Quorum Briefing
Lockheed Martin Corporation

David Lounsbury
Vice President
Open Group Advanced Research
http://www.opengroup.org/ar

THE Open GROUP

Vision

- Technical Challenge:
  - Allow **reliable** processing of critical, high-value, time-urgent data in networked systems using commercial off-the-shelf components
- Business Vision:
  - Participate in software research and advanced development which will enhance customer's ability to provide highly-available, high-capacity IT services
  - Serve as a consulting and technology transfer resource to allow commercial and government customers to use DARPA-sponsored technologies

The QoS Challenge

- The Internet provides the opportunity to conduct business at vastly increased scales using a shared-cost infrastructure
- However, to take advantage of this opportunity, companies are “increasingly dependent on large-scale distributed systems that operate in unbounded network environments”(IEEE *Internet Computing* 11/99)

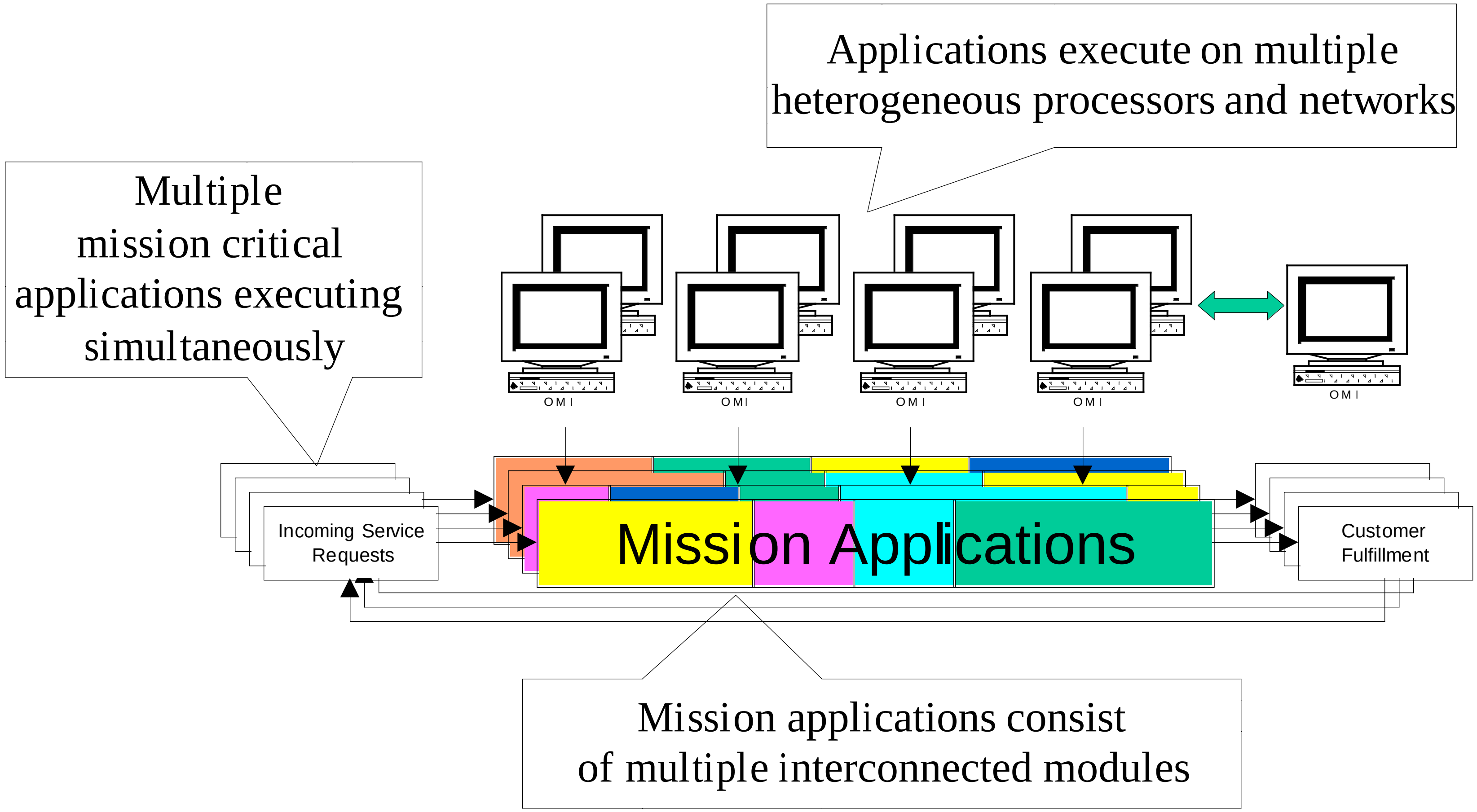
THE Open GROUP

QoS Opportunity

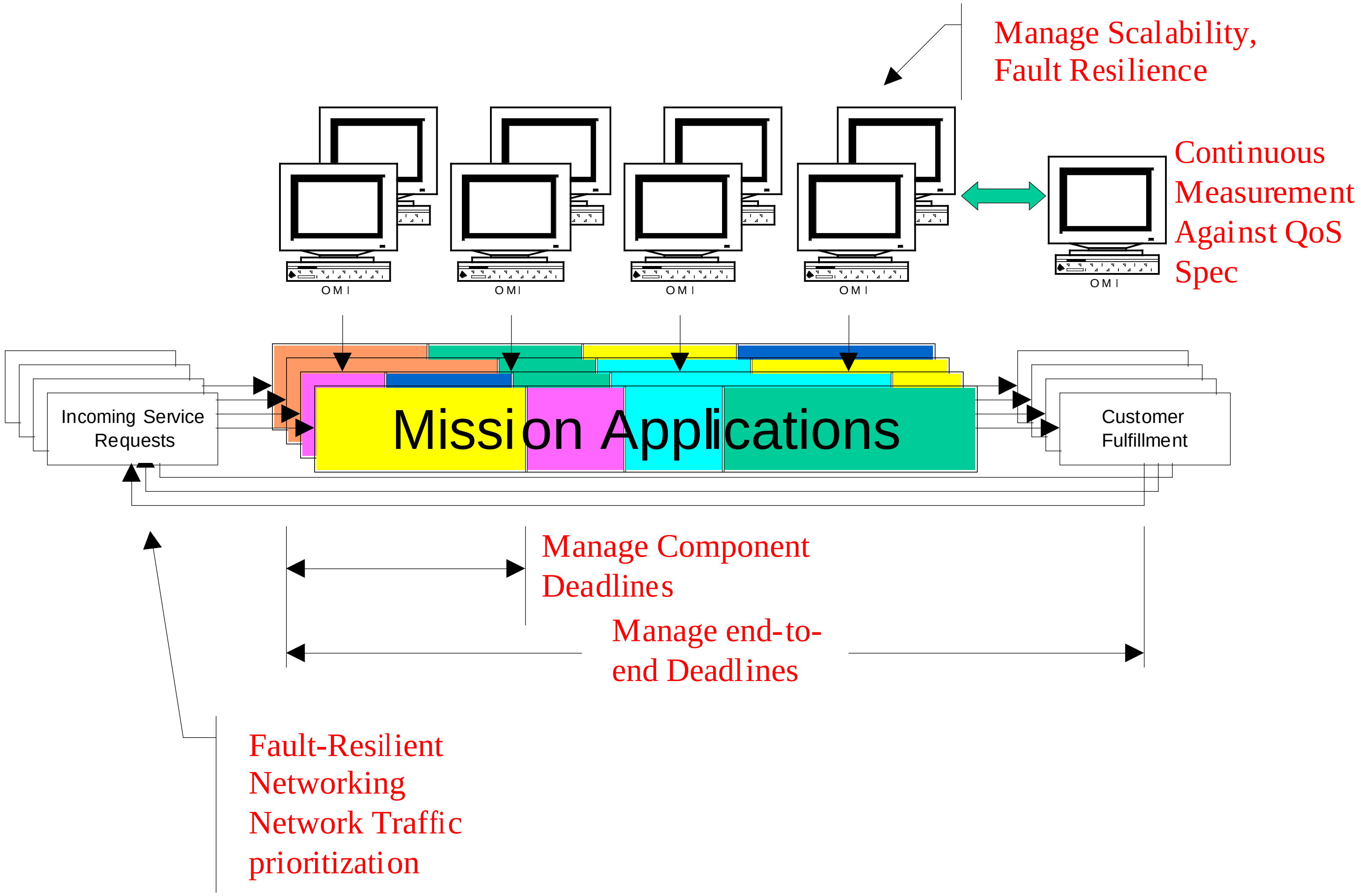
- As value of transactions on unbounded networks grow, companies will seek guarantees of dependability, performance, and efficiency for distributed applications and networks.
- To provide adequate levels of service to customers, companies need same level of assured operation as they got from the mainframe "Glass House"
  - End-to-end performance
  - Availability and Fault Resilience
  - Adaptivity to changing load and network conditions

THE Open GROUP

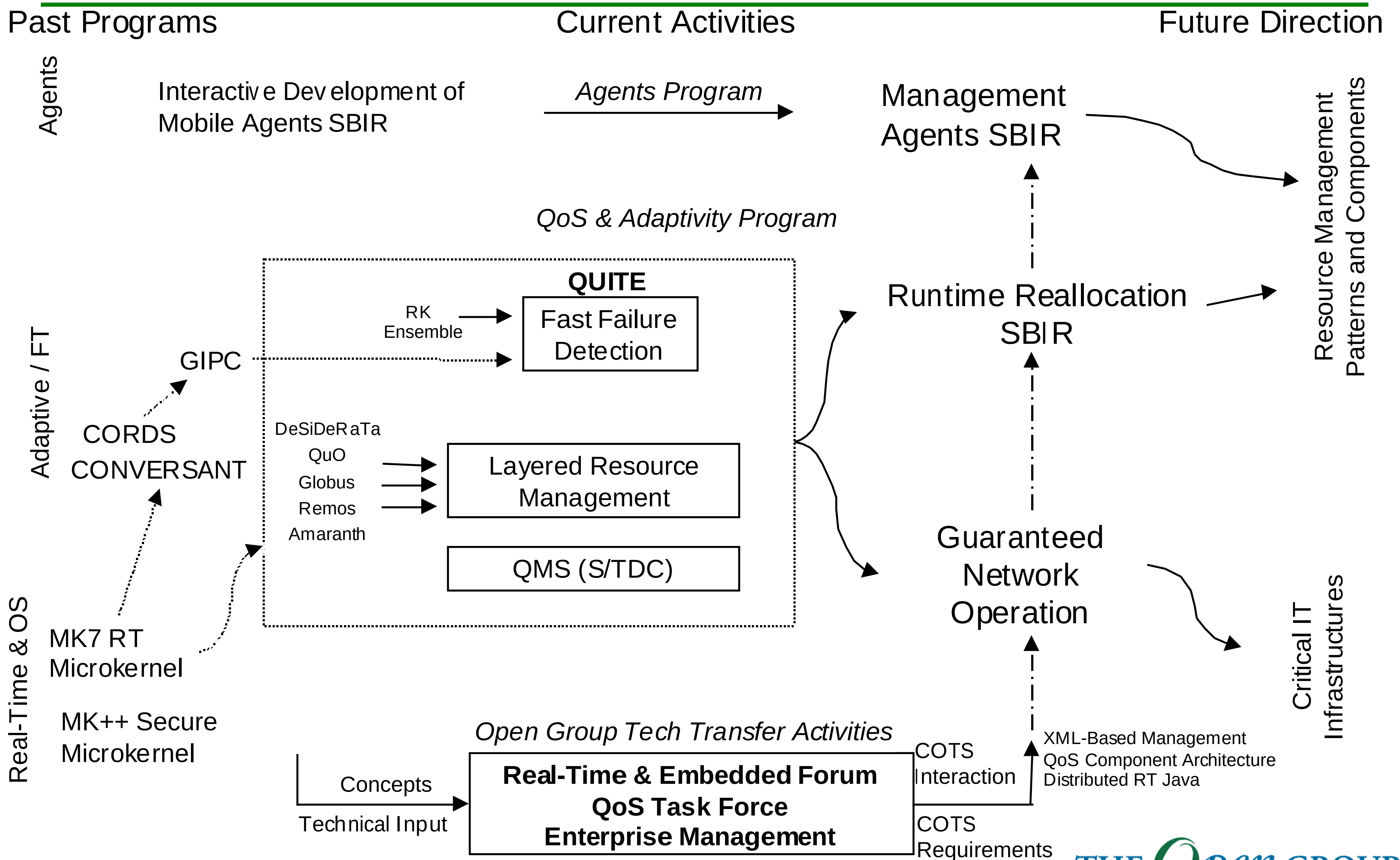
The Managed QoS Environment



Managed QoS Capabilities



Advanced Research Roadmap



Advanced Research Experience

The Open Group Advanced Research offers a 10+ year track record of developing and deploying innovative solutions to the problems of high availability systems and networks from a successful team of experienced engineers.

