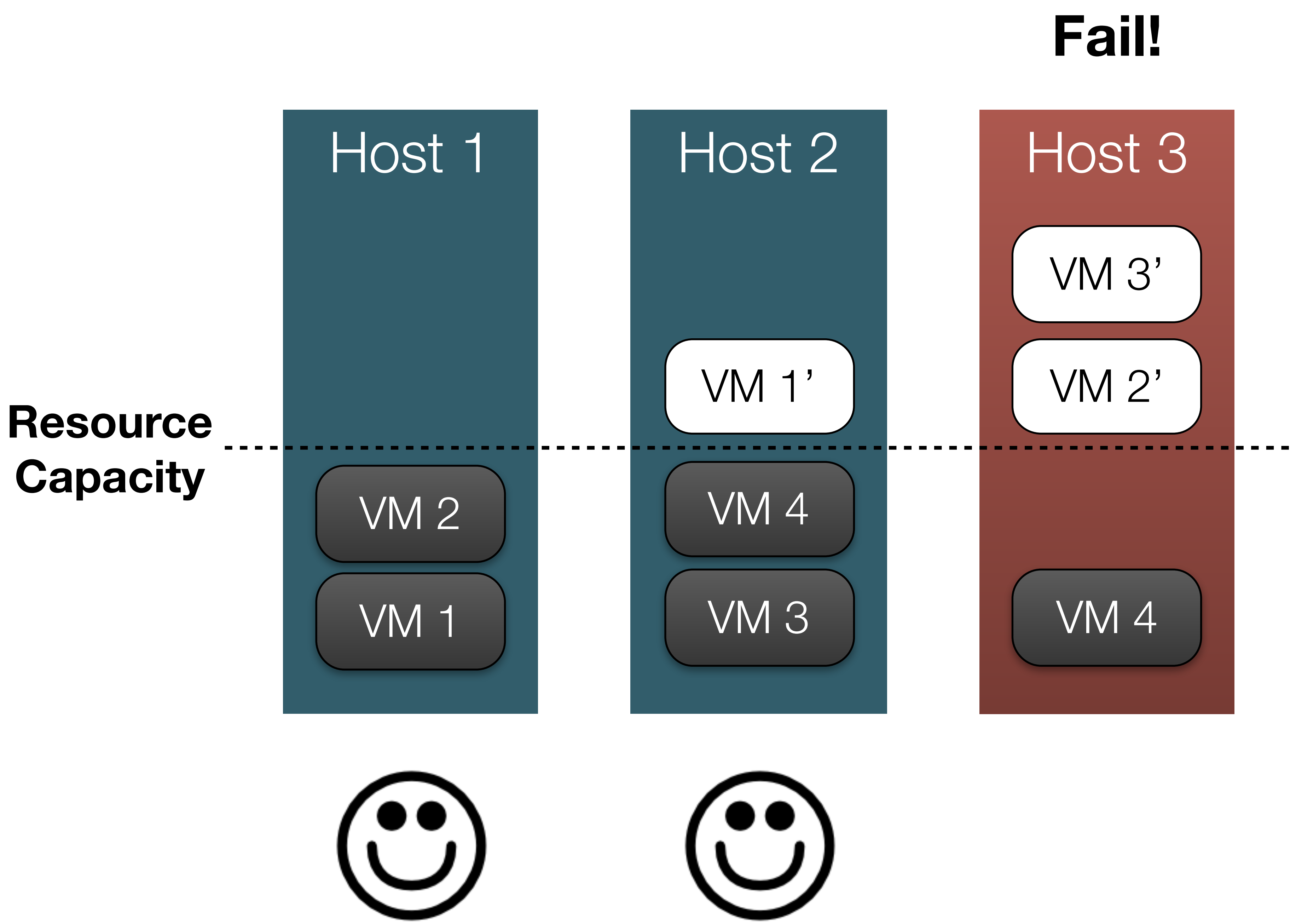
VM Placement
With Considering Resource Constraints



VM Placement
With Considering Resource Constraints



VM Placement
<u>With</u> Considering Resource Constraints



Problem Formulation

Notation	Definition
x_{ij}	Boolean value to determine the i^{th} VM to the j^{th} physical host mapping
x'_{ij}	Boolean value to determine the replication of the i^{th} VM to the j^{th} physical host mapping
y_j	Boolean value to determine usage of the physical host j
c_i	CPU usage of the i^{th} VM
c'_i	CPU usage of the i^{th} VM's replica
m_i	Memory usage of the i^{th} VM
m'_i	Memory usage of the i^{th} VM's replica
b_i	Network bandwidth usage of the i^{th} VM
b'_i	Network bandwidth usage of the i^{th} VM's replica
C_j	CPU capacity of the j^{th} physical host
M_j	Memory capacity of the j^{th} physical host
B_j	Network bandwidth of the j^{th} physical host

Problem Formulation

Minimize the number of used PMs

Make sure all VMs are deployed

Make sure all backups of VMs are deployed

CPU capacity used by VMs should be less than PMs

Memory capacity used by VMs should be less than PMs

Network capacity used by VMs should be less than PMs

Primary and backup VMs should be located in different PMs

$$\text{minimize} \quad \sum_{j=1}^{m} y_j \tag{II.1}$$

$$\text{subject to} \quad \sum_{j=1}^{m} x_{ij} = 1 \qquad \forall i \tag{II.2}$$

$$\sum_{j=1}^{m} x'_{ij} = 1 \qquad \forall i \tag{II.3}$$

$$\sum_{i=1}^{n} c_i x_{ij} + \sum_{i=1}^{n} c'_i x'_{ij} \le C_j y_j \qquad \forall j \tag{II.4}$$

$$\sum_{i=1}^{n} m_i x_{ij} + \sum_{i=1}^{n} m'_i x'_{ij} \le M_j y_j \qquad \forall j \tag{II.5}$$

$$\sum_{i=1}^{n} b_i x_{ij} + \sum_{i=1}^{n} b'_i x'_{ij} \le B_j y_j \qquad \forall j \tag{II.6}$$

$$\sum_{i=1}^{n} x_{ij} + \sum_{i=1}^{n} x'_{ij} = 1 \qquad \forall j \tag{II.7}$$

$$x_{ij} = \{0,1\},\ x'_{ij} = \{0,1\},\ y_j = \{0,1\} \tag{II.8}$$

Related Work

Related Research
B. Cully, G. Lefebvre, D. Meyer, M. Feeley, N. Hutchinson, A. Warfield, Remus: High Availability via Asynchronous Virtual Machine Replica- tion, in: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, USENIX Association, 2008, pp. 161–174.
Y. Tamura, K. Sato, S. Kihara, S. Moriai, Kemari: Virtual machine synchronization for fault tolerance, In USENIX 2008 Poster Session.
K.-Y. Hou, M. Uysal, A. Merchant, K. G. Shin, S. Singhal, Hydravm: Low-cost, transparent high availability for virtual machines, Tech. rep., HP Laboratories (2011)
C. Hyser, B. McKee, R. Gardner, B. Watson, Autonomic center, Hewlett Packard Laboratories, Tech. Rep. HPL-2
S. Lee, R. Panigrahy, V. Prabhakaran, V. Ramasubrahma Validating heuristics for virtual machines consolidation, Microsoft Research, MSR-TR-2011-9.

Good for highly available systems that do not require automatic VM placement

Related Work

Related Research
B. Cully, G. Lefebvre, D. Meyer, M. Feeley, N. Hutchinson, A. Warfield, Remus: High Availability via Asynchronous Virtual Machine Replica- tion, in: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, USENIX Association, 2008, pp. 161–174.
Y. Tamura, K. Sato, S. Kihara, S. Moriai, Kemari: Vi … nce, In USENIX 2008 Poster Session.
K.-Y. Hou, M. Uysal, A. Merchant, K. G. Shin, S. S … availability for virtual machines, Tech. rep., HP Laboratories (2011)
C. Hyser, B. McKee, R. Gardner, B. Watson, Autonomic virtual machine placement in the data center, Hewlett Packard Laboratories, Tech. Rep. HPL-2007-189.
S. Lee, R. Panigrahy, V. Prabhakaran, V. Ramasubrahmanian, K. Talwar, L. Uyeda, U. Wieder, Validating heuristics for virtual machines consolidation, Microsoft Research, MSR-TR-2011-9.

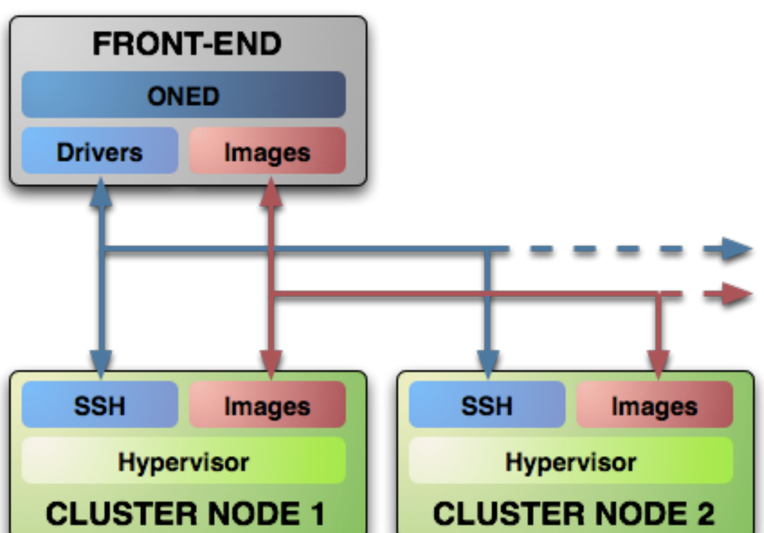
Good for systems guaranteeing expected latency via automatic VM placement, but not support high availability