Problem Domain	Technologies	Customers
Commercial OS Technology Distributed Real-Time Security	MK++ AD3 MkLinux	Hewlett-Packard IBM Honeywell Space Systems DASCOM Apple
End-User/ Technology Transfer	Air Force AWACS Navy Aegis	NSWC JHU-APL Locheed-Martin Hewlett-Packard
Real-Time Protocols and Group Communication	CORDS GIPC SHAWS	NSWC DASCOM Honeywell Space Systems Novell

Going Forward with The Open Group

We seek strategic relationships with customers with needs for assured end to end system QoS, e.g:

- QoS requirements definition
- Joint design and trade off studies
- Advanced development/ pre-production test beds
- Acquisition and integration of commercial QoS related software
- Development of QoS software required for effective end to end system QoS design and implementation

Research Areas

Quorum Integration, Testbed, and Exploitation (QUITE)

Group Communications

Fault Management

Layered Resource Management

Adaptive Applications

Technology Transfer

QUITE

- Integration of 40+ QoS research projects sponsored under DARPA Quorum program
- Quorum program goal: develop innovative software-based approaches to end-to-end QoS
- QUITE provides testbed, characterizes and combines promising research results, transfers technology to government and commercial markets

Architectural Patterns Explored within Quorum (i)

- Adaptive Application
  - An application that can operate using differing algorithms and/or strategies based on the sets of resources that are available.
  - QuO, Quasar, HPF, Linux/RK, RT-ARM
- Application Path
  - An execution sequence that requires a particular set of resources to execute successfully. (A POSIX thread is a special case of this abstraction for CPU usage.)
  - DeSiDeRaTa, Sesco, CORDS/GIPC

Architectural Patterns Explored in Quorum (ii)

- Resource Management Components
  - The extraction of resource usage strategy from individual applications into a separate component in support of a more comprehensive strategy in utilizing available resources.
  - DeSiDeRaTa, Sesco, Globus
- Fault Management
  - The extraction of information about failures and failure dependencies into a separate component in support of a more comprehensive strategy in handling failures and in predicting future failures.
  - FFD

Architectural Patterns Explored in Quorum (iii)

- QoS Property Factoring
  - The structuring of applications based on the QoS requirements of individual subcomponents.
  - AQuA, HPF, Quasar, Darwin, Linux/RK
- Scalable Fault Tolerance
  - The parallelization of application algorithms in support of scalability and fault tolerance.
  - Resource Management, Group Comms

Design Patterns Explored within Quorum (i)

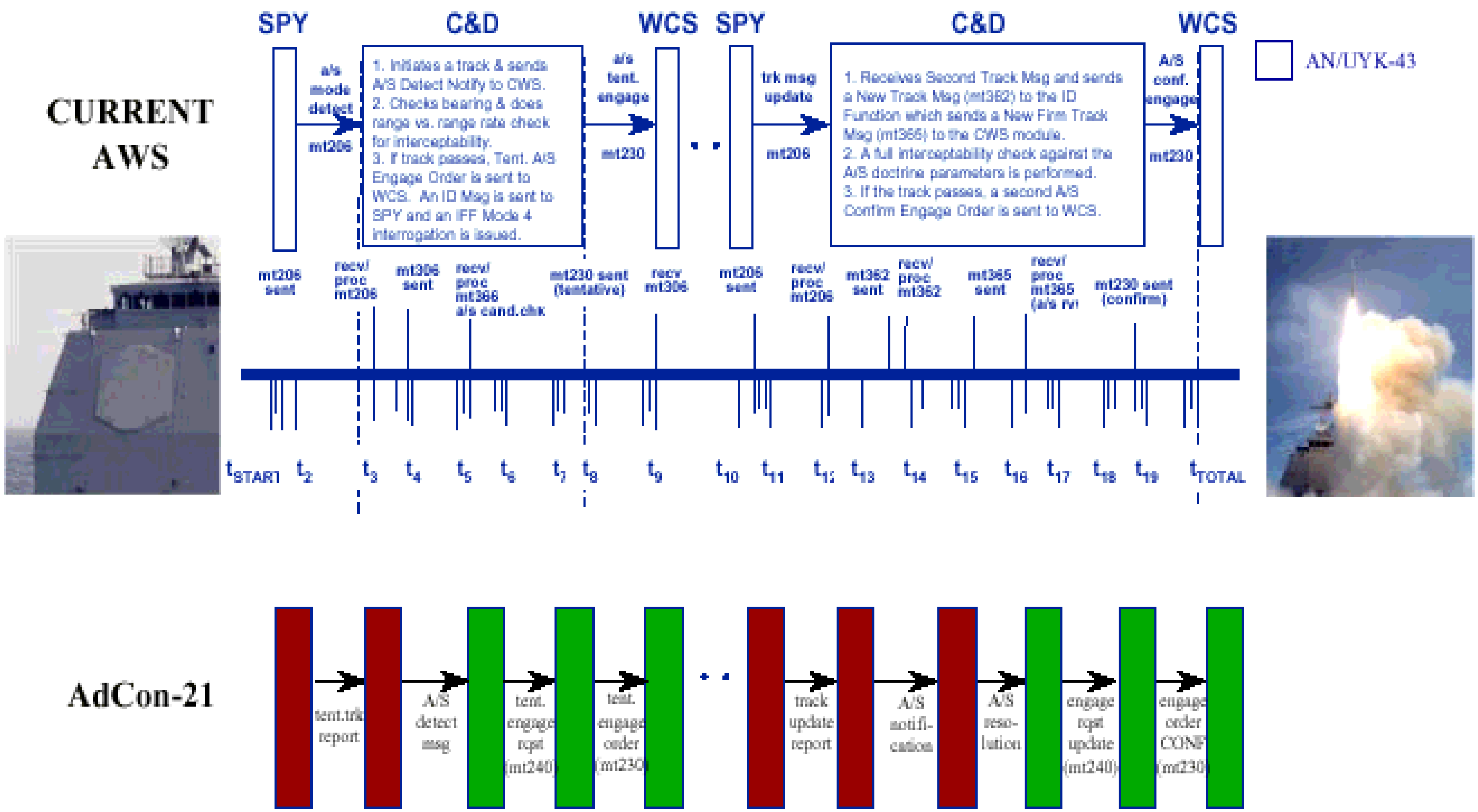
- Layered Resource Management
  - The separation of resource management into multiple components within a hierarchy in support of scalable systems.
  - RT-ARM, Sesco (w/ enhancements),DeSiDeRaTa, Globus
- Group Communications
  - A method for reliably communicating multicast messages in support of scalability and fault tolerance.
  - Ensemble, CORDS/GIPC, Cactus, Armada

Design Patterns Explored within Quorum (ii)

- Integrated Instrumentation
  - Use of dynamically execution status for the purpose of adaptively assigning resources towards the most effective use.
  - QMS, DeSiDeRaTa, Remos
- Real-Time Middleware
  - Middleware that propagates guarantees on QoS properties from lower levels, such as OS and hardware.
  - TAO, CORDS/GIPC, Java/RK

Group Communications

Why DD-21 Needs Assured Response: SPY Radar Auto-Special Time-Line



Group Communications

- Reliable multicast technique based on atomic multicast—each message is reliably delivered to either (exclusive or)
  - —all designated recipients
  - —no recipients
- Popularized by Ken Birman at Cornell U.
  - Initial research/product was Isis
  - Current implementation is Ensemble

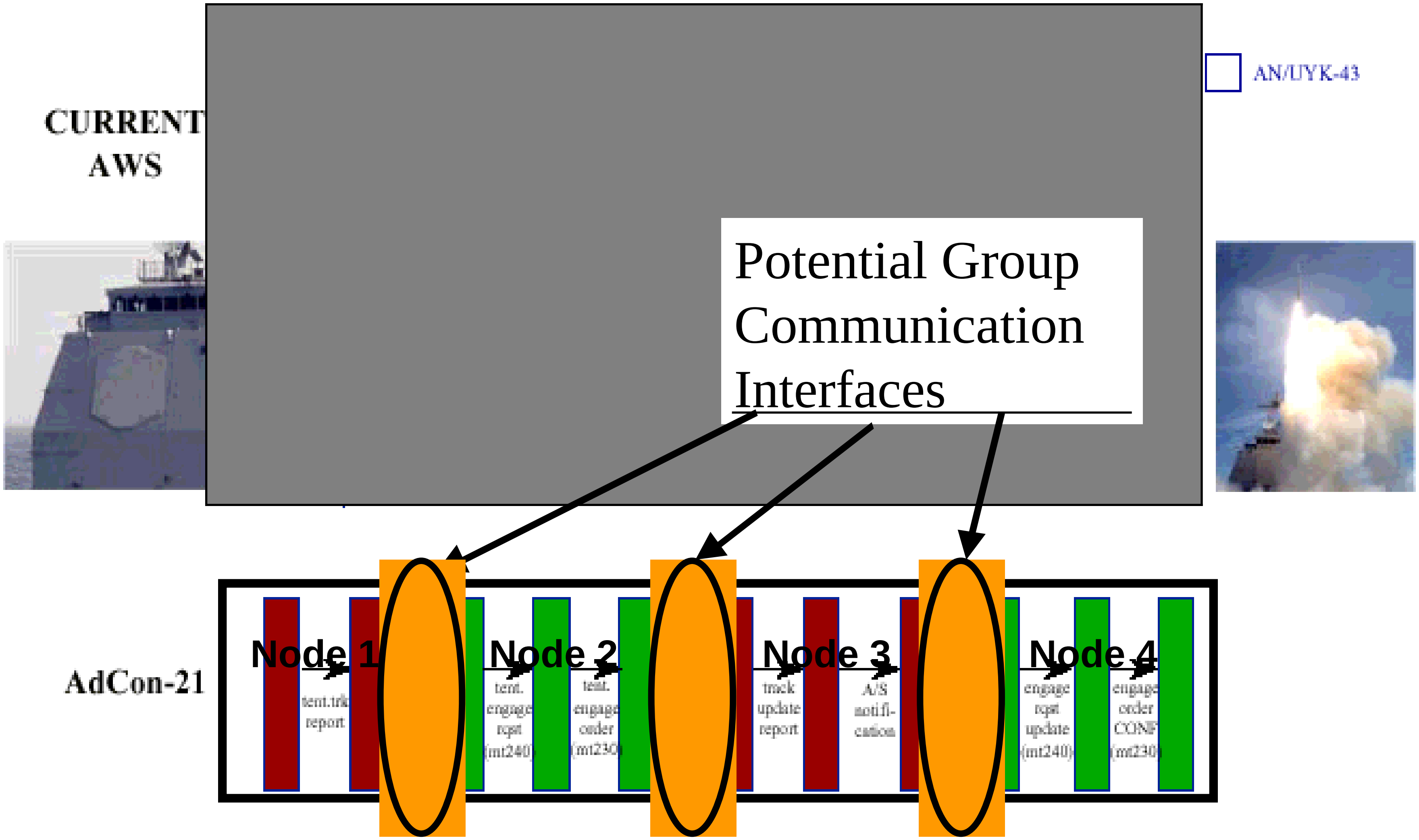
(Simplified) Overview of Group Communication Operation

- Start with known set of group members
- Message is sent (multicast) to agent on host node of each recipient
- Receipt acknowledgements are exchanged
- When all nodes have acknowledged, release message to each application group member
- Otherwise—after a time-out event occurs
  - Reform group by ejecting tardy members
  - Restart message delivery process with new group membership set