Solution Approach

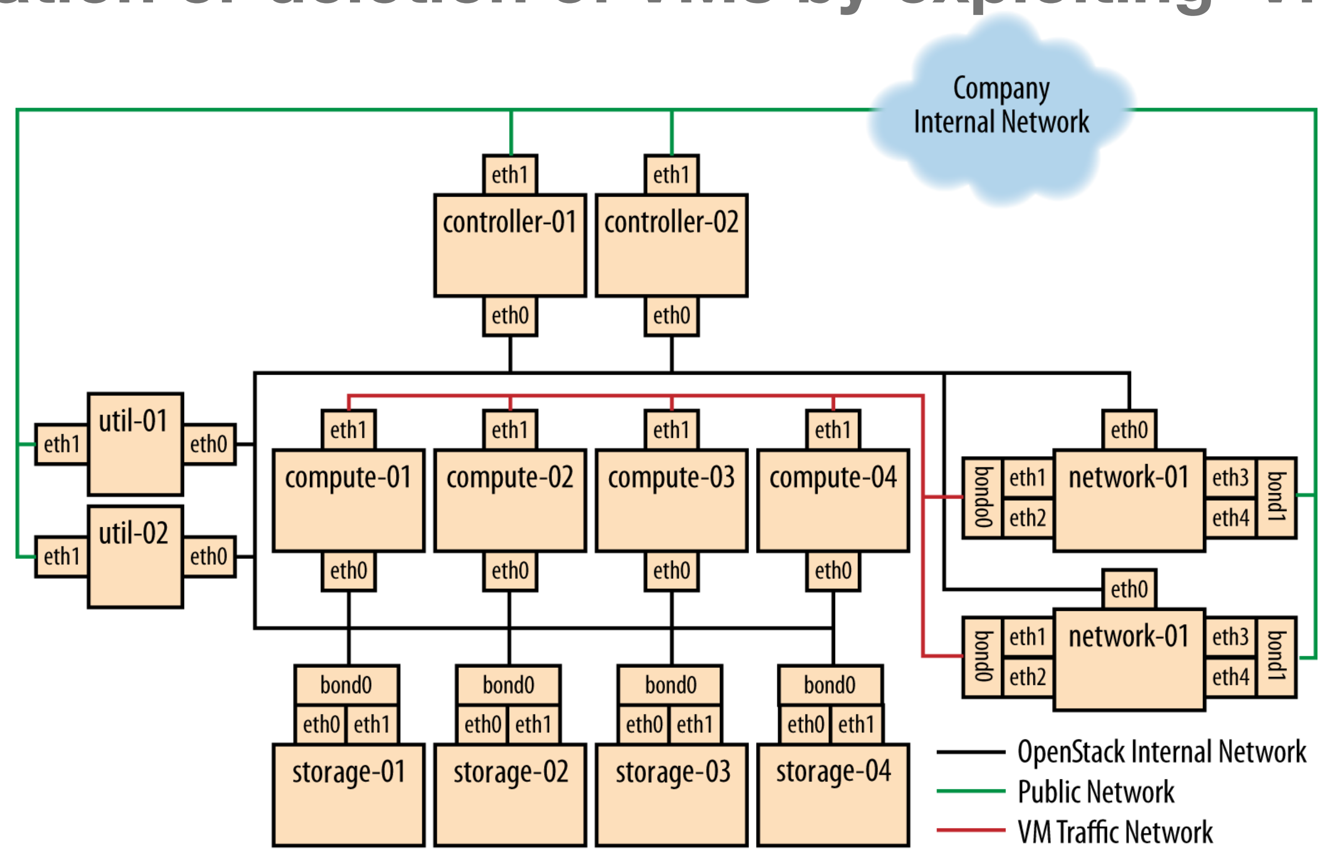
- Fault-tolerant middleware in the cloud to provide high availability for real-time applications
  - Automated placement of VM backups with avoiding resource contention to guarantee low latency
  - The design of a pluggable framework that enables application developers to provide their strategies for choosing physical hosts for VM replicas

Architectures of Cloud Middleware

- Two-level architectures
  - Front-end (controller) node: Manages and schedules compute and network resources of a cluster upon requests from clients
  - Cluster (compute) node: Creation or deletion of VMs by exploiting VM hypervisors



OpenNebula



OpenStack

Reference from http://archives.opennebula.org/documentation:rel3.2.bck:plan
Reference from http://docs.openstack.org/openstack-ops/content/example_architecture.html

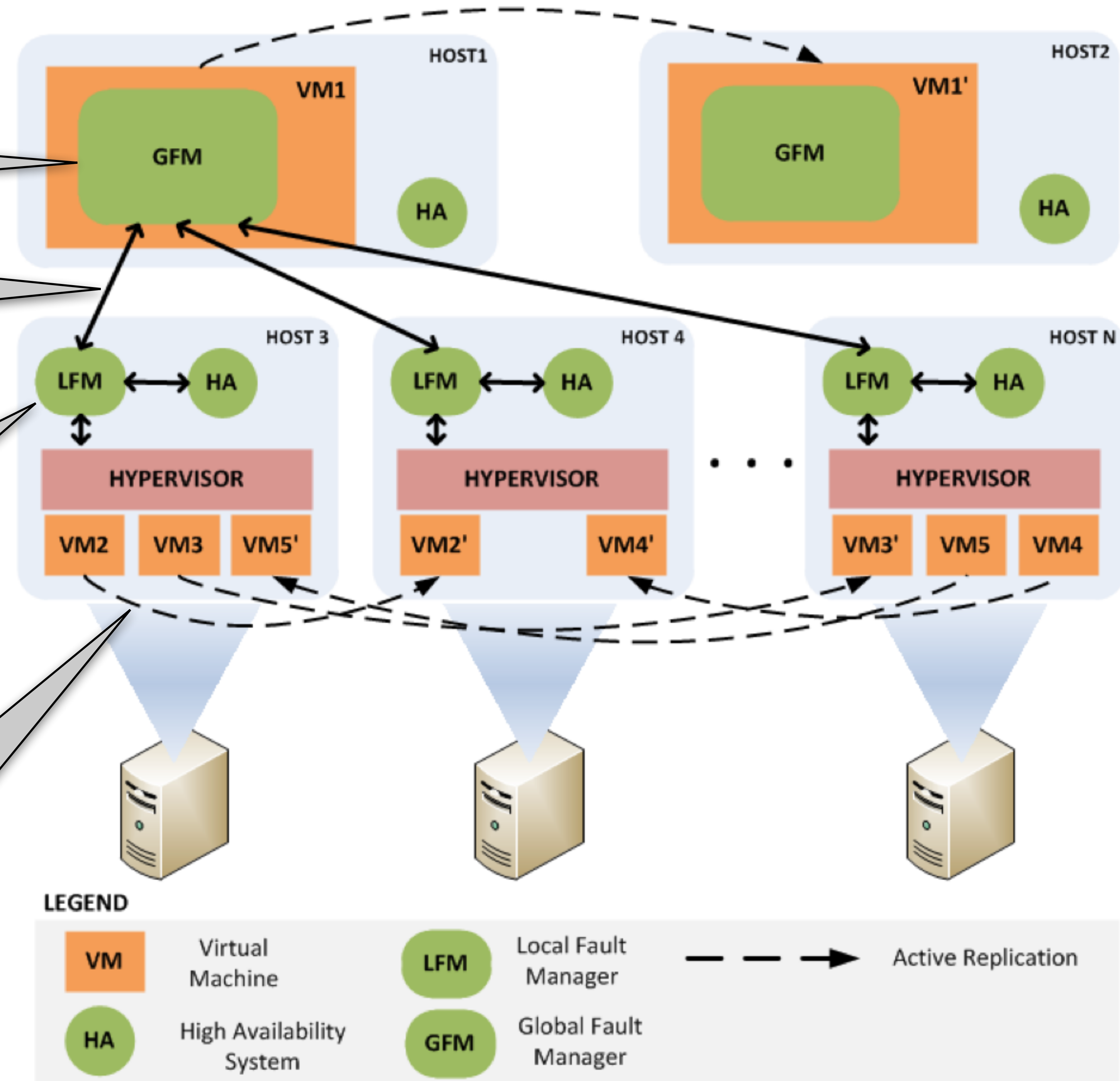
System Architecture

Controller node that deals with global view of resource management and failures

communications for monitoring resources and communications for VM backup placement

Compute node that deals with local view of resource management and failures

Continuous state synchronization over networks

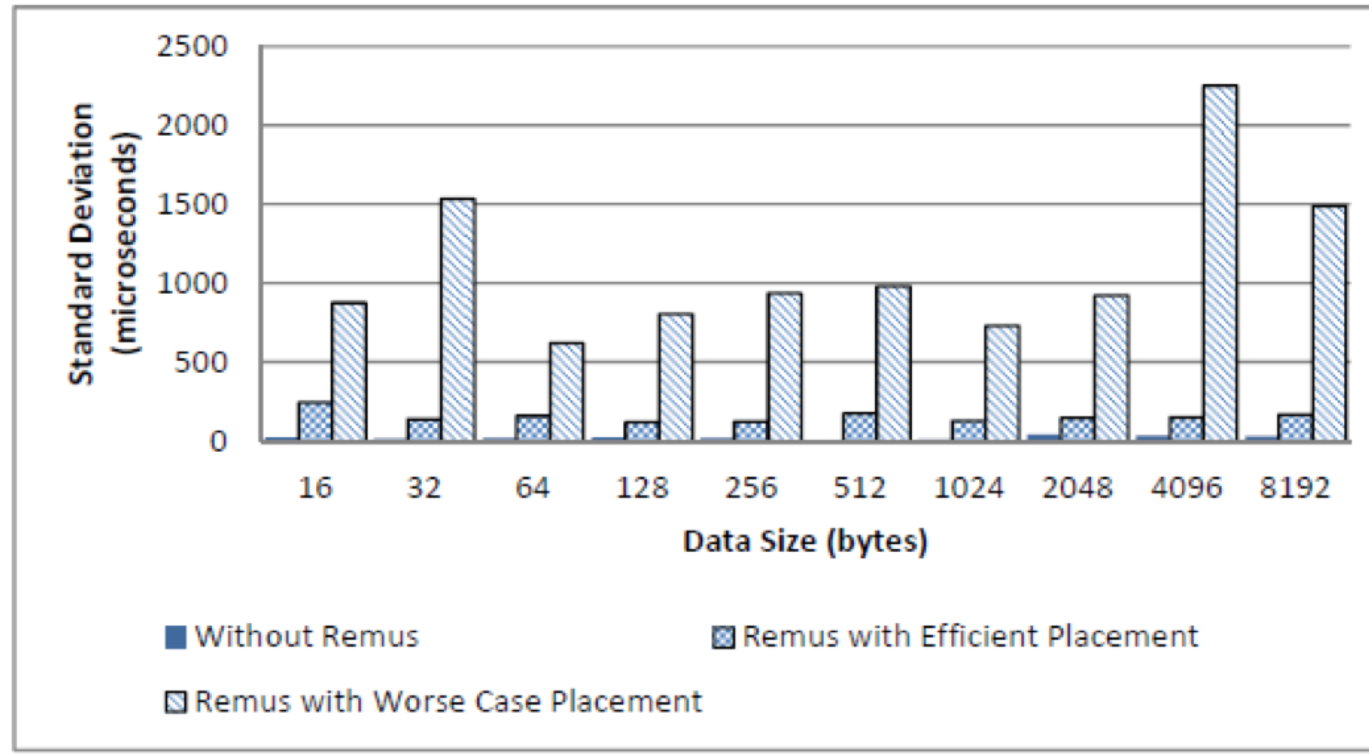


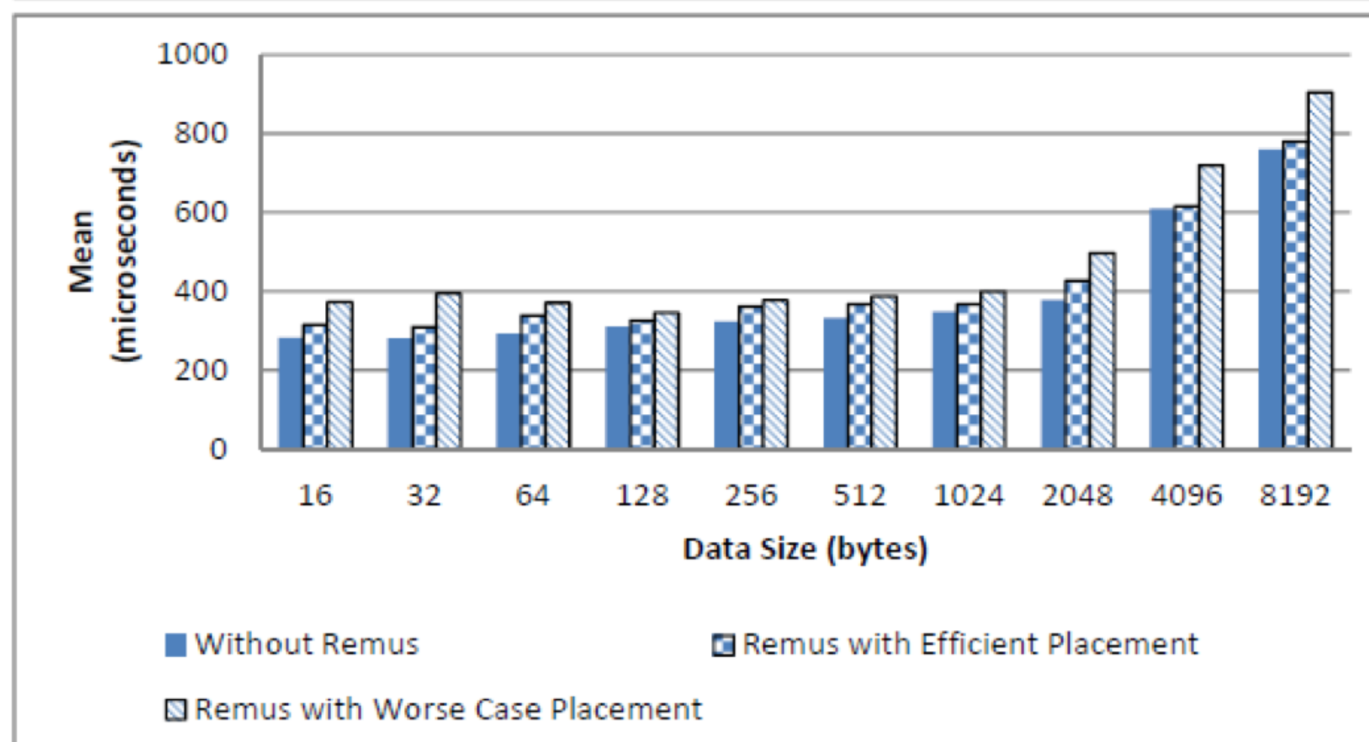
Testbed Environment

- A cluster of 20 machines connected to 1Gb network
  - Each machine has 12 cores and 32G RAM
- OpenNebula 3.0 for a cloud platform
  - Network File System (NFS) for VM disk images
  - Xen for VM hypervisor
- RTI Connext DDS 5.0 for testing applications

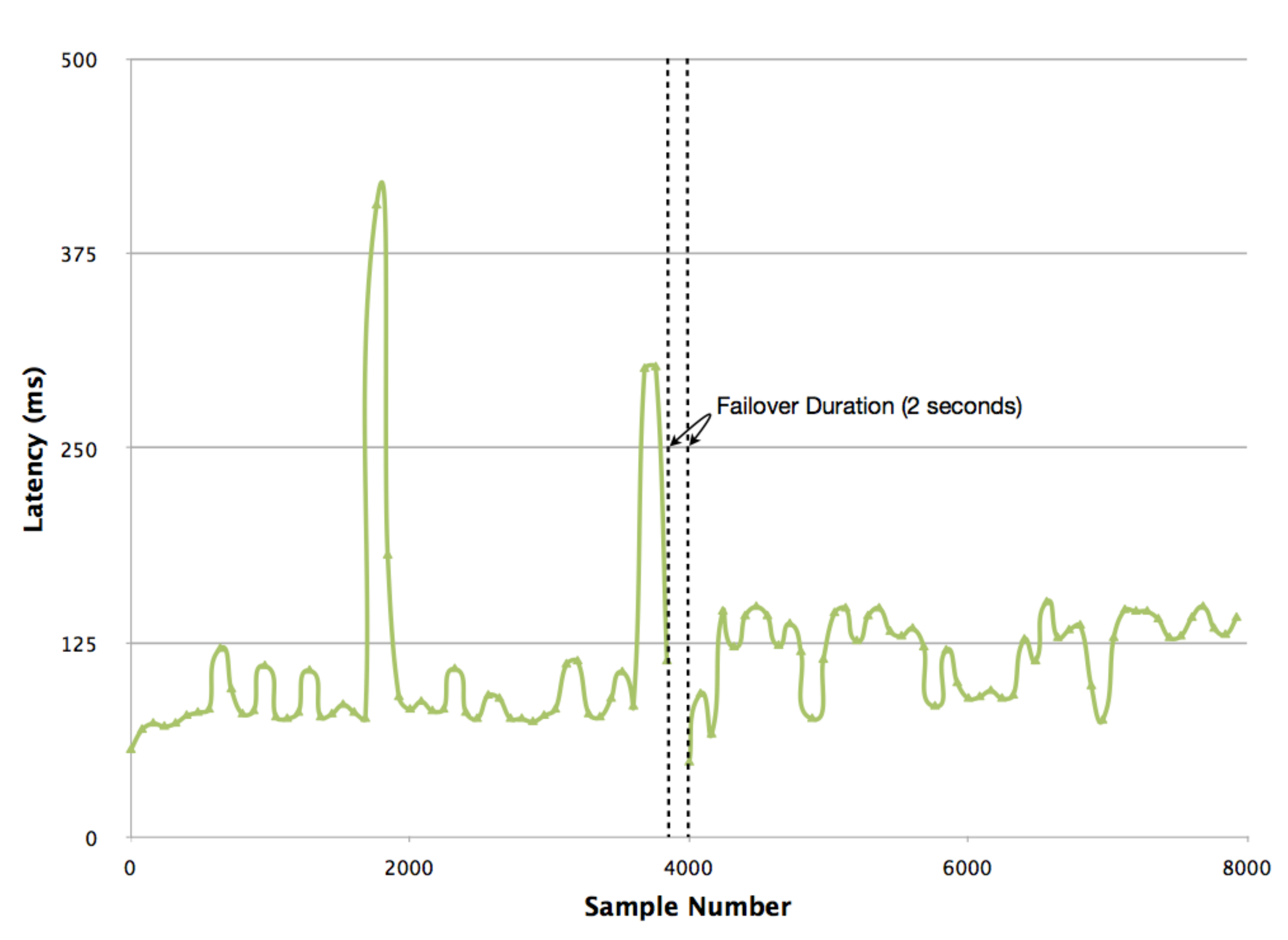
Latency Performance Test

- Used DDS performance benchmark
- Evaluate standard deviation (jitter) and mean of latency
- Some overhead caused by HA solutions
- High jitter cased by random placement of VM backups





Failover Impact on Latency of Remus



- **There is a failover duration (samples are missing) about 2 seconds**
- **Latency is slightly increased after failover phase**

Middleware-based Failover vs. VM-based Failover: Failover Impact on Sample Missed Ratio

VM-based failover

Middleware-based failover

Both VM and Middleware

	Missed Samples (total of 8000)	Missed Samples Percentage (%)
Experiment 3	221	2.76
Experiment 4	33	0.41
Experiment 5	14	0.18
Experiment 6	549	6.86

- Middleware-based failover makes less overhead than VM-based failover
  - Less number of missed samples when failover
- VM-based failover does not require specific failover implementations on middleware or applications

Lessons Learned

- Remus with efficient placement supports low fluctuation on latency after failover using cloud resources in an optimal way.

- VM level fault-tolerance incurs more overheads than middleware level fault-tolerance, but it must be used for some cases when middleware does not support high availability.

- Timeliness can be guaranteed with both resource scheduling by hypervisors and resource allocation.

- Kyoungho An, Shashank Shekhar, Faruk Caglar, Aniruddha Gokhale, and Shivakumar Sastry, "*A Cloud Middleware for Assuring Performance and High Availability of Soft Real-time Applications*", The Elsevier Journal of Systems Architecture (JSA): Embedded Systems Design, 2014.

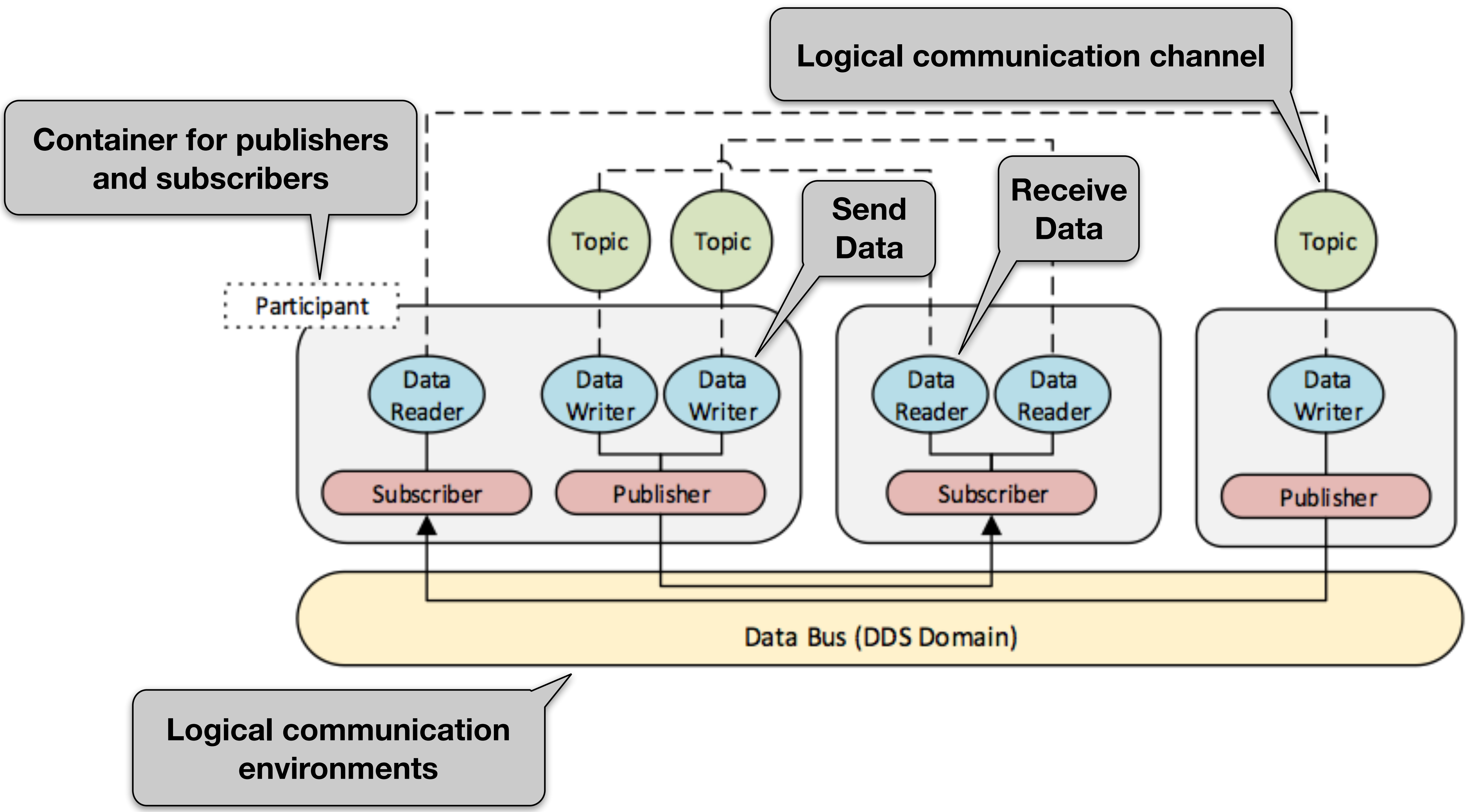
Focus Area 2:
Scalable discovery protocols for pub/sub middleware