Observations

- Use of time-out is derived from requirement that timeliness is more important than tardy operation at full capacity
- Time-out event transforms timing fault into an (apparent) component failure
- Individual message delivery time-outs typically must operate an order of magnitude faster than overall system time constraint
- Example
  - End-to-end 1 second deadline might require 0.1 second time-out at each stage of group communications

Why DD-21 Needs Assured Response: SPY Radar Auto-Special Time-Line



Problems in Utilizing Real-Time Group Communications

- Time constraints in RT systems must be met even in extreme conditions, not just in speed-of-light micro-benchmarks
- Group communication time-out periods are often of same order of magnitude as scheduling jitter in non-RT OS's
- False positives (tardy nodes declared dead), while handled correctly, are expensive
  - Node is forced "down," then allowed to rejoin
  - Requires reacquisition of application state
- COTS components (Isis, Ensemble) not designed using real-time techniques

More Problems in Real-Time Group Communications

- Different interfaces have different timing constraints. A node may be declared down in one context, but must remain “up” in another.
  - Notional interface time-out periods
    - HiPer-D AAW path: 0.5 second
    - Instrumentation: 3 seconds
    - Resource Management 10 seconds
  - Timing constraints (and time-outs) are usually associated with an interface to an external component—not an entire application
  - Note: this problem is not limited to group communication interfacess

CORDS and GIPC

- CORDS: A framework for constructing high-performance, real-time communication protocols
- GIPC: A protocol (built using CORDS) which offers real-time group communication services
  - All members of group are guaranteed to receive messages in identical order
  - Rapid recovery from failure of group member
- Group-ordered communications + Layered Resource Management = Scalable, Fault-tolerant systems

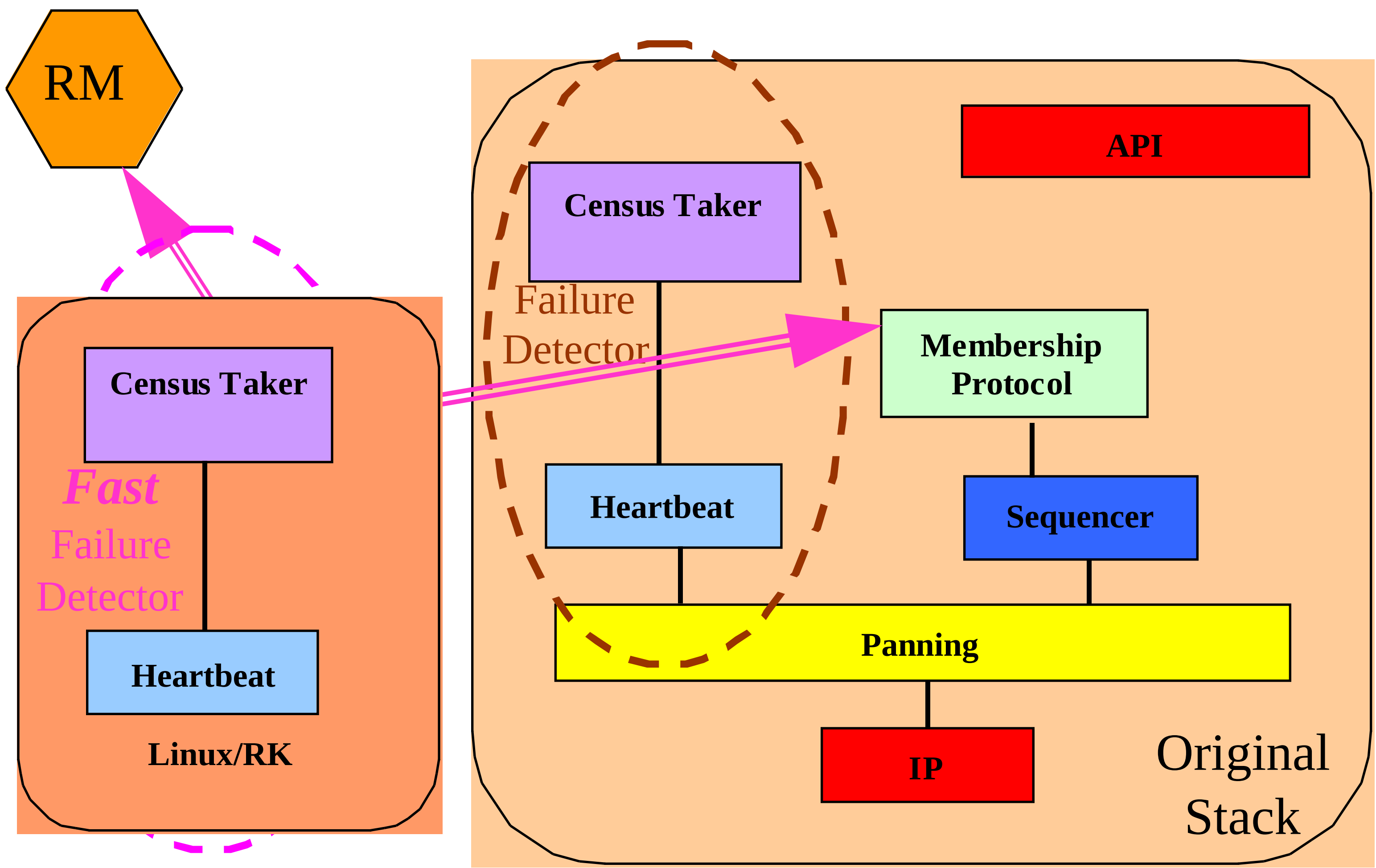
Fault Management

Fast Failure Detector (FFD) Objectives

- General Goal of FFD:
  - Provide faster, more reliable detection of host node failure than other components
- Specific Goal of FFD Integration Effort:
  - Detect and report host failure within 250 msec
  - This should allow an application to recover from a host node failure within 1 second, even with a substantial state reacquisition cost

THE Open GROUP

Group Membership Protocol Stack



THE Open GROUP

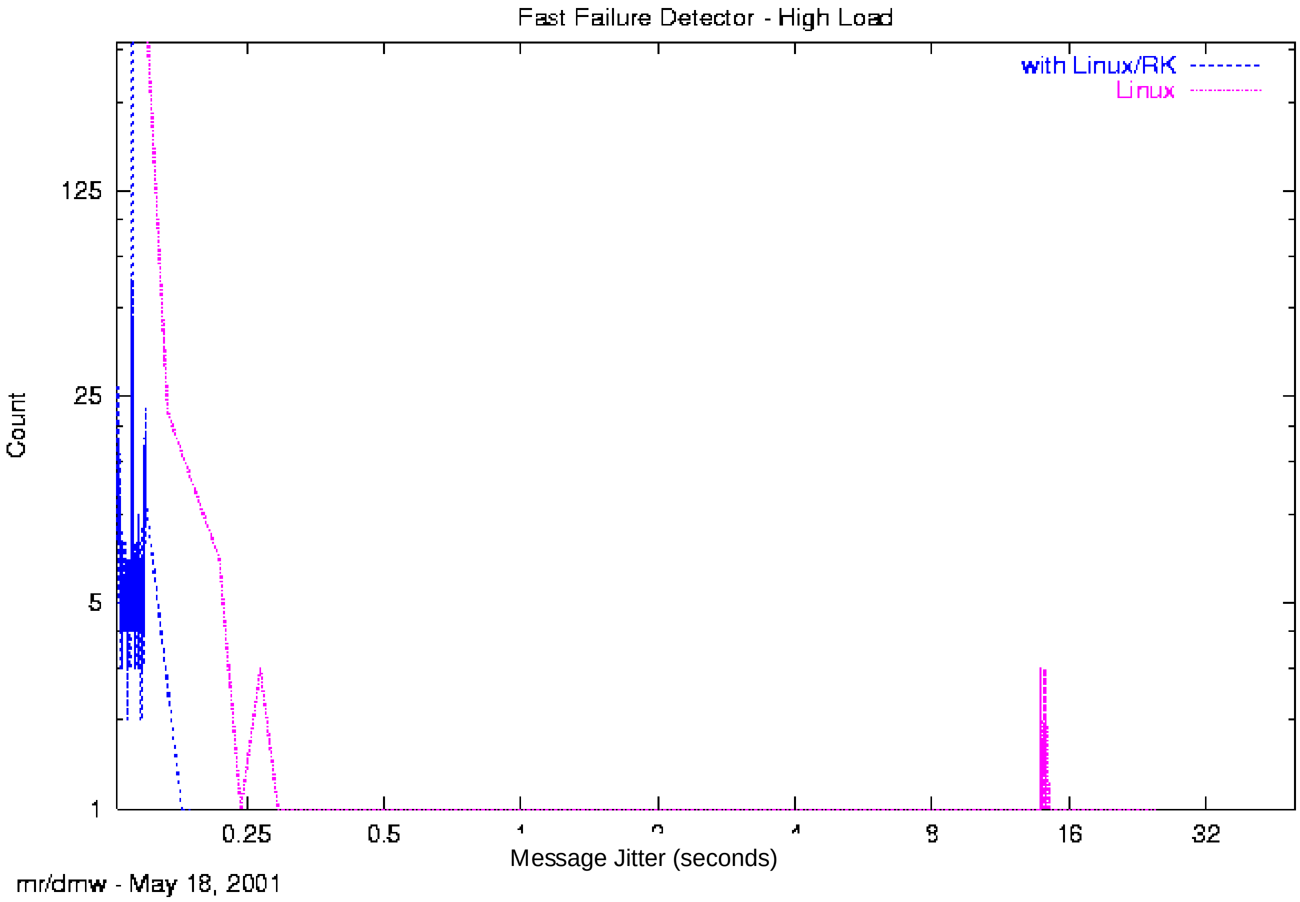
FFD Design Considerations (i)

- FFD (and Ensemble) utilize heartbeat (watch-dog/dead-man timer) pattern
  - Generation and monitoring of heartbeat messages (via time-outs) is a common method of detecting node crash failures
  - Reducing timeouts on missing heartbeat messages allows faster identification of failed nodes and thus supports shorter deadlines
  - Heavy loads cause queuing delays (jitter), which cause heartbeat messages to be tardy, which cause time-outs, which cause nodes to be erroneously declared down, which cause expensive, unnecessary reconfigurations

FFD Design Considerations (ii)

- Assertions on Host Failure Detection
  - Providing dedicated resources for heartbeat generation and monitoring functions can reduce jitter, thus allowing use of shorter timeouts, thus improving real-time properties
  - Dedicated resources can best be provided in a separate host failure detector component that has been specifically designed to support real-time properties