	Turbine (Machine)	Wind farm (Plant)	Power producer (Enterprise)
Focus Area 1			Fault-tolerant and real-time cloud infrastructures
Focus Area 2		Scalable discovery protocols for pub/sub middleware	
Focus Area 3		Coordination service for pub/sub middleware in WANs	Coordination service for pub/sub middleware in WANs
Focus Area 4		Performance testing for different combinations of QoS configurations	Performance testing for different combinations of QoS configurations

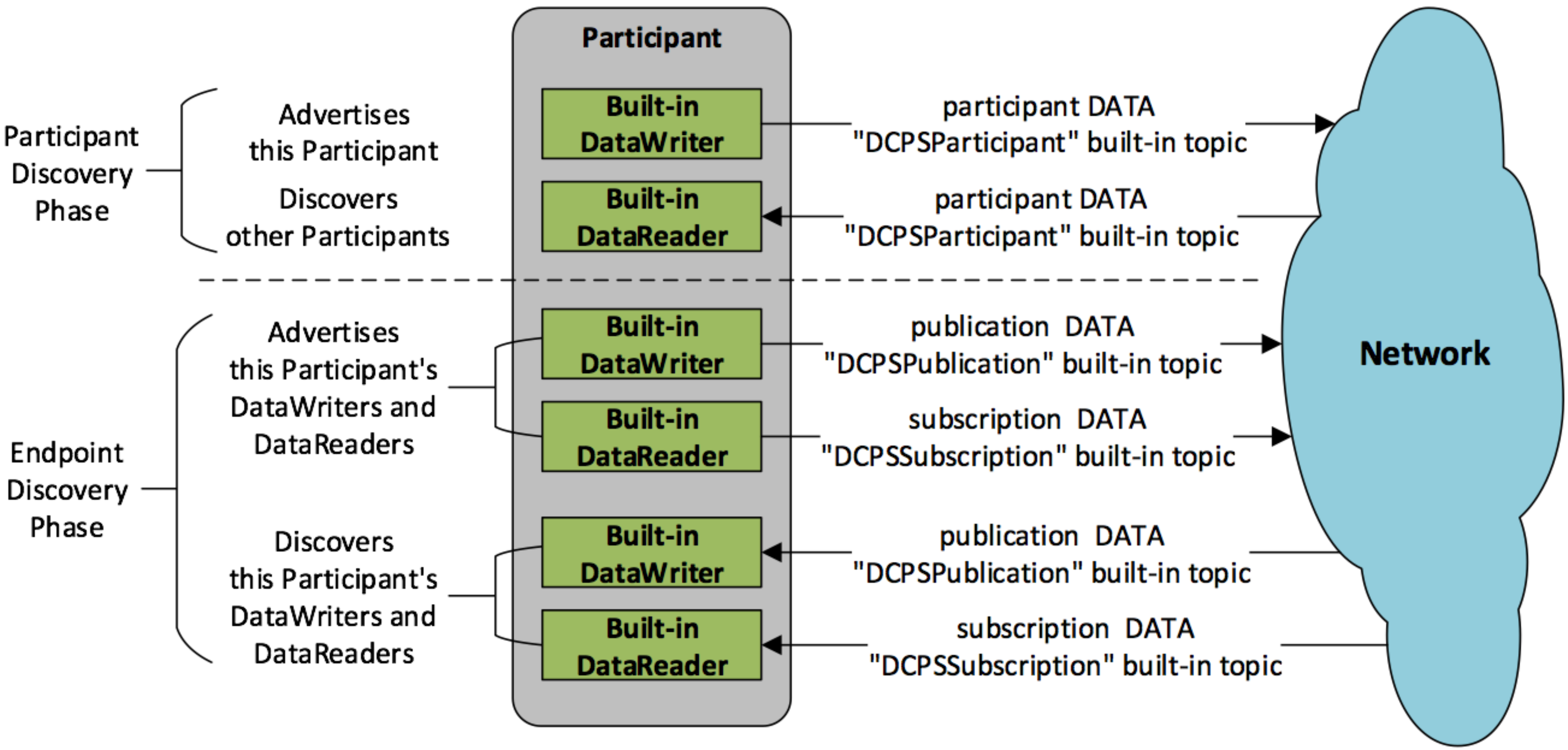
Context

- Data-centric pub/sub middleware can be used as a platform to share data in Industrial Internet systems
- Data Distribution Service (DDS) is an OMG standard specification for data-centric publish/subscribe middleware
    - Data-centric addressing
    - Decoupling between publishers and subscribers
    - Many-to-many communications
    - QoS and smart filtering supports
- These features can be realized at the discovery phase

DDS Architecture

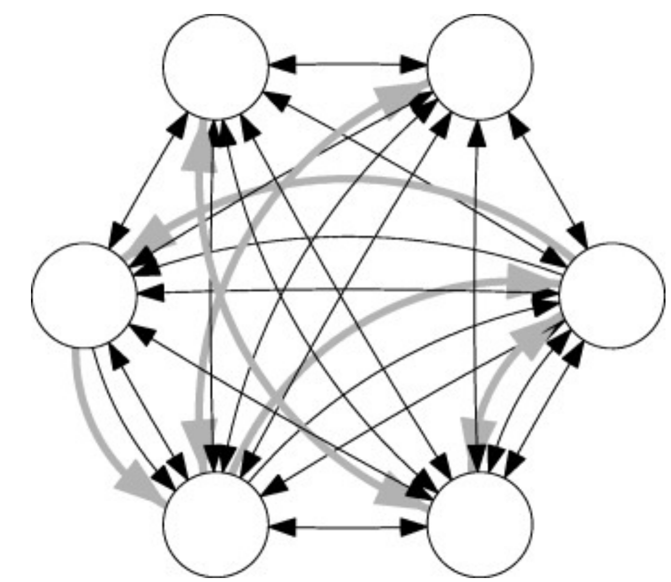


DDS Discovery Protocol Entities



Challenges

- Simple Discovery Protocol (SDP) is a default DDS discovery protocol
- SDP scales poorly as the number of peers and their endpoints increases in a domain
- Why?
  - Each participant sends/receives discovery messages to/from all participants in the same domain regardless of topics or endpoint types
- For a large scale system, substantial network, memory, and computing resources are consumed just for the discovery process
  - This overhead degrades discovery completion time and hence overall scalability



Related Work

Related Research
OCI's Centralized Repository for Discovery. Open DDS Developer's Guide. http://download.ociweb.com/OpenDDS/OpenDDS-latest.pdf, 2013.
RTI's Enterprise Discovery Protocol. RTI Connext DDS User's Manual. http://community.rti.com/rti-doc/510/ndds.5.1. 0/doc/pdf/RTI_CoreLibrariesAndUtilities_ UsersManual.pdf, 2013.
RTI. Limited-Bandwidth Plug-ins for DDS. http: //www DDS_Over_Low_Bandwidth.pdf, 2011.
J. Sanchez-Monedero, J. Povedano-Molina, J. M. Lop Filter-based Discovery Protocol for DDS Middleware.
J. Hoffert, S. Jiang, and D. C. Schmidt. A Taxonomy of Discovery Services and Gap Analysis for Ultra-large Scale Systems. In Proceedings of the 45th annual southeast regional conference, pages 355–361. ACM, 2007.

Good for a system that requires scalable discovery process, but a centralized service could be a single point of failure

Related Work

Related Research
OCI's Centralized Repository for Discovery. Open DDS Developer's Guide. http://download.ociweb.com/OpenDDS/OpenDDS-latest.pdf, 2013.
RTI's Enterprise Discovery Protocol. RTI Connext DDS User's Manual. http://community.rti.com/rti-doc/510/ndds.5.1. 0/doc/pdf/RTI_CoreLibrariesAndUtilities_ UsersManual.pdf, 2013.
RTI. Limited-Bandwidth Plug-ins for DDS. http: //www.rti.com/docs/DDS_Over_Low_Bandwidth.pdf, 2011.
J. Sanchez-Monedero, J. Povedano-Molina, J. M ... Filter-based Discovery Protocol for DDS Middlew ...
J. Hoffert, S. Jiang, and D. C. Schmidt. A Taxono ... Ultra-large Scale Systems. In Proceedings of the 45th annual southeast regional conference, pages 355–361. ACM, 2007.

Good for a system having low bandwidth, but this approach requires significant configuration efforts

Related Work

Related Research
OCI’s Centralized Repository for Discovery. Open DDS Developer’s Guide. http:// download.ociweb.com/OpenDDS/OpenDDS-latest.pdf, 2013.
RTI’s Enterprise Discovery Protocol. RTI Connext DDS User’s Manual. http://community.rti.com/rti-doc/510/ndds.5.1. 0/doc/pdf/RTI_CoreLibrariesAnd
RTI. Limited-Bandwidth Plug-ins for DDS. http: //w DDS_Over_Low_Bandwidth.pdf, 2011.
J. Sanchez-Monedero, J. Povedano-Molina, J. M. Lopez-Vega, and J. M. Lopez-Soler. Bloom Filter-based Discovery Protocol for DDS Middleware.
J. Hoffert, S. Jiang, and D. C. Schmidt. A Taxonomy of Discovery Services and Gap Analysis for Ultra-large Scale Systems. In Proceedings of the 45th annual southeast regional conference, pages 355–361. ACM, 2007.

Bloom Filter was used to reduce network and memory usage for discovery, and provided simulation results

Related Work

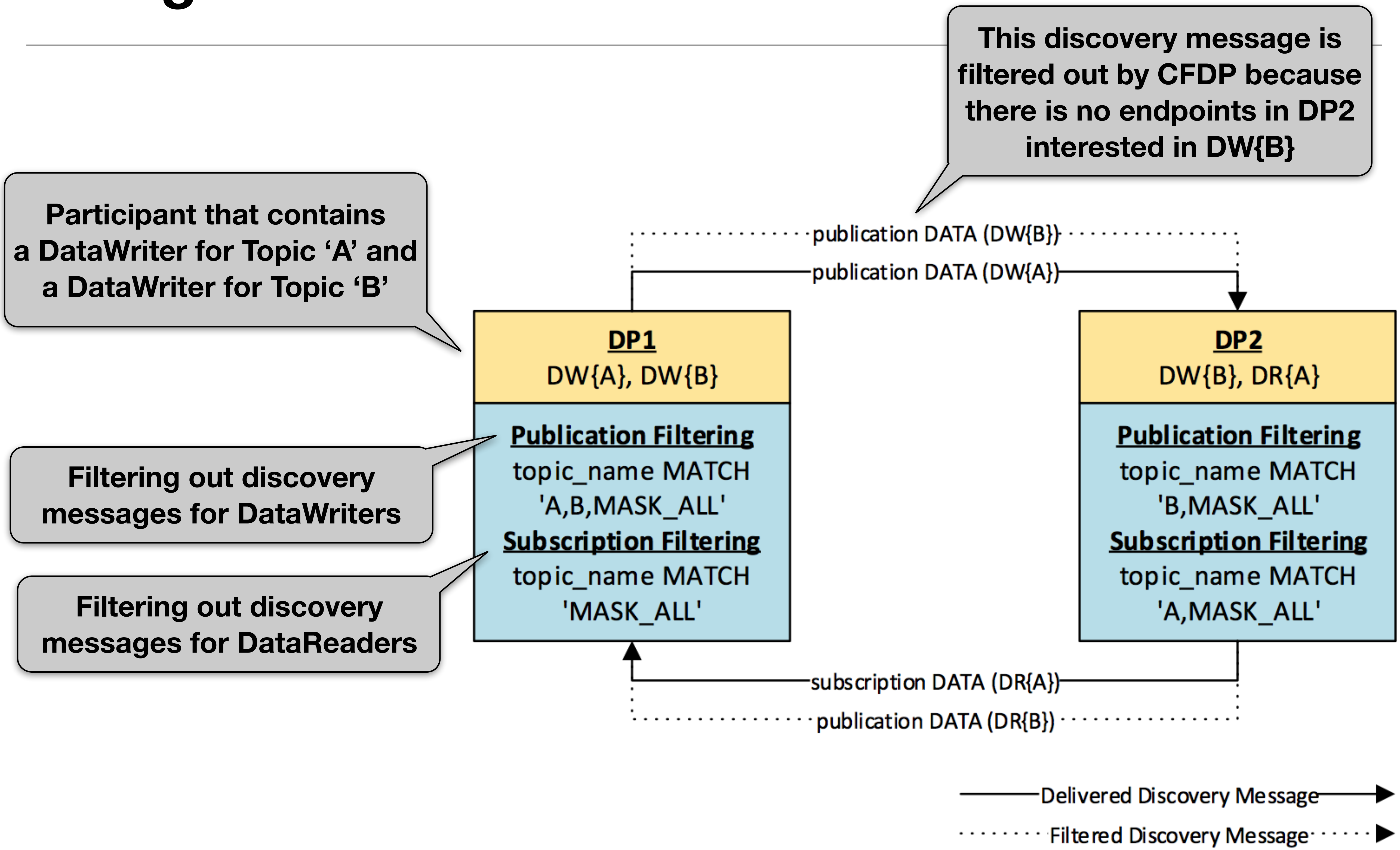
Related Research
OCI's Centralized Repository for Discovery. Open DDS Developer's Guide. http://download.ociweb.com/OpenDDS/OpenDDS-latest.pdf, 2013.
RTI's Enterprise Discovery Protocol. RTI Connext DDS User's Manual. http://community.rti.com/rti-doc/510/ndds.5.1. 0/doc/pdf/RTI_CoreLibrariesAndUtilities_ UsersManual.pdf, 2013.
RTI. Limited-Bandwidth Plug-ins for DDS. http: //www.rti.com/docs/DDS_Over_Low_Bandwidth.pdf, 2011.
J. Sanchez-Monedero, J. Povedano-Molina, J. M. Filter-based Discovery Protocol for DDS Middlewa
J. Hoffert, S. Jiang, and D. C. Schmidt. A Taxonomy of Discovery Services and Gap Analysis for Ultra-large Scale Systems. In Proceedings of the 45th annual southeast regional conference, pages 355–361. ACM, 2007.

Discovery services for ULS introduced including DDS SDP

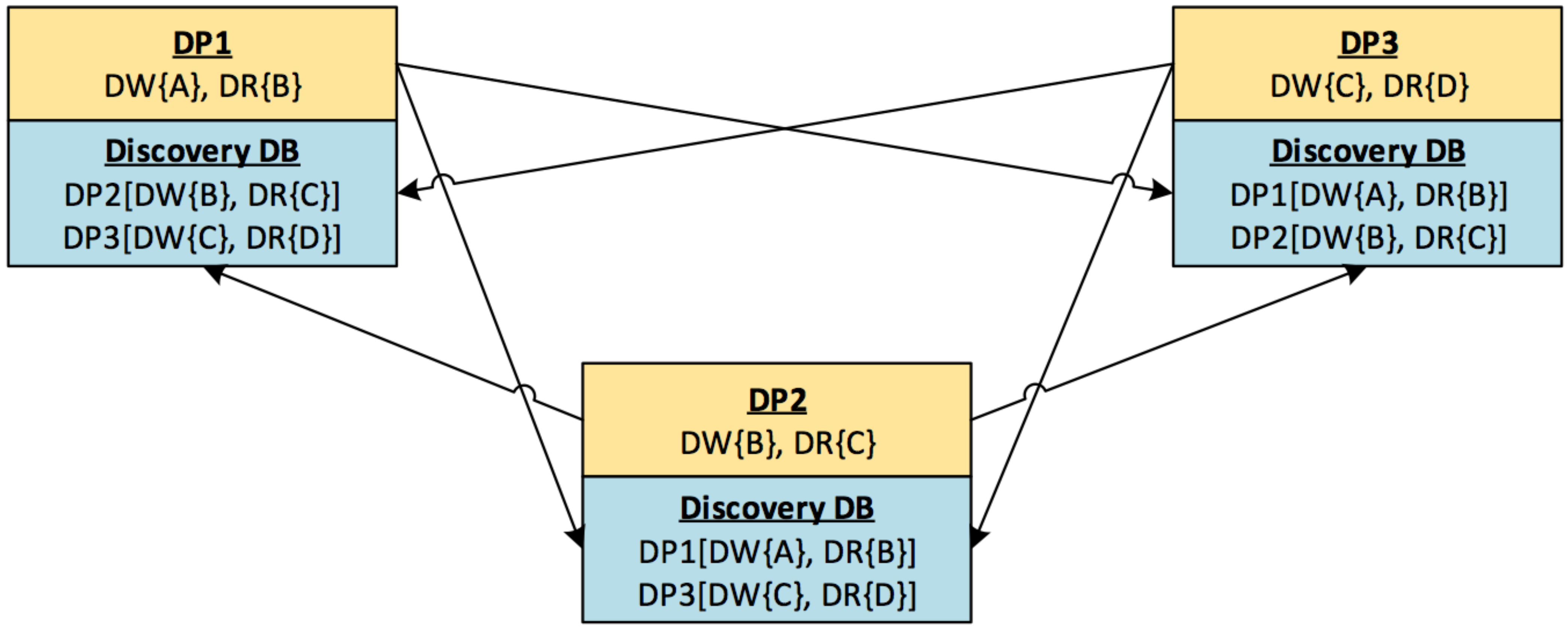
Solution Approach

- A new mechanism for scalable DDS discovery protocol named Content-based Filtering Discovery Protocol (CFDP)
- CFDP employs content-based filtering on the sending peers to filter out unnecessary discovery messages by exchanging filtering expressions that limit the range of interests
- For the prototype implementation, CFDP uses Content Filtered Topic (CFT) for built-in discovery entities

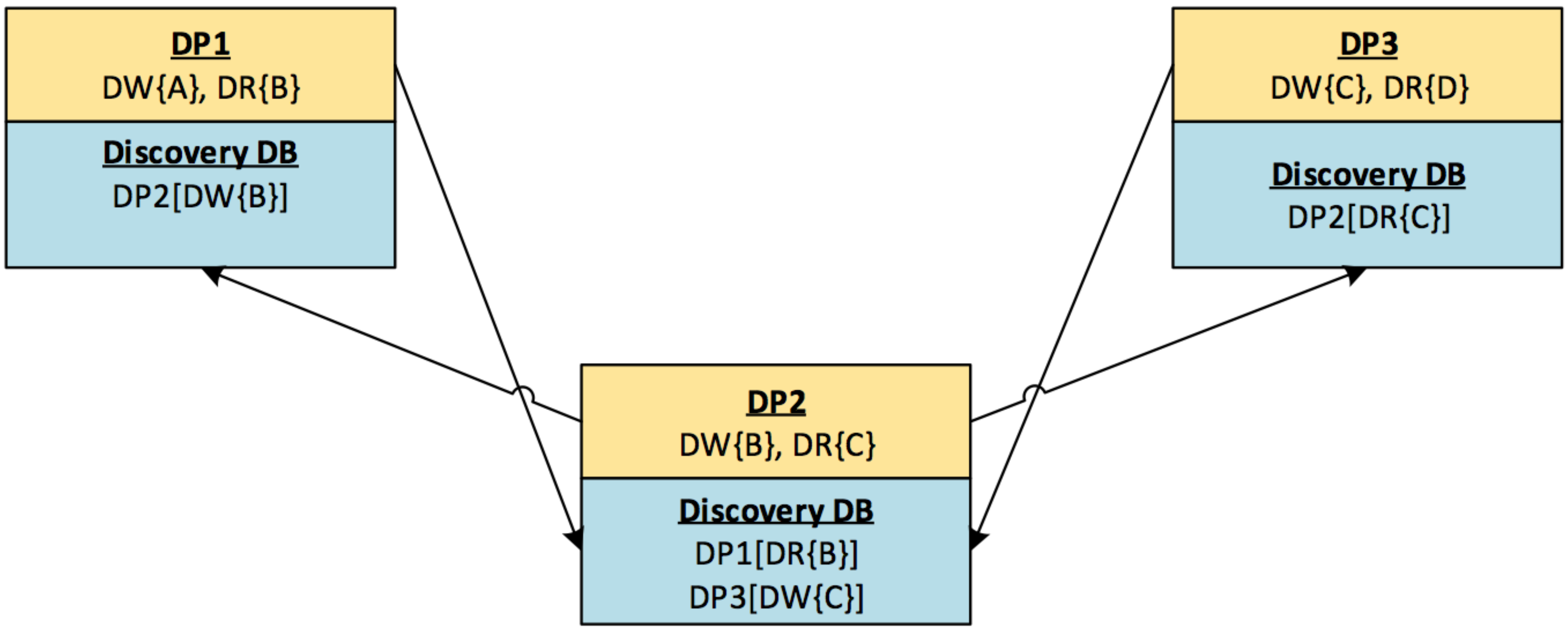
Design of CFDP



SDP Example


DP1
DW{A}, DR{B}
Discovery DB
DP2[DW{B}, DR{C}]
DP3[DW{C}, DR{D}]
DP3
DW{C}, DR{D}
Discovery DB
DP1[DW{A}, DR{B}]
DP2[DW{B}, DR{C}]
DP2
DW{B}, DR{C}
Discovery DB
DP1[DW{A}, DR{B}]
DP3[DW{C}, DR{D}]