FFD Message Latency (Jitter) Characterization



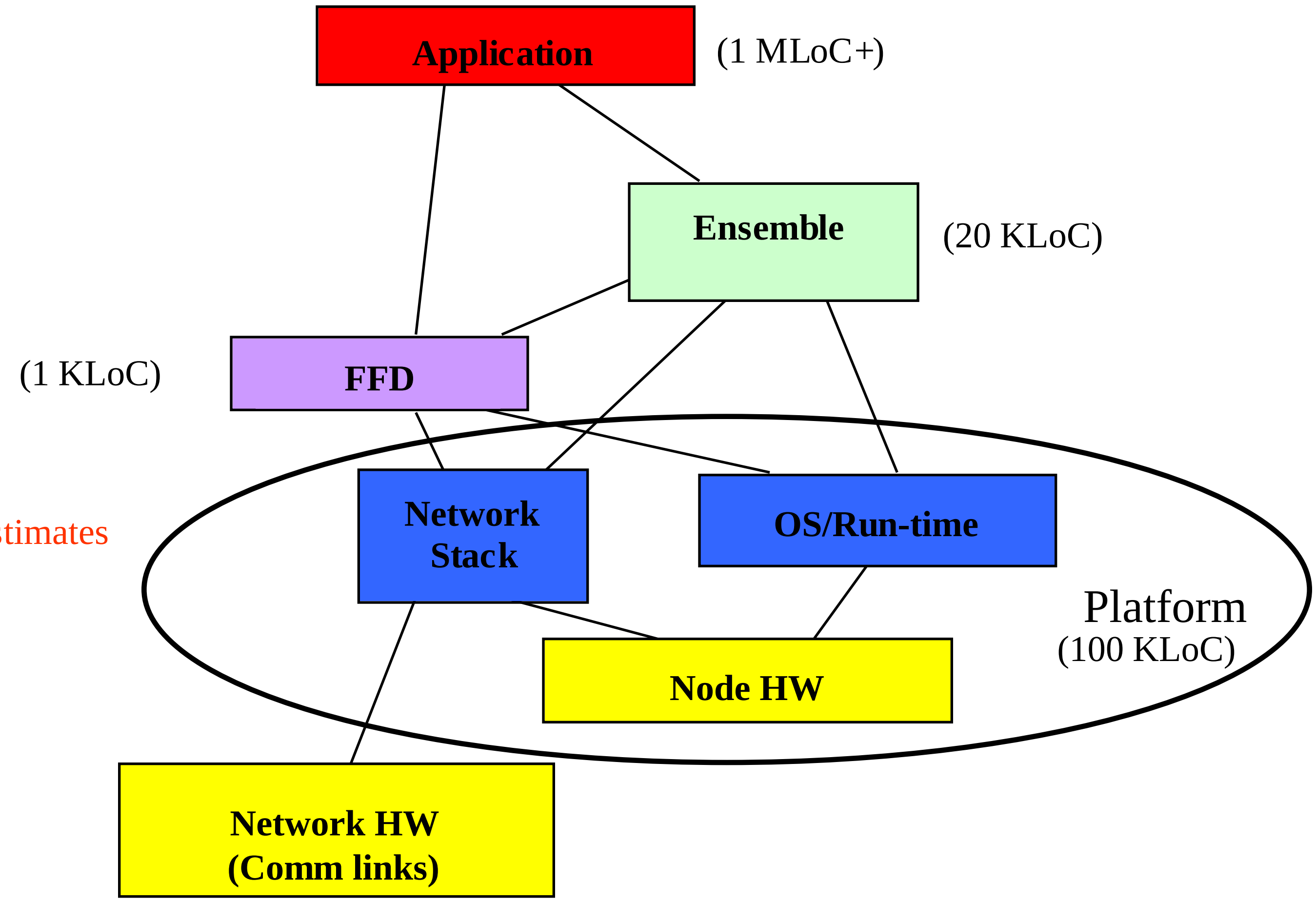
Note on Resource Consumption

- Test-bed: 5 nodes, 10 Mbps Ethernet® LAN
- FFD parameters
  - Time-out period: 0.5 second
  - Replication factor: 5 (i.e., 100 msec heartbeat)
- FFD uses <1% of 100 Hz, 32 MB PC
  - Note: value is imprecise due to use of pseudo-Monte Carlo measuring technique in UNIX® and Linux®
- FFD uses <5% of network bandwidth
  - Note: value is minimum value reported on hub

Some Simplifying Assumptions for First-Order Fault Analysis

- A component failure is due to either internal fault, environmental fault, or failure in other ("depends upon") component
- Internal component failure rate is proportional to number of errors (bugs) in it
- HW component bug count is proportional to transistor count
- SW component bug count is proportional to lines of code (LoC)

(Simplified) Fault Dependency Graph of Node Failure Detection Function



Caution: LoC estimates are notional!

THE Open GROUP

First-Order Fault Analysis

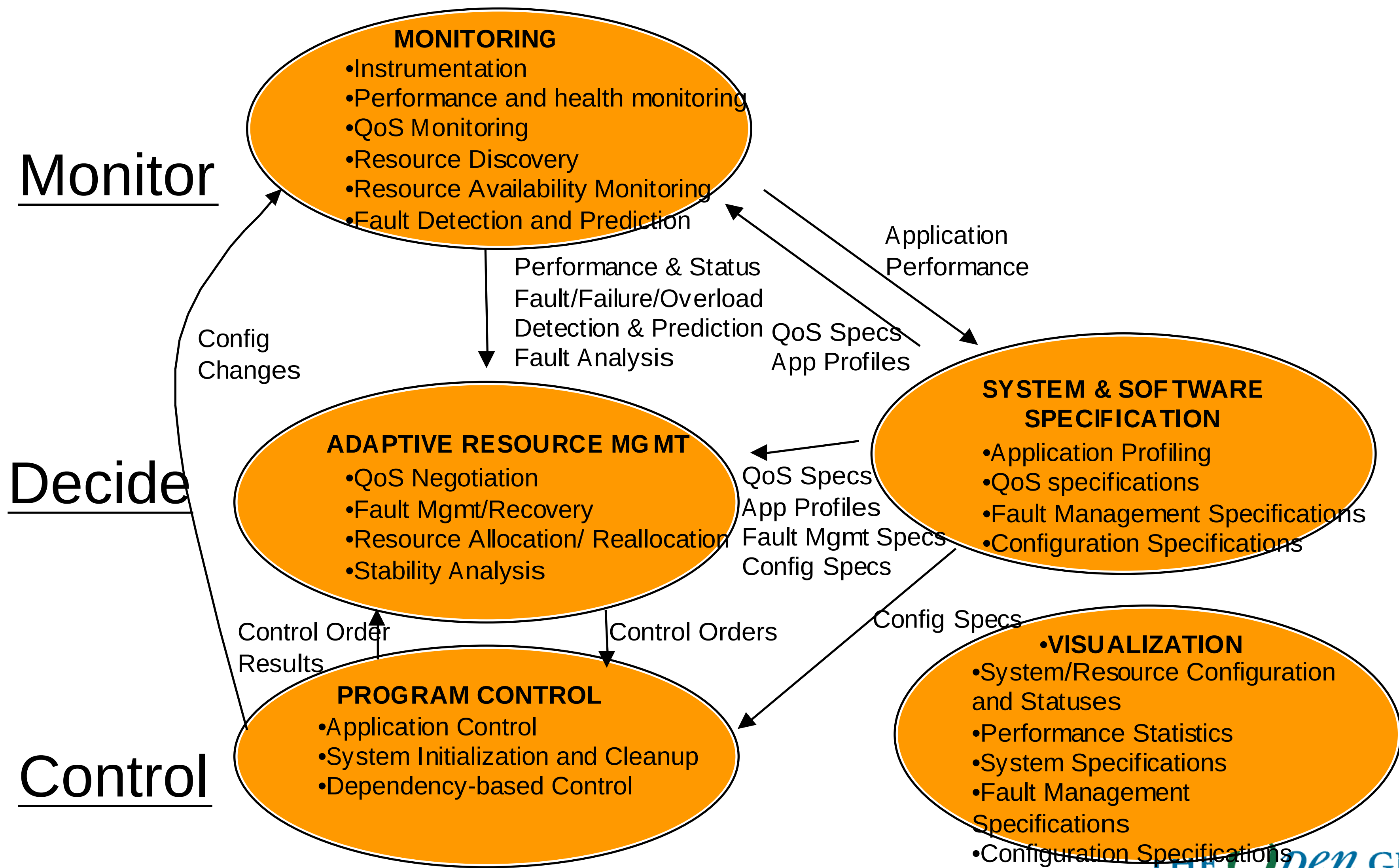
- Examine projected failure rates of fielded components based on bug rates (br)
- Example failure rates (fr) w/o FFD
  - fr(Ensemble) ≈ br(20K) + fr(platform) + fr(net)
  - fr(application) ≈ br(1M) + fr(Ensemble)
- Example failure rates w/ FFD
  - fr(FFD) ≈ br(2K) + fr(platform)+fr(net HW)
  - fr(Ensemble´) ≈ fr(Ensemble) + fr(FFD)
- Therefore
  - FFD should be more reliable than Ensemble or application

Failure Detection Types and Failure Correlation

- True negative: normal operation
- True positive: correctly detected failure
- False positive: erroneously asserted failure
  - Will wastefully perform system reconfiguration
- False negative: overlooked a failure condition
  - Unable to mask failure
  - May lead to overall system failure
- False positives can be tolerated as long as there aren't "too many" of them
- False negatives can potentially lead directly to system failure

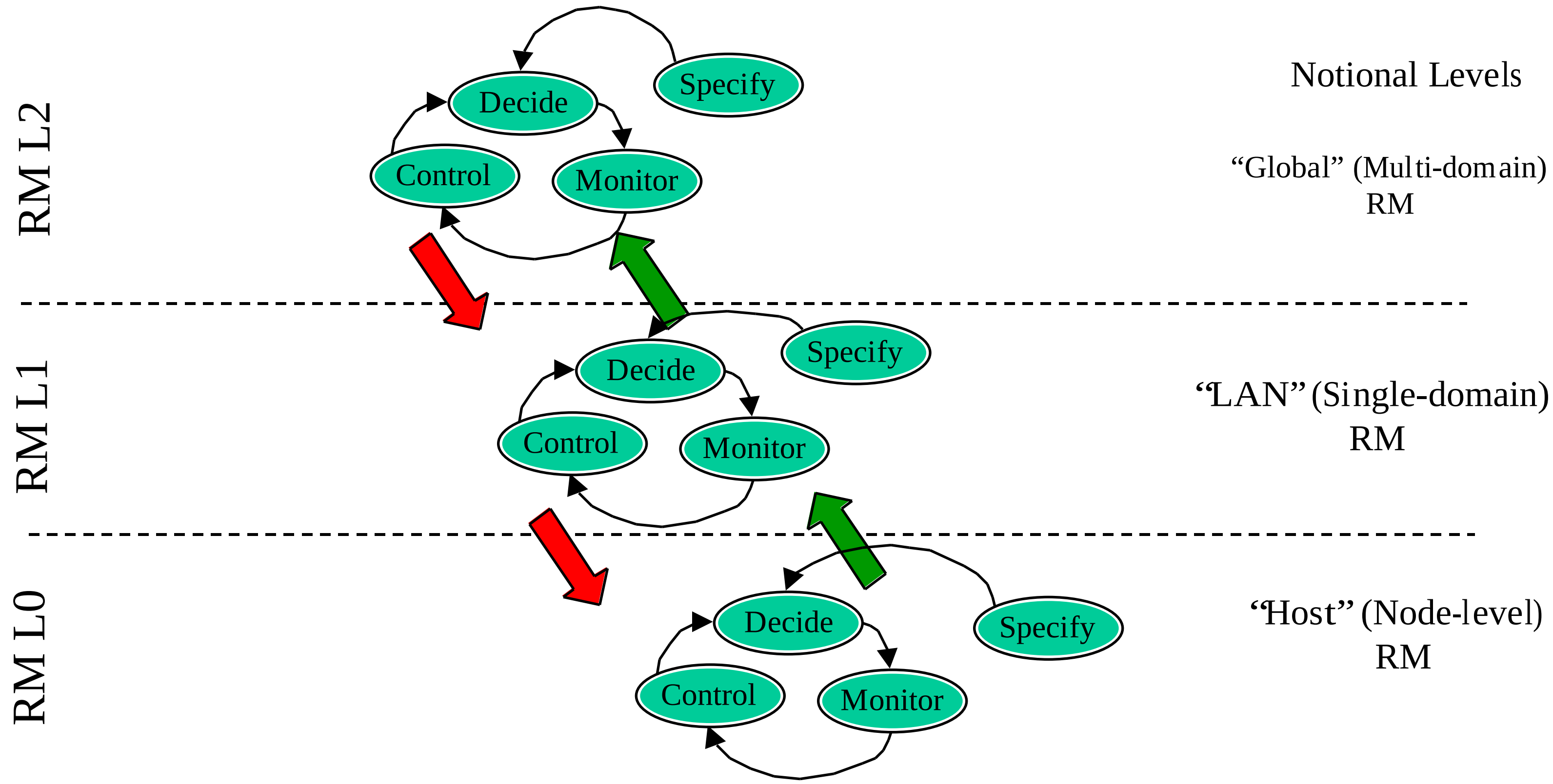
Layered Resource Management

Basic Resource Management Functions

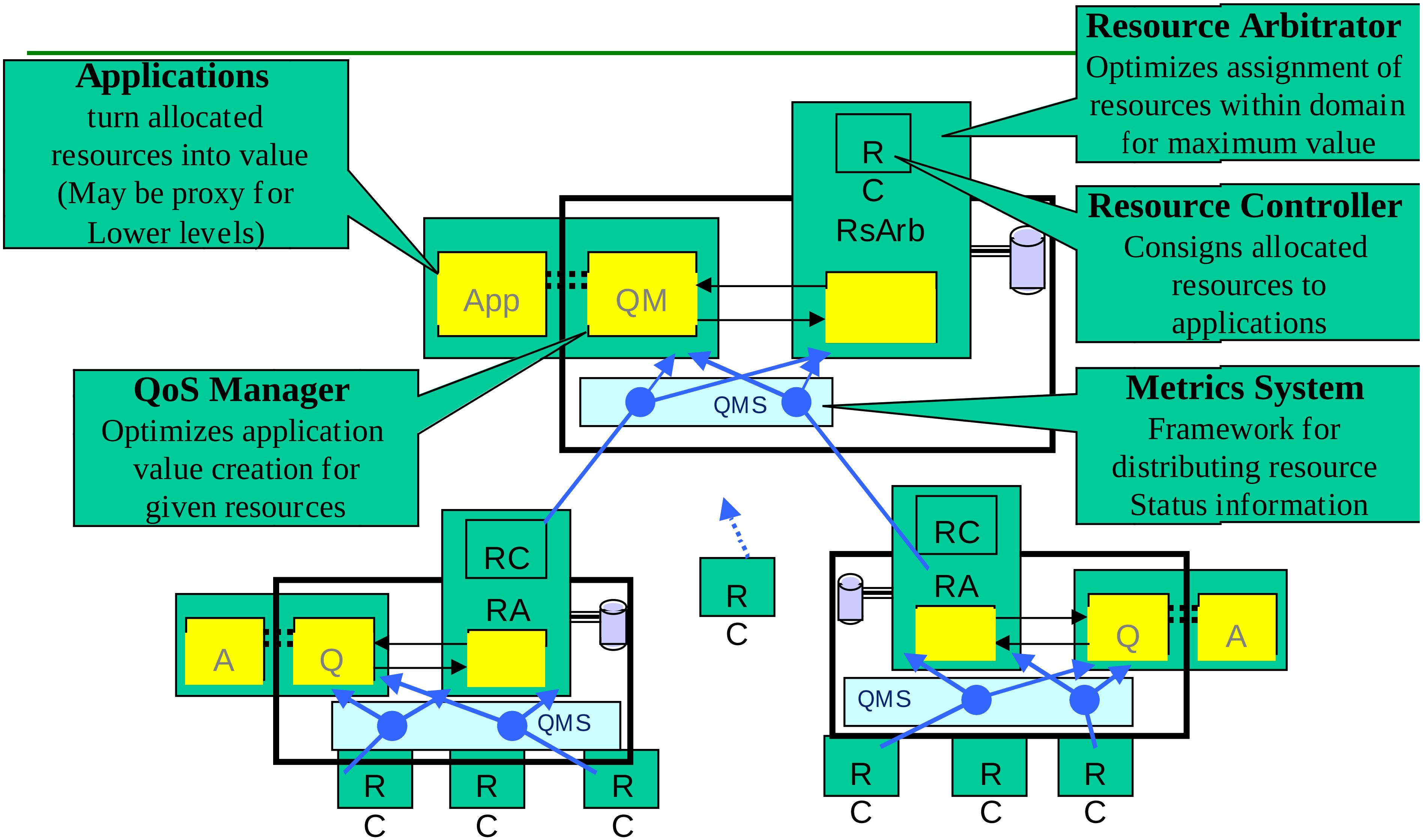


Resource Management Levels

- Accepts directives from higher levels
- Provides status to higher levels
- Manages lower levels
- Receives information about performance of lower levels

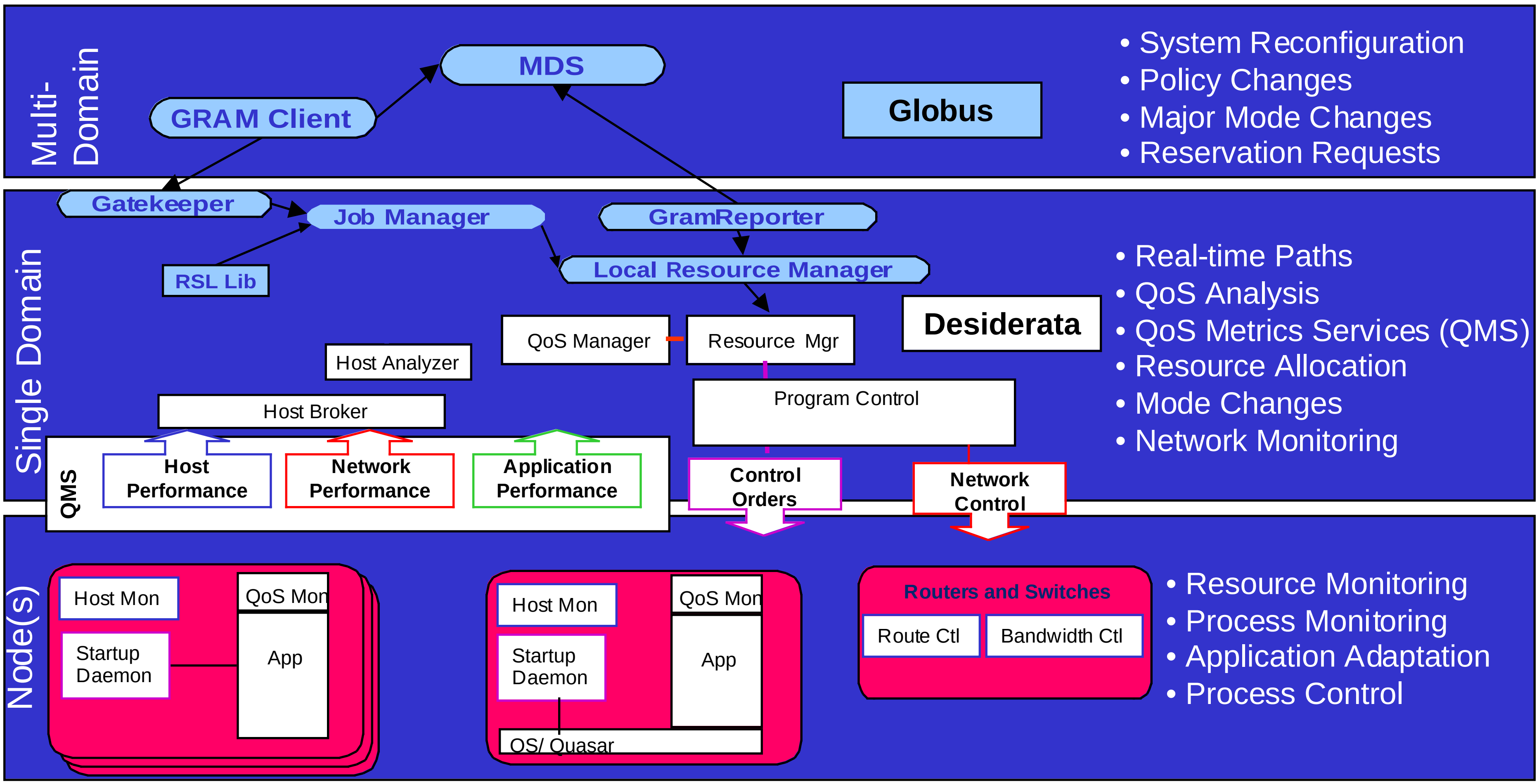


Functional LRM Architecture

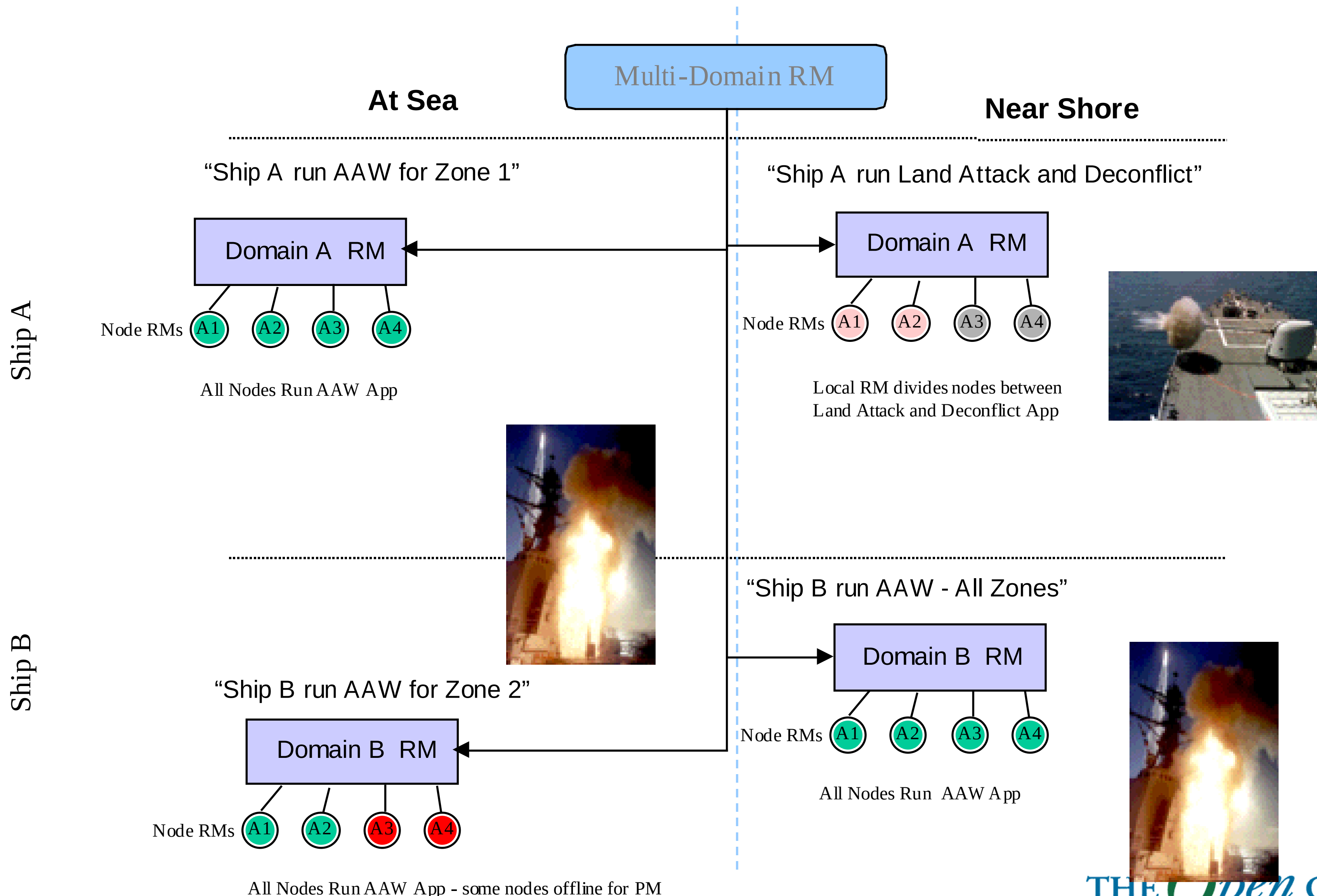


THE Open GROUP

Multilevel Resource Management Implementation



Example of Gobal Adaption of Resource Allocations to Mission Assignments



Background and Motivation for Layered Resource Management (LRM)

- Integration: Node, Domain & Multi-Domain Resource Managers have different areas of competence
- Operational Scope: Different approaches needed for different scales of operation
  - Unlikely to find one algorithm, instrumentation, or control approach that scales to fit all sizes
- Fit with human and mission needs

THE Open GROUP

LRM Architectural Precepts

- Provide Monitor-Policy/Decision-Control loop at multiple scopes (granularities of action & control)
- LRM based on principle of delegation
  - "Opaque" assignments
  - Assume competence of lower layer
- Do what a human manager would have done, only faster/automated
- Practical layering splits will determined by response time, determinacy of environment
- May be multiple lower-layer managers under each higher layer

Adaptive Applications and Application Paths

Application & QoS Models (DeSiDeRaTa)