CFDP Example



SDP Network Usage with Multicast

Number of endpoints in a system

Cost for announcement

Cost for discovery

Network usage by a participant

$$N_{multi\_participant} = \frac{E}{P} + \frac{E}{P} \cdot (P - 1) \tag{1}$$

Number of participants in a system

$$= \frac{E}{P} + E - \frac{E}{P} \tag{2}$$

$$= E \tag{3}$$

$$N_{multi\_total} = E \cdot P \tag{4}$$

Network usage by a system

CFDP Network Usage with Multicast

Matching ratio by topic names and endpoint types

$$N_{multi\_participant} = \frac{E}{P} + \frac{E}{P} \cdot (P-1) \cdot R \tag{12}$$

$$= \frac{E}{P} + E \cdot R - \frac{E}{P} \cdot R \tag{13}$$

$$= F \cdot (1-R) + E \cdot R \tag{14}$$

$$\sim E \cdot R \tag{15}$$

The number of receiving discovery messages is reduced by the matching ratio through filtering

$$N_{multi\_total} \sim E \cdot P \cdot R \tag{16}$$

SDP and CFDP Network Usage with Unicast

$$
\begin{aligned}
N_{uni\_participant} &= \frac{E}{P} \cdot (P-1) + \frac{E}{P} \cdot (P-1) && (5)\\
&= 2 \cdot \frac{E}{P} \cdot (P-1) && (6)\\
&\because (P-1) \sim P && (7)\\
&\sim 2 \cdot E && (8)
\end{aligned}
$$

$$
\boxed{N_{uni\_total} \sim 2 \cdot E \cdot P} \qquad (9)
$$

$$
\begin{aligned}
N_{uni\_participant} &= \frac{E}{P} \cdot (P-1) \cdot R + \frac{E}{P} \cdot (P-1) \cdot R && (17)\\
&= 2 \cdot \frac{E}{P} \cdot (P-1) \cdot R && (18)\\
&\sim 2 \cdot E \cdot R && (19)
\end{aligned}
$$

$$
\boxed{N_{uni\_total} \sim 2 \cdot E \cdot P \cdot R} \qquad (20)
$$

SDP and CFDP Memory Usage

$$M_{participant} = E \tag{10}$$

$$\boxed{M_{total} = E \cdot P} \tag{11}$$

$$\begin{aligned} M_{participant} &= \frac{E}{P} + \frac{E}{P} \cdot (P-1) \cdot R & (21) \\ &\sim E \cdot R & (22) \end{aligned}$$

$$\boxed{M_{total} \sim E \cdot P \cdot R} \tag{23}$$

Empirical Evaluation

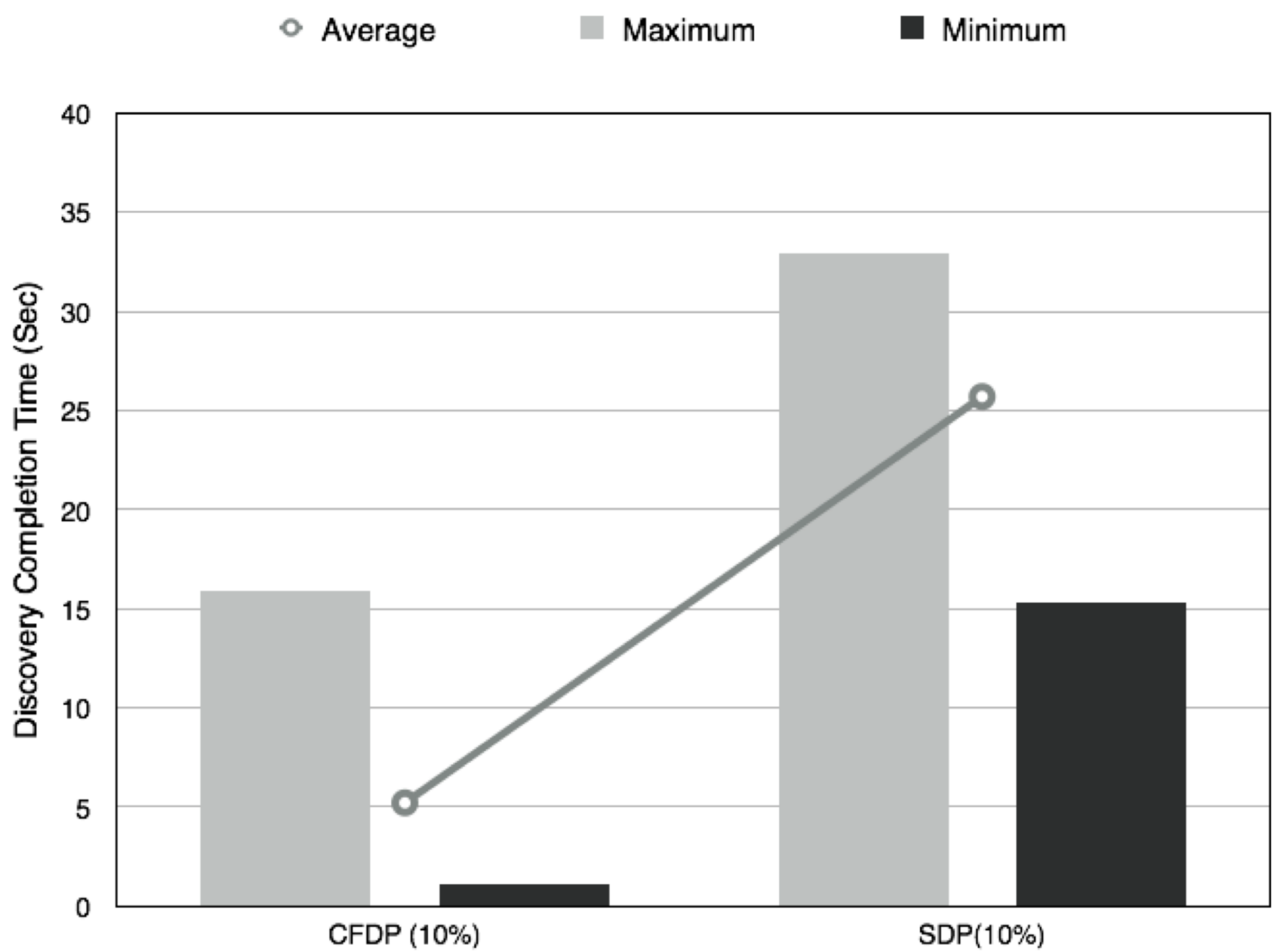
- Discovery Completion Time
  - CFDP vs. SDP (10% matching)
- CPU Usage
  - CFDP vs. SDP (10% matching)
  - CFDP (10%, 30%, 50%)
- Memory & Network Usage
  - CFDP vs. SDP (10%, 50%, 100%)

Empirical Evaluation

- Testbed
  - Six 12-core machines
  - 1Gb Ethernet connected to a single network switch
  - RTI Connext DDS 5.0
- Experiment Setup
  - 480 applications (participants)
  - Each participant has 20 endpoints
  - Default matching ratio is 0.1 (10%)
  - SDP uses multicast and CFDP uses unicast

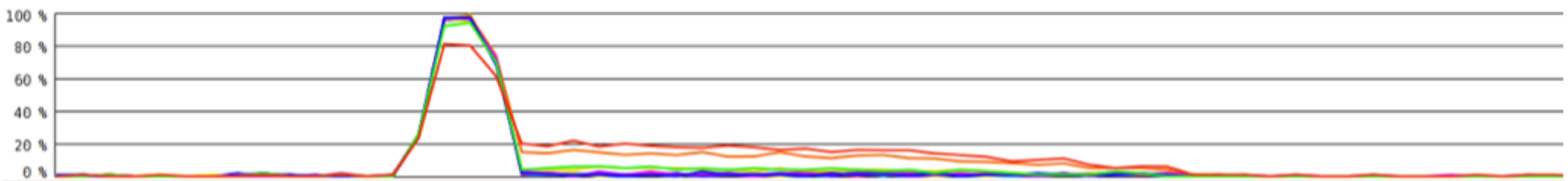
Discovery Completion Time

- **Discovery completion time is defined as the time needed to completely discover all matching endpoints in a domain**



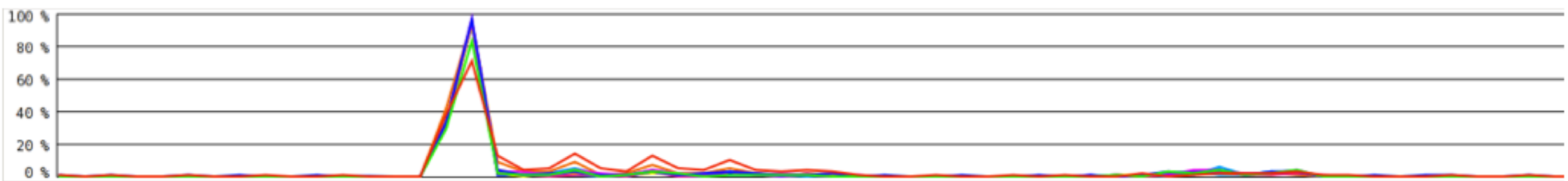
CFDP and SDP CPU Usage

SDP CPU Usage (10% Matching)



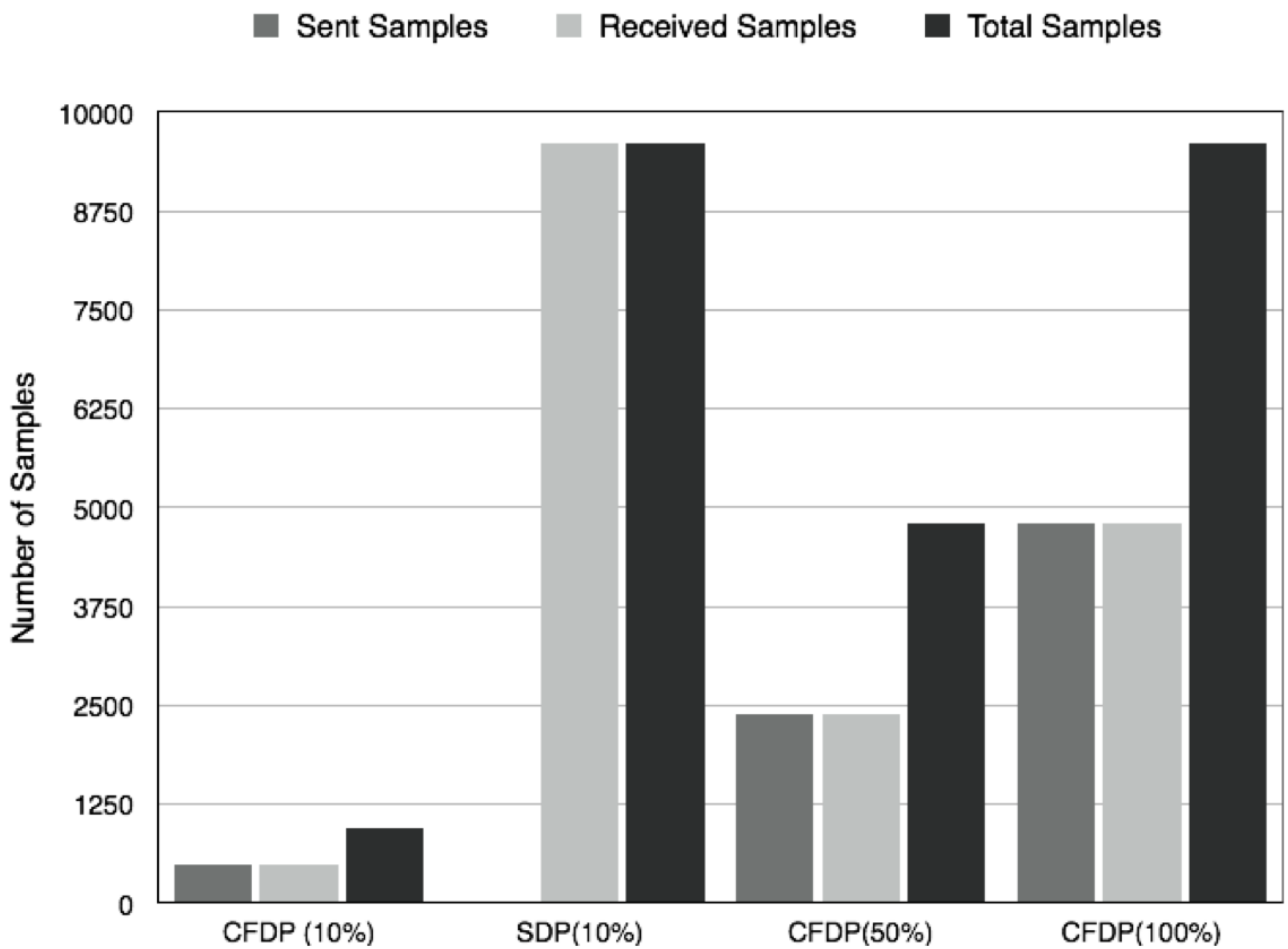
CFDP CPU Usage (10% Matching)

CFDP CPU Usage (30% Matching)



CFDP CPU Usage (50% Matching)

Sent/Received Discovery Messages



Lessons Learned

- CFDP is more efficient and scalable than SDP
- CFDP's current lack of support for multicast can impede scalability
- Instance-based filtering can help to make CFDP scalable in a large-scale system with a small set of topics
- Kyoungho An, Sumant Tambe, Paul Pazandak, Gerardo Pardo-Castellote, Aniruddha Gokhale, and Douglas Schmidt, *"Content-based Filtering Discovery Protocol (CFDP): Scalable and Efficient OMG DDS Discovery Protocol"*, 8th ACM International Conference on Distributed Event-Based Systems (DEBS 2014), Mumbai, India, May 26-29, 2014.