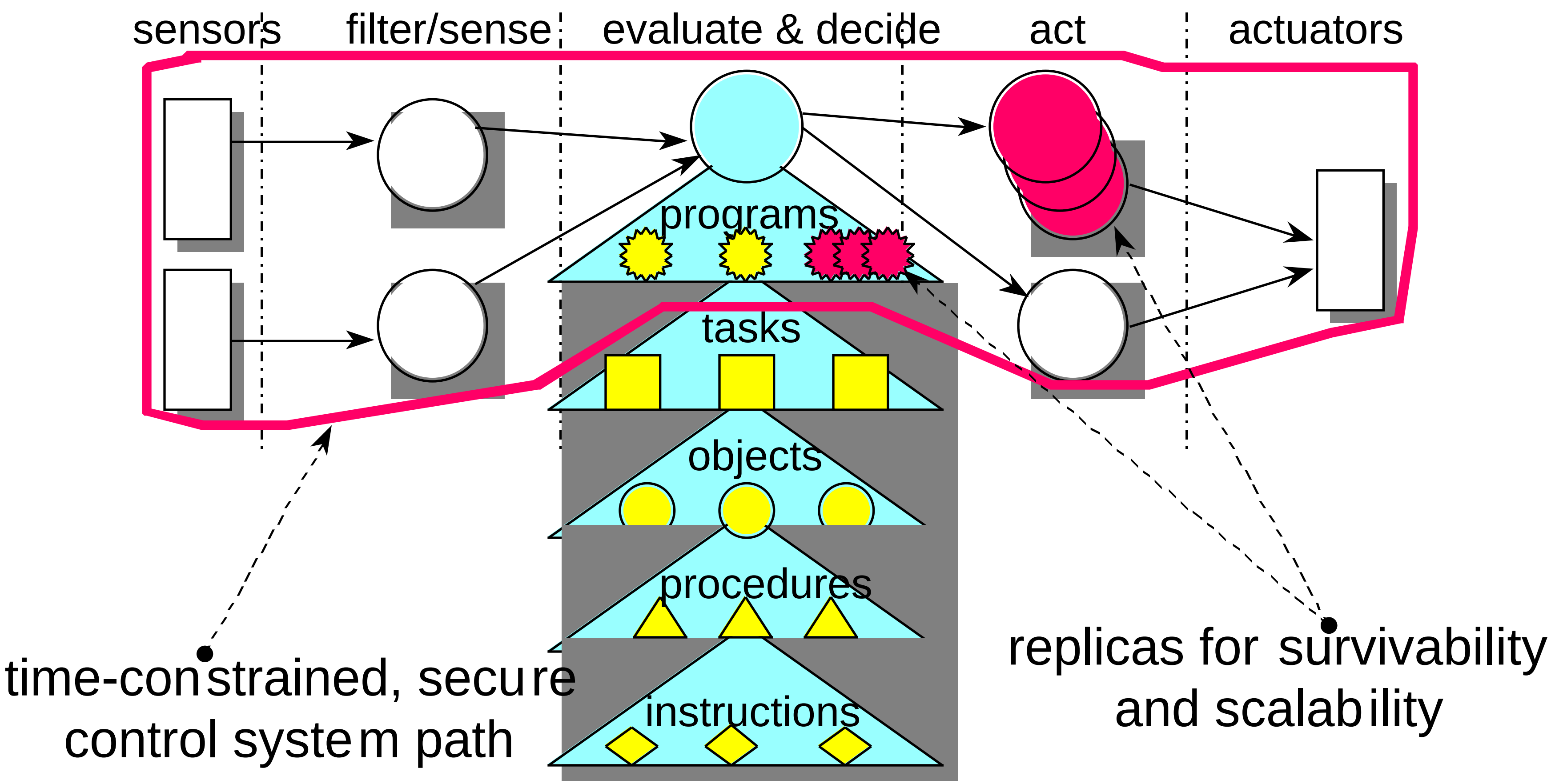
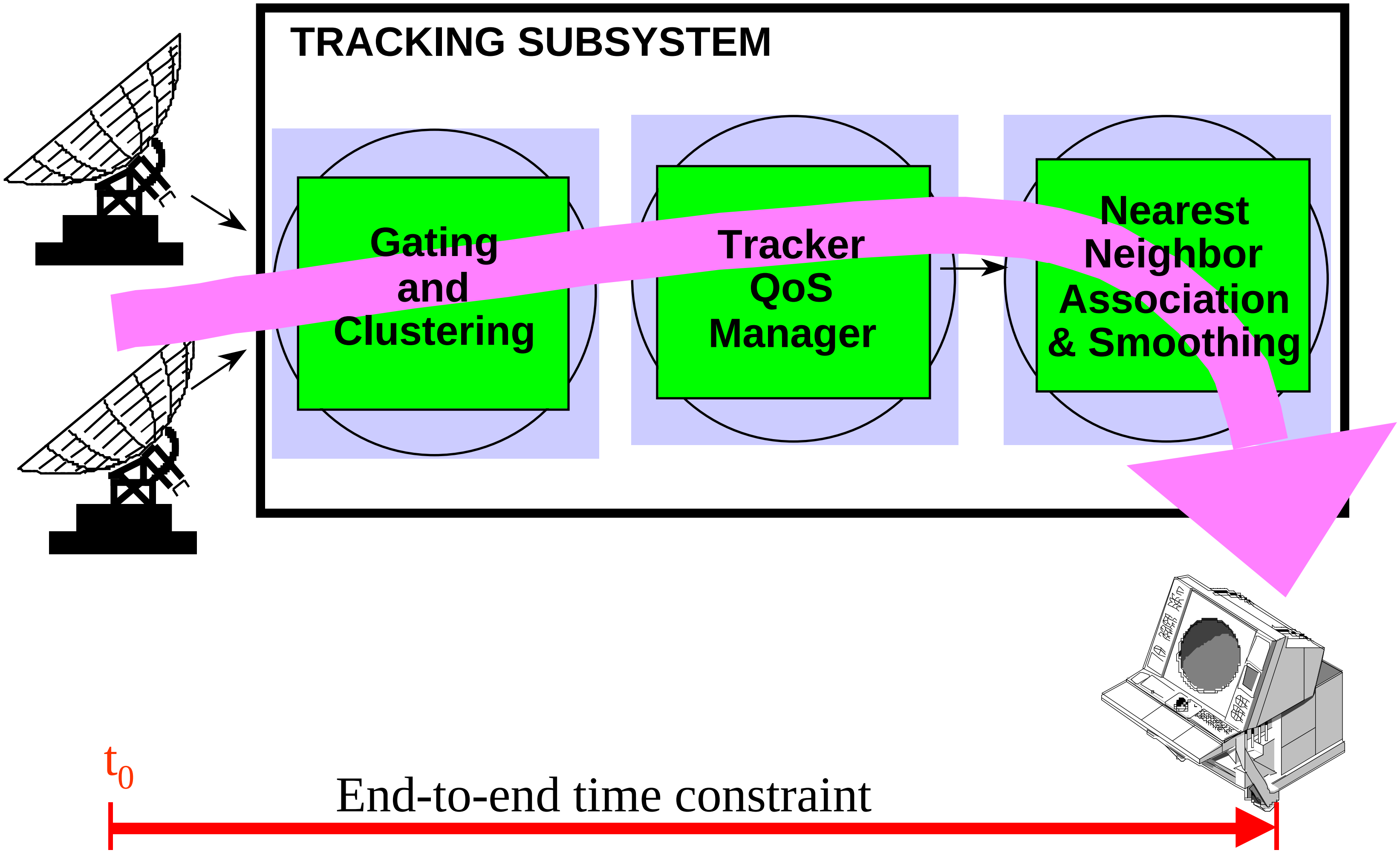
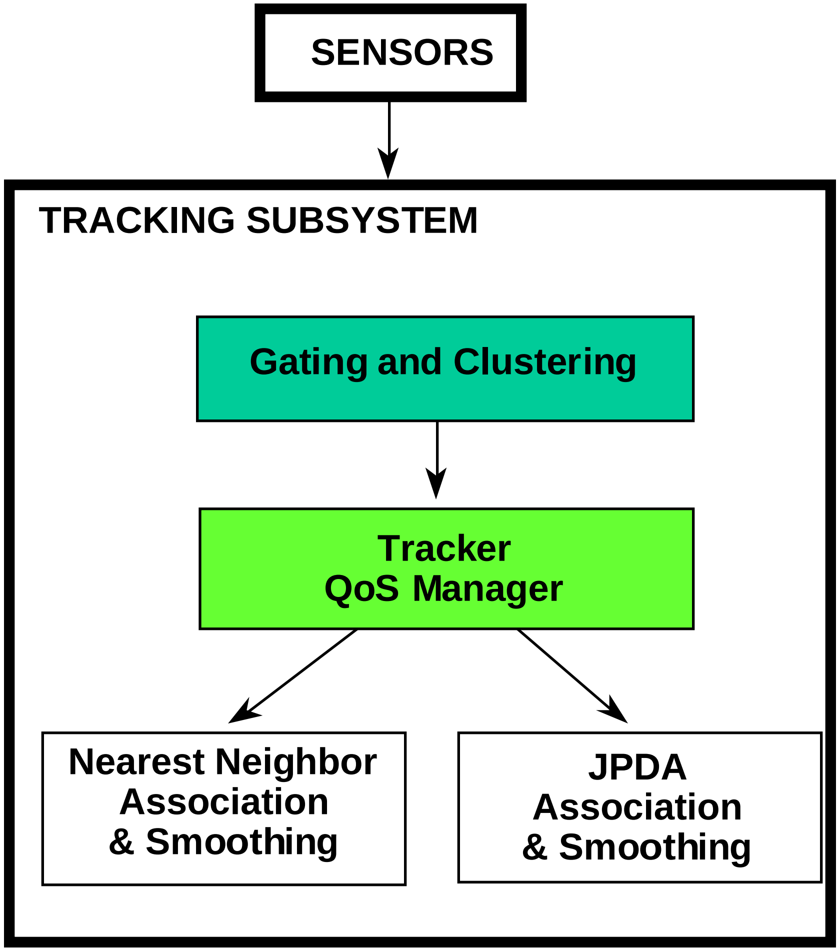


Diagram courtesy of Lonnie Welch, Ohio U.

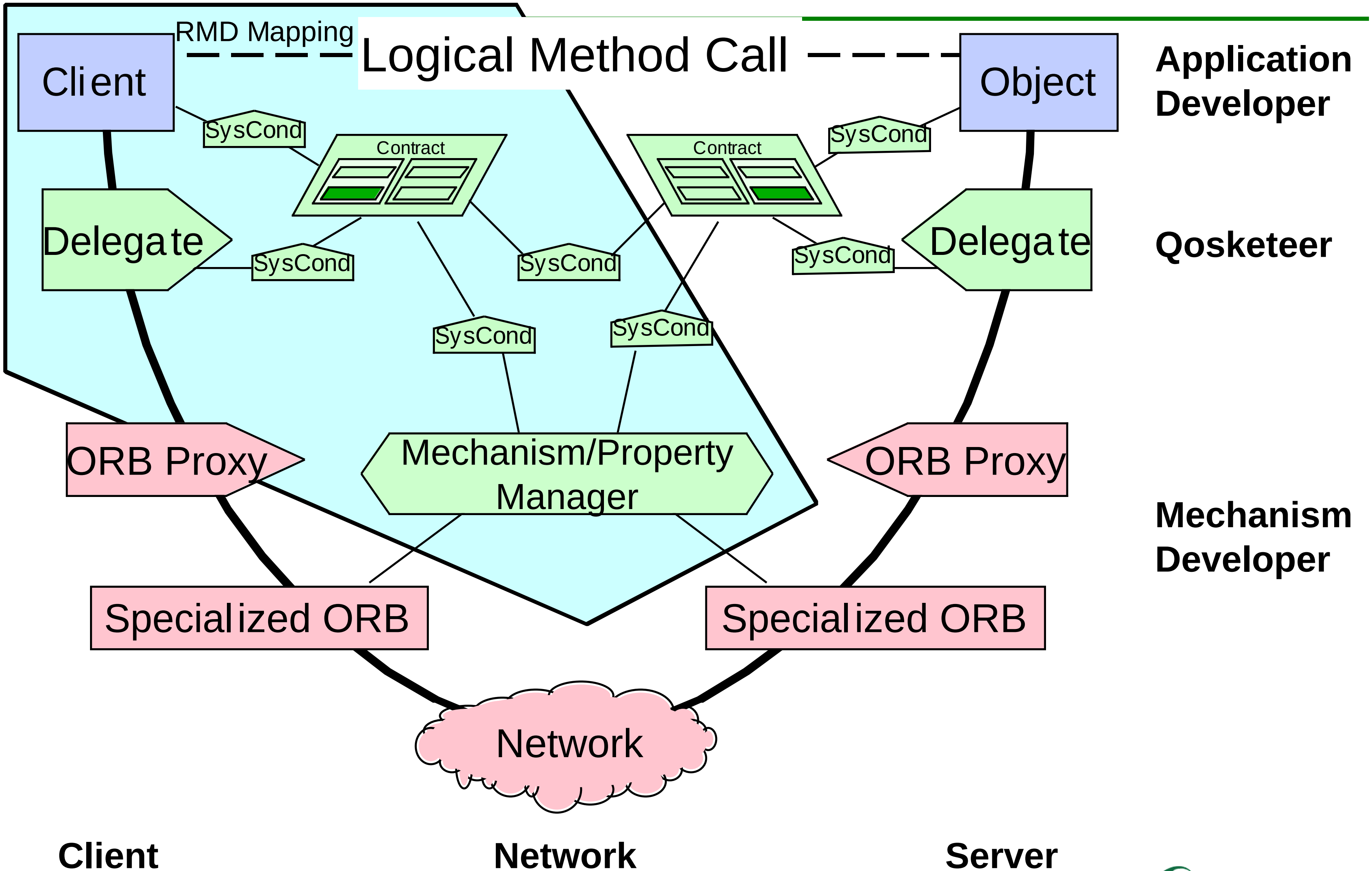
QoS-Driven Adaptive Tracking



QoS-Driven Adaptive Tracking with Enhanced Infrastructure

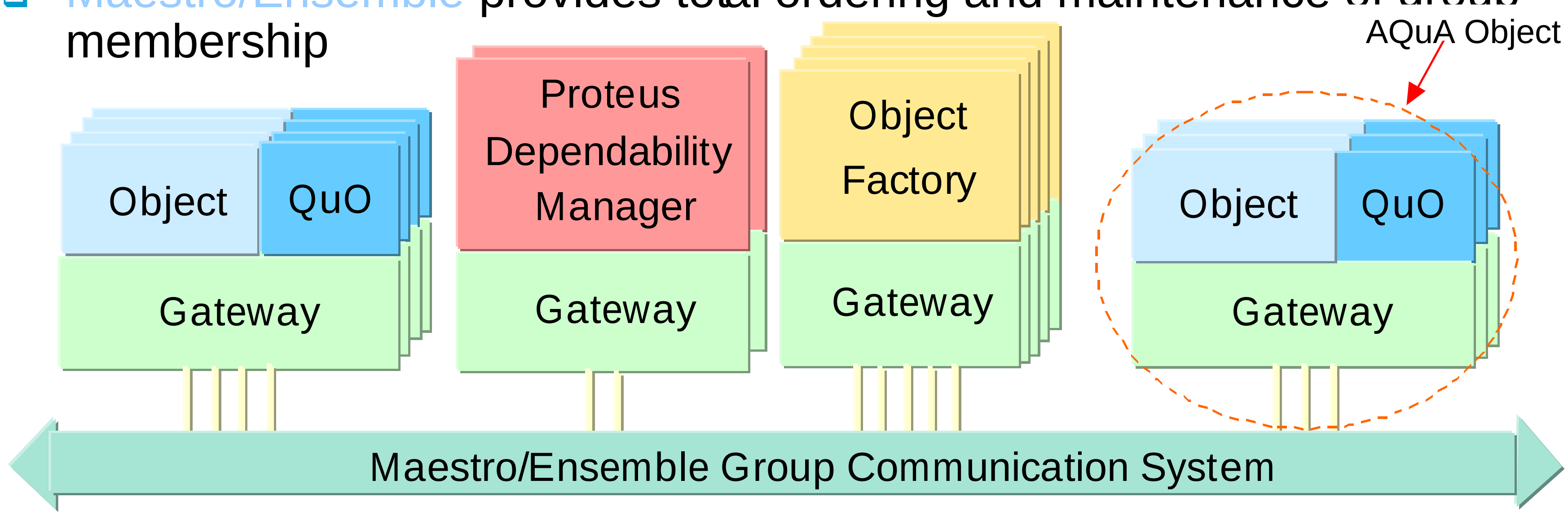


QuO


RMD Mapping
Logical Method Call
Client
Object
Application Developer
SysCond
Contract
Delegate
Qosketeer
SysCond
SysCond
SysCond
SysCond
SysCond
SysCond
SysCond
Delegate
ORB Proxy
Mechanism/Property Manager
ORB Proxy
Mechanism Developer
Specialized ORB
Specialized ORB
Network
Client
Network
Server

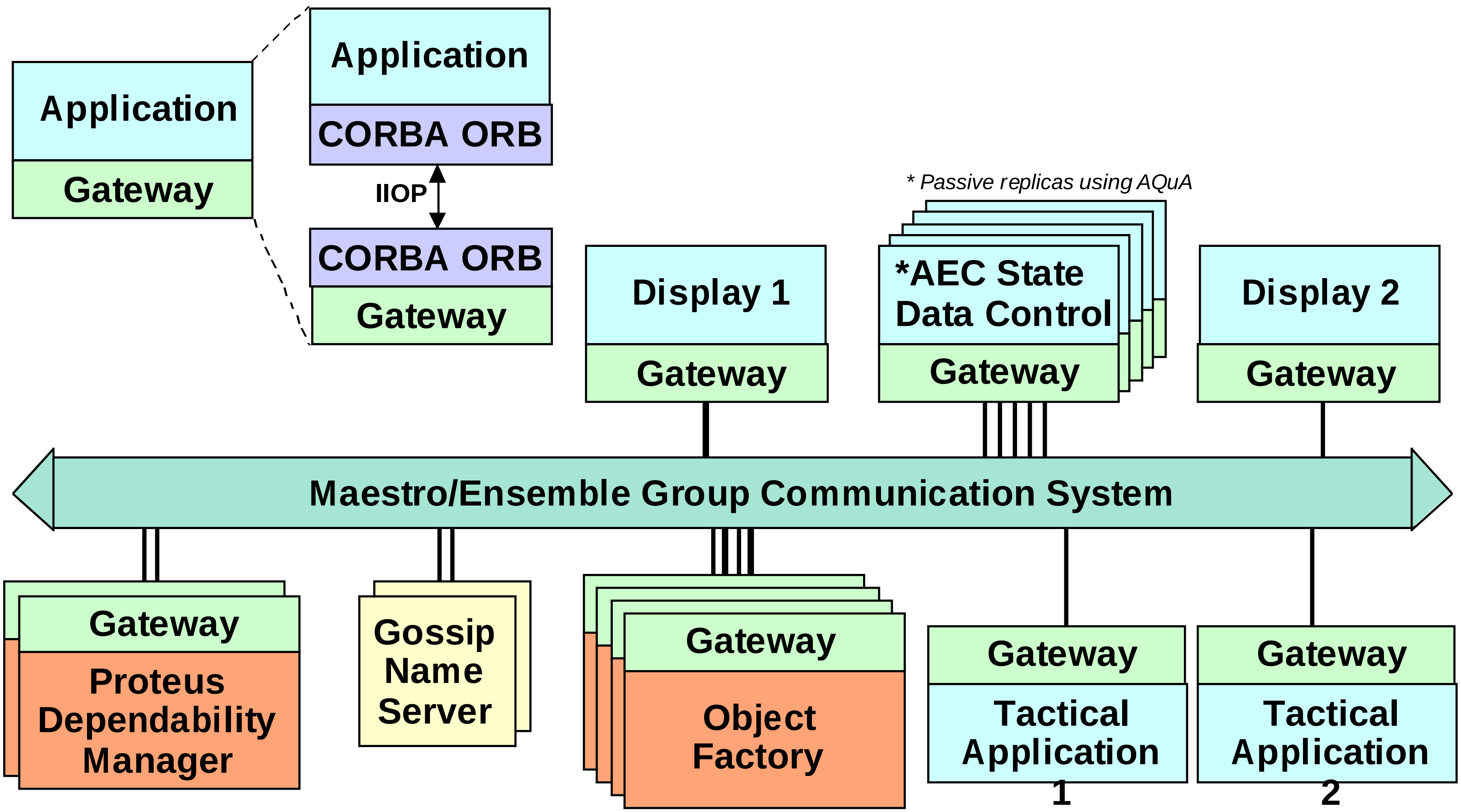
AQuA Architecture

- Quality Objects (QuO) specify the level of dependability for application objects
- Proteus manages fault tolerance depending on application dependability requirements and faults that occur in the system
- Gateways:
  - Translate CORBA remote method calls into group communication messages
  - Implement multiple replication and communication mechanisms
- Maestro/Ensemble provides total ordering and maintenance of group membership