Focus Area 3:
Coordination service for pub/sub middleware in WANs

	Turbine (Machine)	Wind farm (Plant)	Power producer (Enterprise)
Focus Area 1			Fault-tolerant and real-time cloud infrastructures
Focus Area 2		Scalable discovery protocols for pub/sub middleware	
Focus Area 3		Coordination service for pub/sub middleware in WANs	
Focus Area 4		Performance testing for different combinations of QoS configurations	

Context

- The current OMG DDS specification does not define coordination and discovery services for DDS message brokers
- Why DDS message brokers are needed?
  - DDS uses multicast for discovery
  - Network Address Translation (NAT)
  - Network firewalls

Challenges

- Some DDS broker solutions exist
  - DDS Proxy developed by A. Hakiri et al.
  - DDS Routing Service by Real-Time Innovations (RTI)
- A middleware solution to discover and coordinate DDS brokers for internet-scale applications does not exist
- It is challenging to provide scalability and expected latency as well as consistency of dynamic data dissemination paths on overlay networks

Related Work

Related Research
A. Hakiri, P. Berthou, A. Gokhale, D. C. Schmidt, and G. Thierry. Supporting end-to-end scalability and real-time event dissemination in the omg data distribution service over wide area networks. Elsevier Journal of Systems Software (JSS), 2013.
RTI Routing Service. RTI Routing Service user's manual. http://community.rti.com/rti-doc/510/RTI_ Routing_Service_5.1.0/doc/pdf/RTI_Routing_ Service_UsersManual.pdf, 2013.
P. Hunt, M. Konar, F. P. Junqueira, and B. Reed. Zookeeper: wait-free coordination for internet-scale systems. In Proceedings of the 2010 USENIX conferen volume 8, pages 11–11, 2010.
M. Li, F. Ye, M. Kim, H. Chen, and H. Lei. A scalable ar Parallel & Distributed Processing Symposium (IPDPS), 2011 IEEE International, pages 1254–1265. IEEE, 2011.
Amazon Simple Notification Service. http://aws.amazon.com/sns/.

Good for connecting DDS endpoints located in different networks, but requires manual configurations

Related Work

Related Research
A. Hakiria, P. Berthoua, A. Gokhalec, D. C. Schmidtc, and G. Thierrya. Supporting end-to-end scalability and real-time event dissemination in the omg data distribution service over wide area networks. Elsevier Journal of Systems Software (JSS), 2013.
RTI Routing Service. RTI Routing Service user's manual. http://community.rti.com/rti-doc/510/RTI_ Routing_Service_5.1.0/doc/pdf/RTI_Routing_ Service_UsersManual.pdf, 2013.
P. Hunt, M. Konar, F. P. Junqueira, and B. Reed. Zookeeper: wait-free coordination for internet-scale systems. In Proceedings of the 2010 USENIX conference on USENIX annual technical conference, volume 8, pages 11–11, 2010.
M. Li, F. Ye, M. Kim, H. Chen, and H. Lei. A scalable and elastic publish/subscribe service. In Parallel & Distributed Processing Symposium (IPDPS) IEEE, 2011.
Amazon Simple Notification Service. http://aws.amazon.com/sns/.

Good for coordinating for internet-scale distributed systems

Related Work

Related Research
A. Hakiria, P. Berthoua, A. Gokhalec, D. C. Schmidtc, and G. Thierrya. Supporting end-to-end scalability and real-time event dissemination in the omg data distribution service over wide area networks. Elsevier Journal of Systems Software (JSS), 2013.
RTI Routing Service. RTI Routing Service user's manual. http://community … 0/doc/pdf/RTI_Routing_ Service_UsersM …
P. Hunt, M. Kona … -free coordination for internet-scale systems. In Proc … ENIX annual technical conference, volume 8, pages 11–11, 2010.
M. Li, F. Ye, M. Kim, H. Chen, and H. Lei. A scalable and elastic publish/subscribe service. In Parallel & Distributed Processing Symposium (IPDPS), 2011 IEEE International, pages 1254–1265. IEEE, 2011.
Amazon Simple Notification Service. http://aws.amazon.com/sns/.

Good for scalable content-based pub/sub service, but does not support QoS policies for mission critical applications

Related Work

Related Research
A. Hakiria, P. Berthoua, A. Gokhalec, D. C. Schmidtc, and G. Thierrya. Supporting end-to-end scalability and real-time event dissemination in the omg data distribution service over wide area networks. Elsevier Journal of Systems Software (JSS), 2013.
RTI Routing Service. RTI Routing Service user's manual. http://community.rti.com/rti-doc/510/RTI_ Routing_Service_5.1.0/doc/pdf/RTI_Routing_ Service_UsersManual.pdf, 2013.
P. Hunt, M. Konar, F. P. Junqueira, and B. Reed. Zookeeper: wait-free coordination for internet-scale systems. In Proceedings of the 2010 USENIX conference on USENIX annual technical conference, volume 8, pages
M. Li, F. Ye, M. Ki /subscribe service. In Parallel & Distribu national, pages 1254–1265. IEEE, 2011.
Amazon Simple Notification Service. http://aws.amazon.com/sns/.

Good for scalable topic-based pub/sub service, but lack of attribute-based expressiveness

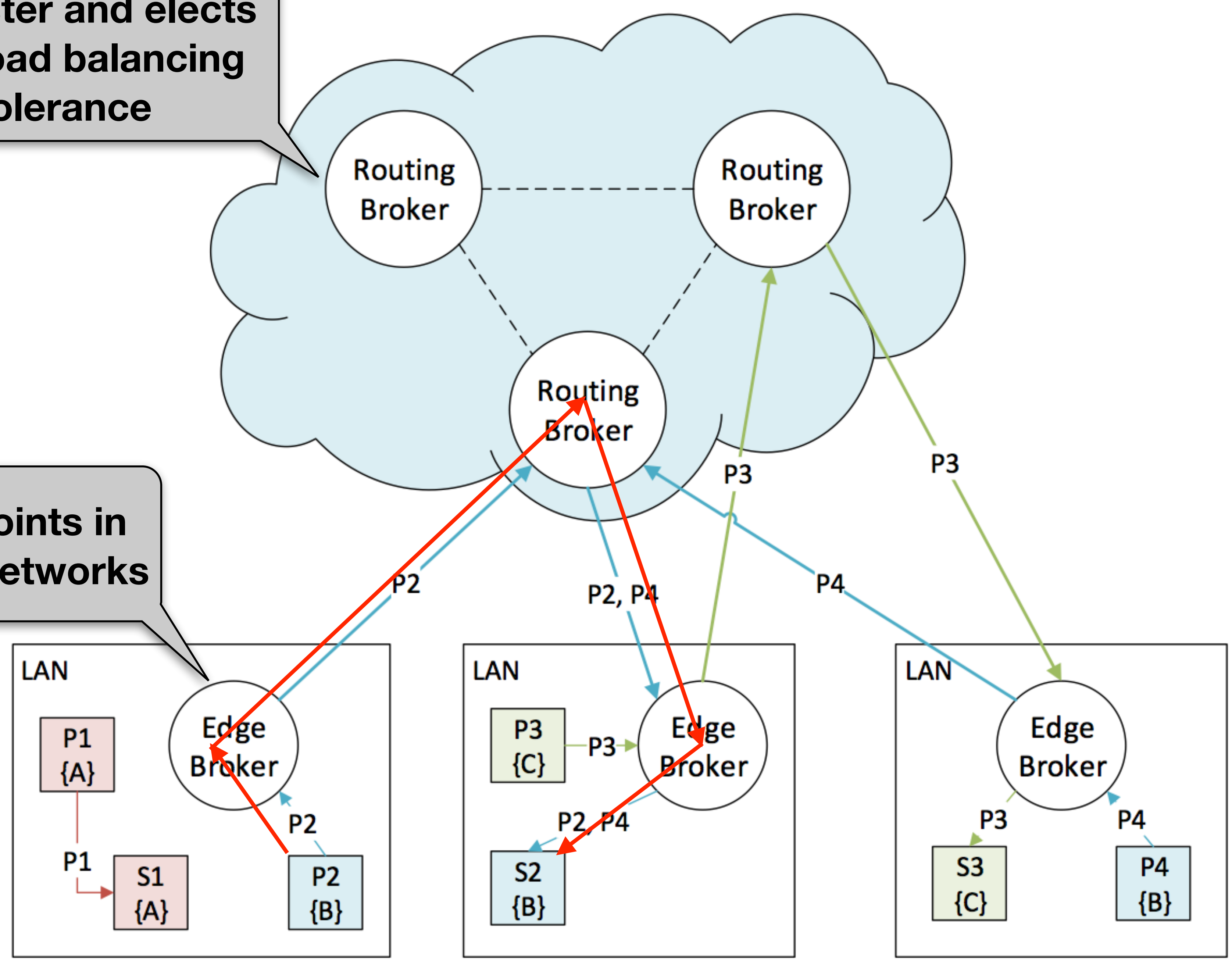
Solution Approach

- PubSubCoord: Cloud-enabled discovery and coordination service for Internet-scale DDS applications
  - Automatic discovery mechanism
  - Mobility support
  - Scalability
  - Load balancing and Fault-tolerance

PubSubCoord Architecture

Connect edge brokers and formed as a cluster and elects a leader to do load balancing and fault-tolerance

Connect endpoints in LANs to other networks



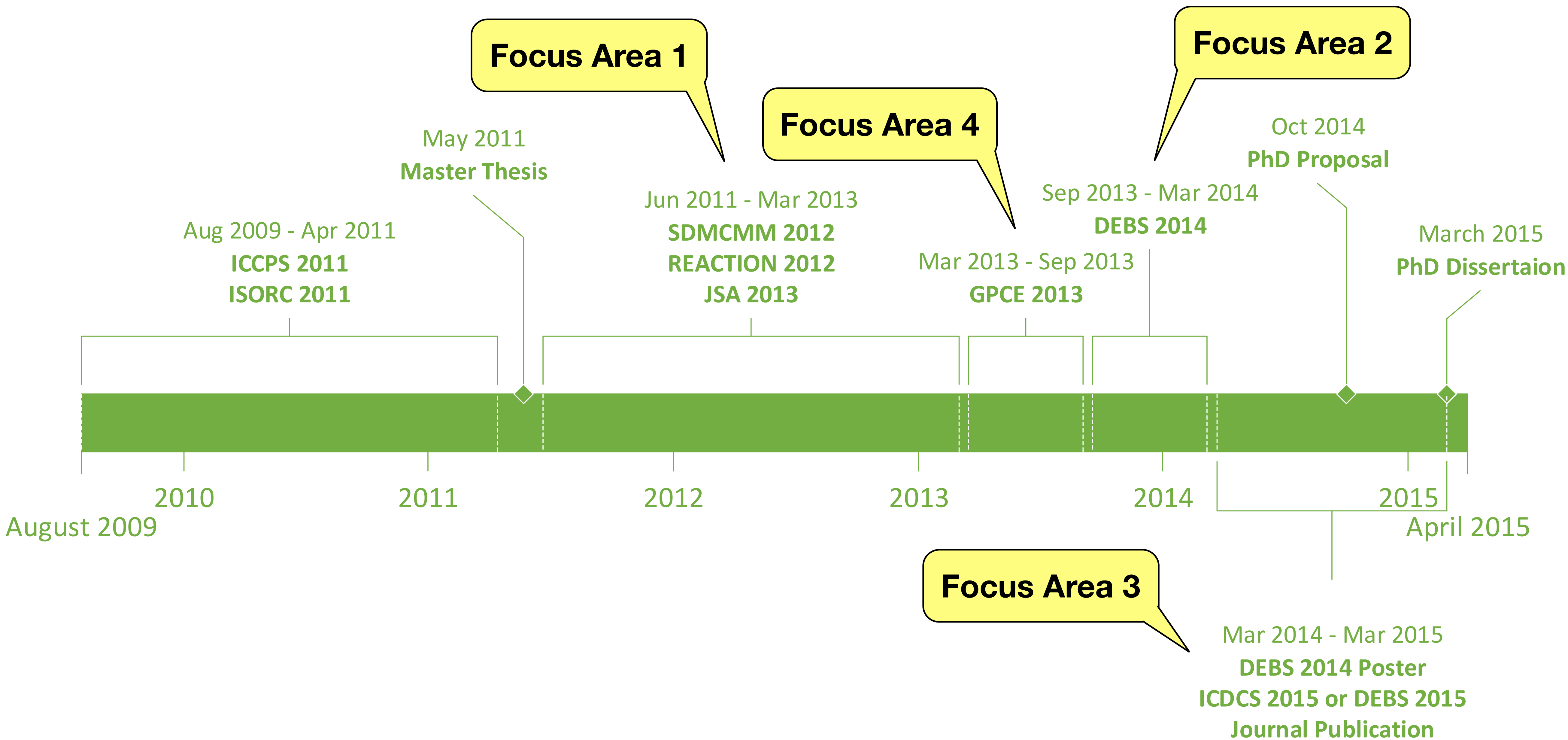
PubSubCoord Architecture

- A two-tier architecture like IBM BlueDove system
  - Edge broker: Directly connected to DDS endpoints in a LAN to behave as a bridge to other networks
  - Routing broker: Links to edge brokers to deliver data between edge brokers
  - Reduces the need for maintaining states for edge brokers
  - Failed brokers do not affect others
  - Routing brokers may be overloaded, but can be scaled by cloud infrastructures

Lessons Learned

- This paper presents preliminary work on a cloud-enabled coordination service for internet-scale DDS applications that supports
  - Scalability
  - Load balancing and fault-tolerance
  - Endpoint mobility
- Experiments will be done as future work to validate this solution approach
- Kyoungho An and Aniruddha Gokhale, "*A Cloud-enabled Coordination Service for Internet-scale OMG DDS Applications*", Poster paper at the 8th ACM International Conference on Distributed Event-Based Systems (DEBS 2014), Mumbai, India, May 26-29, 2014.

Dissertation Timeline



Focus Areas in Industrial Internet Systems

Done | In Progress | Preliminary

	Turbine (Machine)	Wind farm (Plant)	Power producer (Enterprise)
Focus Area 1			Fault-tolerant and real-time cloud infrastructures
Focus Area 2		Scalable discovery protocols for pub/sub middleware	
Focus Area 3		Coordination service for pub/sub middleware in WANs	
Focus Area 4		Performance testing for different combinations of QoS configurations	

Proposed Work

- Focus Area 3 - PubSubCoord
  - Complexity analysis and empirical evaluation
  - Deadline-aware overlay network
- Focus Area 2 - CFDP
  - Instance-based Filtering
  - Multi-channel enabled Filtering

Summary of Publications

Journal Publications

F1 1. Kyoungho An, Shashank Shekhar, Faruk Caglar, Aniruddha Gokhale, and Shivakumar Sastry, A Cloud Middleware for Assuring Performance and High Availability of Soft Real-time Applications, The Elsevier Journal of Systems Architecture (JSA): Embedded Systems Design, 2014.

M **Book Chapters**

2. Kyoungho An, Adam Trewyn, Aniruddha Gokhale and Shivakumar Sastry, Design and Transformation of Domain-specific Language for Reconfigurable Conveyor Systems, Book chapter in Formal and Practical Aspects of Domain-Specific Languages: Recent Developments, IGI Global publishers, Editor: Marjan Mernik, 2012.

First Author
F1: Focus Area 1
F2: Focus Area 2
F3: Focus Area 3
F4: Focus Area 4
M: Master's Thesis
G: Grand Challenge Problem

Summary of Publications

Conference & Symposium Publications

F2 3. Kyoungho An, Sumant Tambe, Paul Pazandak, Gerardo Pardo-Castellote, Aniruddha Gokhale, and Douglas Schmidt, Content-based Filtering Discovery Protocol (CFDP): Scalable and Efficient OMG DDS Discovery Protocol, 8th ACM International Conference on Distributed Event-Based Systems (DEBS 2014), Mumbai, India, May 26-29, 2014.

F4 4. Kyoungho An, Takayuki Kuroda, Aniruddha Gokhale, Sumant Tambe, and Andrea Sorbini, Model-driven Generative Framework for Automated DDS Performance Testing in the Cloud, 12th ACM International Conference on Generative Programming: Concepts & Experiences (GPCE 2013), Indianapolis, IN, Oct 27-28, 2013.

F1 5. Kyoungho An, Resource Management and Fault Tolerance Principles for Supporting Distributed Real-time and Embedded Systems in the Cloud, 9th Middleware Doctoral Symposium (MDS 2012), co-located with ACM/IFIP/USENIX 13th International Conference on Middleware (Middleware 2012), Montreal, Quebec, Canada, Dec 3-7, 2012.

M 6. Kyoungho An, Adam Trewyn, Aniruddha Gokhale and Shivakumar Sastry, Model-driven Performance Analysis of Reconfigurable Conveyor Systems used in Material Handling Applications, Second ACM/IEEE International Conference on Cyber Physical Systems (ICCPS 2011), Chicago, IL, Apr 11-14, 2011.

M 7. Anushi Shah, Kyoungho An, Aniruddha Gokhale and Jules White, Maximizing Service Uptime of Smartphone-based Distributed Real-time and Embedded Systems, 14th IEEE International Symposium on Object/Component/Service-oriented Real-time Distributed Computing (ISORC 2011), Newport Beach, CA, Mar 28-31, 2011.

Summary of Publications

Workshop, Work in Progress, and Poster Publications

F3 8. Kyoungho An and Aniruddha Gokhale, A Cloud-enabled Coordination Service for Internet-scale OMG DDS Applications, Poster paper at the 8th ACM International Conference on Distributed Event-Based Systems (DEBS 2014), Mumbai, India, May 26-29, 2014.

F4 9. Shashank Shekhar, Faruk Caglar, Kyoungho An, Takayuki Kuroda, Aniruddha Gokhale and Swapna Gokhale, A Model-driven Approach for Price/Performance Tradeoffs in Cloud-based MapReduce Application Deployment, MODELS 2013 workshop on Model-Driven Engineering for High Performance and CLoud computing (MDHPCL 2013), Miami, FL, Sep 29, 2013.

F4 10. Kyoungho An and Aniruddha Gokhale, Model-driven Performance Analysis and Deployment Planning for Real-time Stream Processing, Work-in-Progress (WiP) session at 19th IEEE Real-time and Embedded Technology and Applications Symposium (RTAS 2013), Philadelphia PA, Apr 9-11, 2013.

F1 11. Faruk Caglar, Shashank Shekhar, Kyoungho An and Aniruddha Gokhale, WiP Abstract: Intelligent Power- and Performance-aware Tradeoffs for Multicore Servers in Cloud Data Centers, Work-in-Progress (WiP) session at 4th ACM/IEEE International Conference on Cyber Physical Systems (ICCPS 2013), Philadelphia PA, Apr 9-11, 2013.

F1 12. Kyoungho An, Faruk Caglar, Shashank Shekhar and Aniruddha Gokhale, A Framework for Effective Placement of Virtual Machine Replicas for Highly Available Performance-sensitive Cloud-based Applications, RTSS 2012 workshop on Real-time and Distributed Computing in Emerging Applications (REACTION 2012), San Juan, Puerto Rico, Dec 4-7, 2012.

Summary of Publications

F1 13. Kyoungho An, Subhav Pradhan, Faruk Caglar and Aniruddha Gokhale, A Publish/Subscribe Middleware for Dependable and Real-time Resource Monitoring in the Cloud, Middleware 2012 workshop on Secure and Dependable Middleware for Cloud Monitoring and Management (SDMCMM 2012), Montreal, Quebec, Canada, Dec 3-7, 2012.

F1 14. Kyoungho An, Strategies for Reliable, Cloud-based Distributed Real-time and Embedded Systems, Extended abstract for PhD Forum in 31st IEEE International Symposium on Reliable Distributed Systems (SRDS 2012), Irvine, CA, Oct 8-11, 2012.

F4 15. Faruk Caglar, Kyoungho An, Aniruddha Gokhale and Tihamer Levendovszky, Transitioning to the Cloud? A Model-driven Analysis and Automated Deployment Capability for Cloud Services, MODELS 2012 workshop on Model-Driven Engineering for High Performance and CLoud computing (MDHPCL 2012), Innsbruck, Austria, Sep 30 - Oct 5, 2012.

Technical Reports

G 16. Shweta Khare, Sumant Tambe, Kyoungho An, Aniruddha Gokhale, and Paul Pazandak, Scalable Reactive Stream Processing Using DDS and Rx: An Industry-Academia Collaborative Research Experience, ISIS Technical Report, no. ISIS-14-103: Institute for Software Integrated Systems, Vanderbilt University, Nashville TN, April, 2014.

G 17. Kyoungho An, Sumant Tambe, Andrea Sorbini, Sheeladitya Mukherjee, Javier Povedano-Molina, Michael Walker, Nirjhar Vermani, Aniruddha Gokhale, and Paul Pazandak, Real-time Sensor Data Analysis Processing of a Soccer Game Using OMG DDS Publish/Subscribe Middleware, ISIS Technical Report, no. ISIS-13-102: Institute for Software Integrated Systems, Vanderbilt University, Nashville TN, June, 2013.

Thank you! Any Questions?