THE Open GROUP

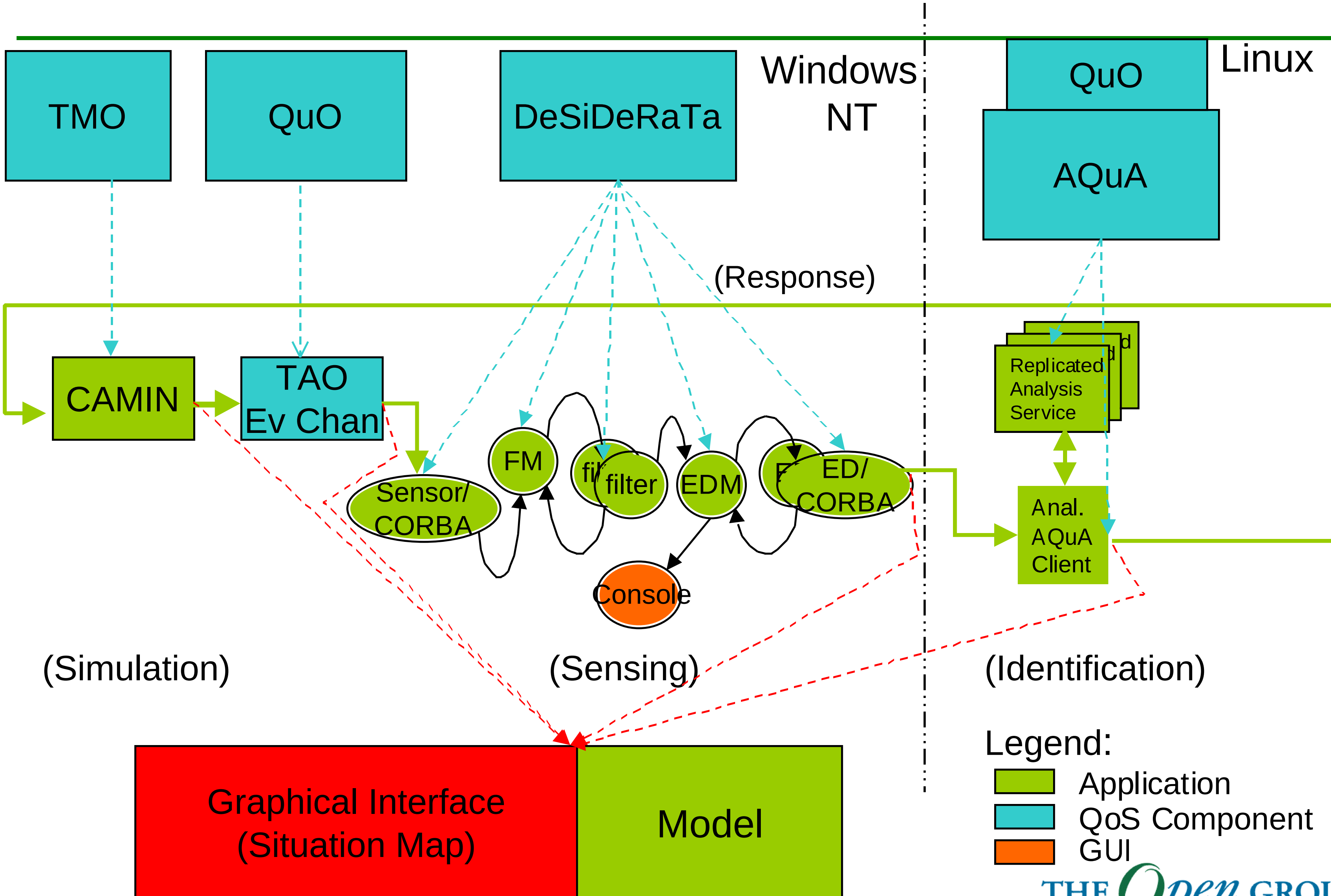
DSS/AQuA Architecture Overview



THE Open GROUP

Technology Demonstration & Transfer

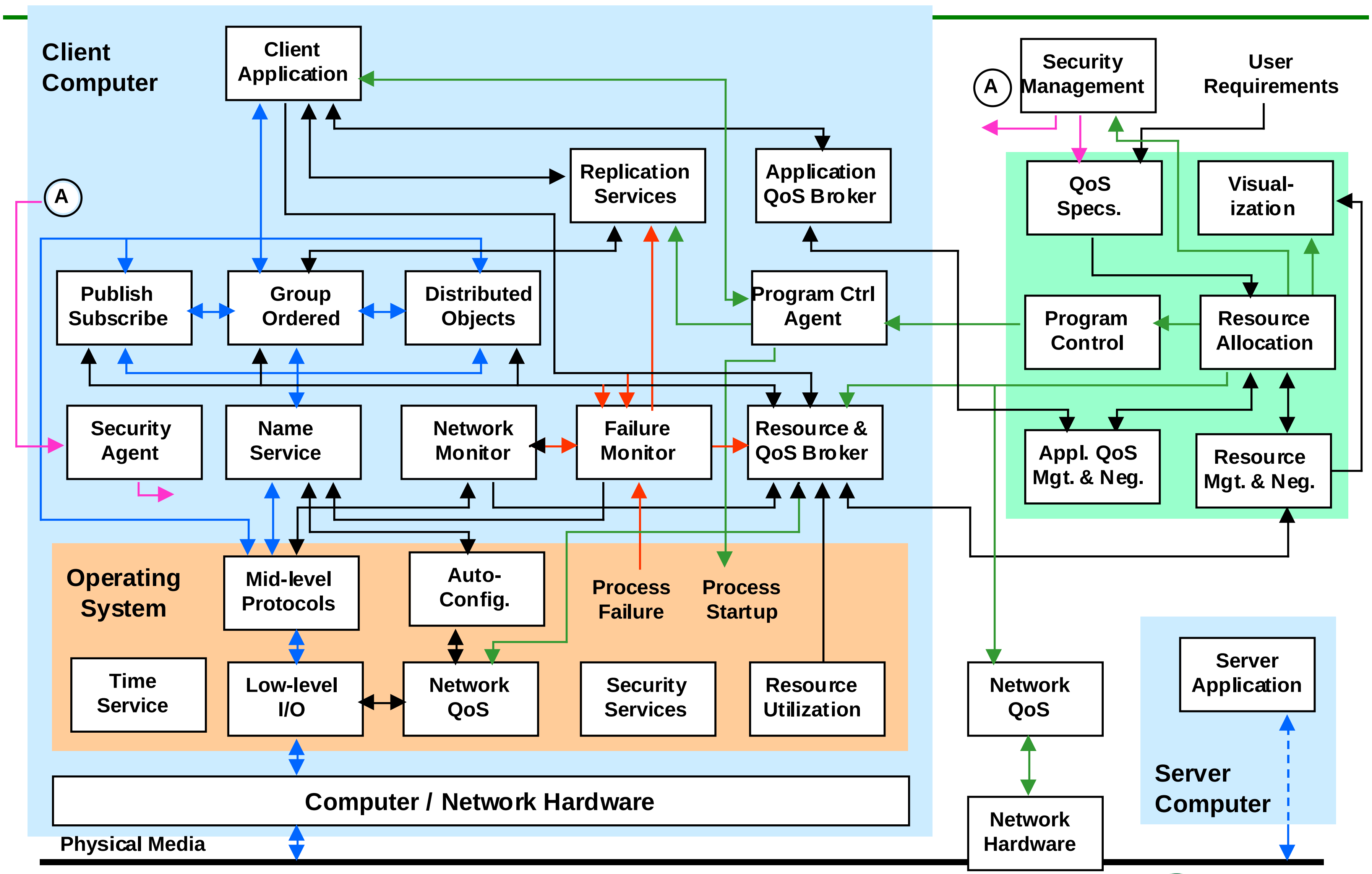
Demo Architecture



THE Open GROUP

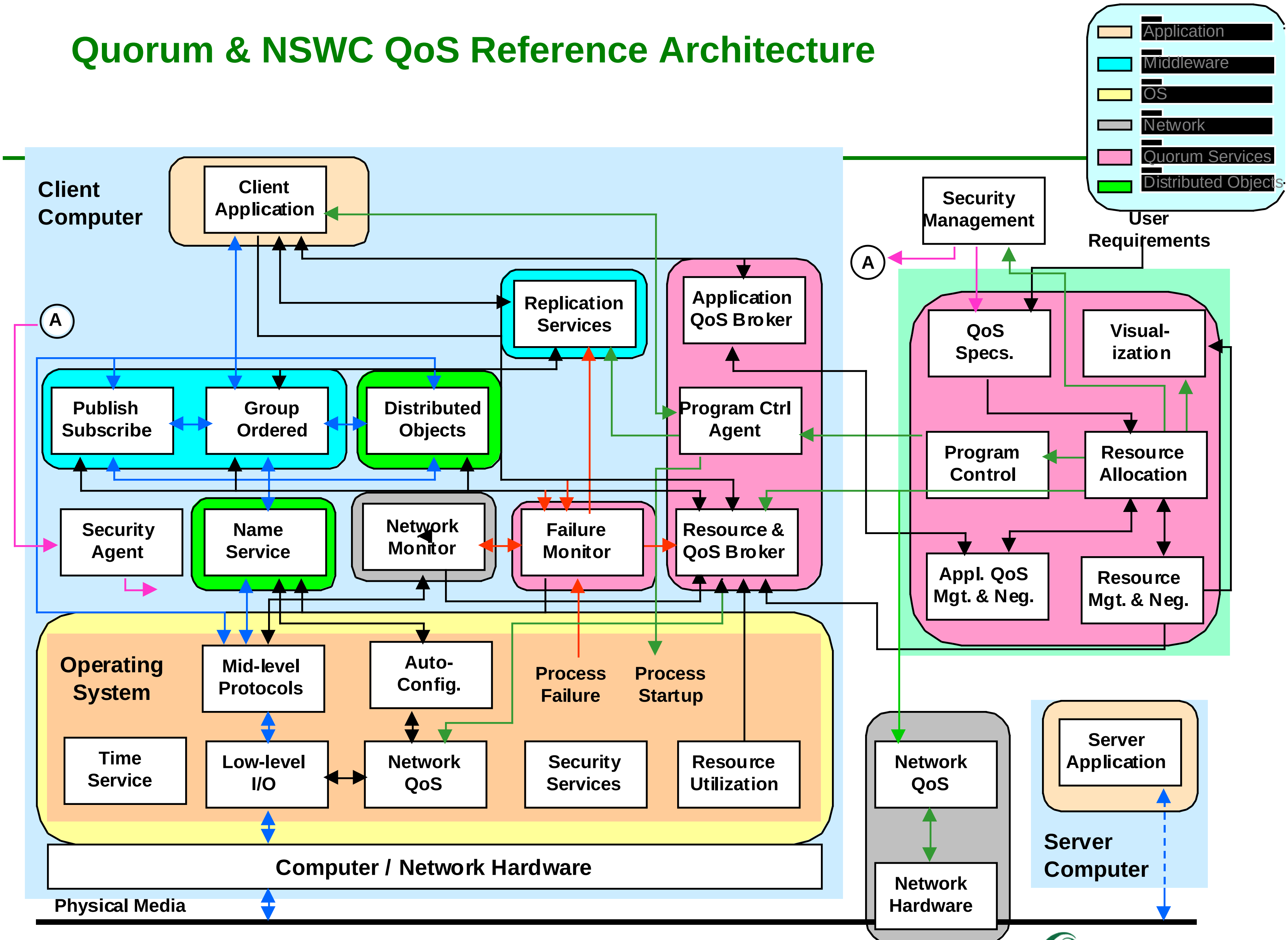
DMW/DML 010905

NSWC QoS REFERENCE ARCHITECTURE

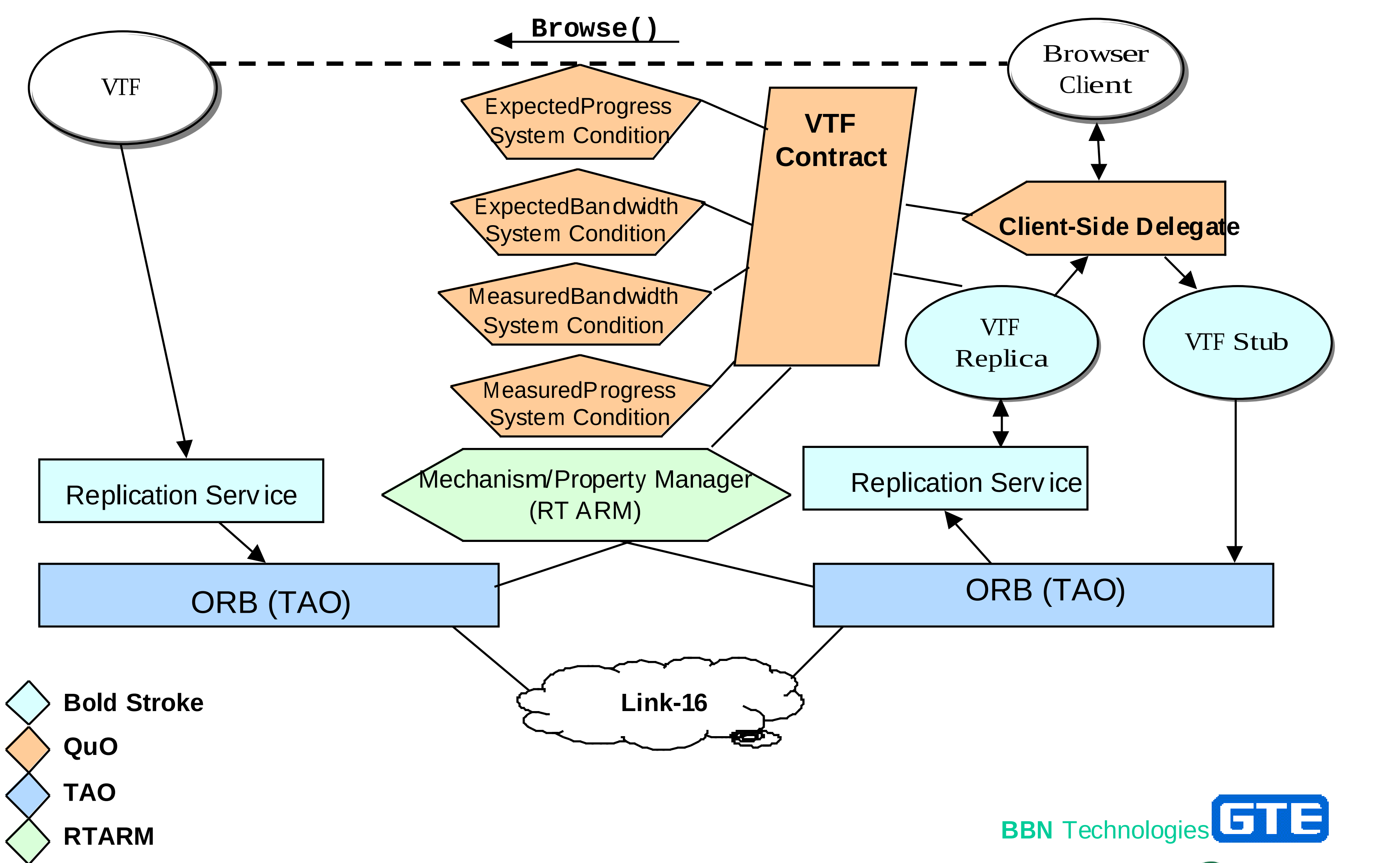


THE Open GROUP

Quorum & NSWC QoS Reference Architecture



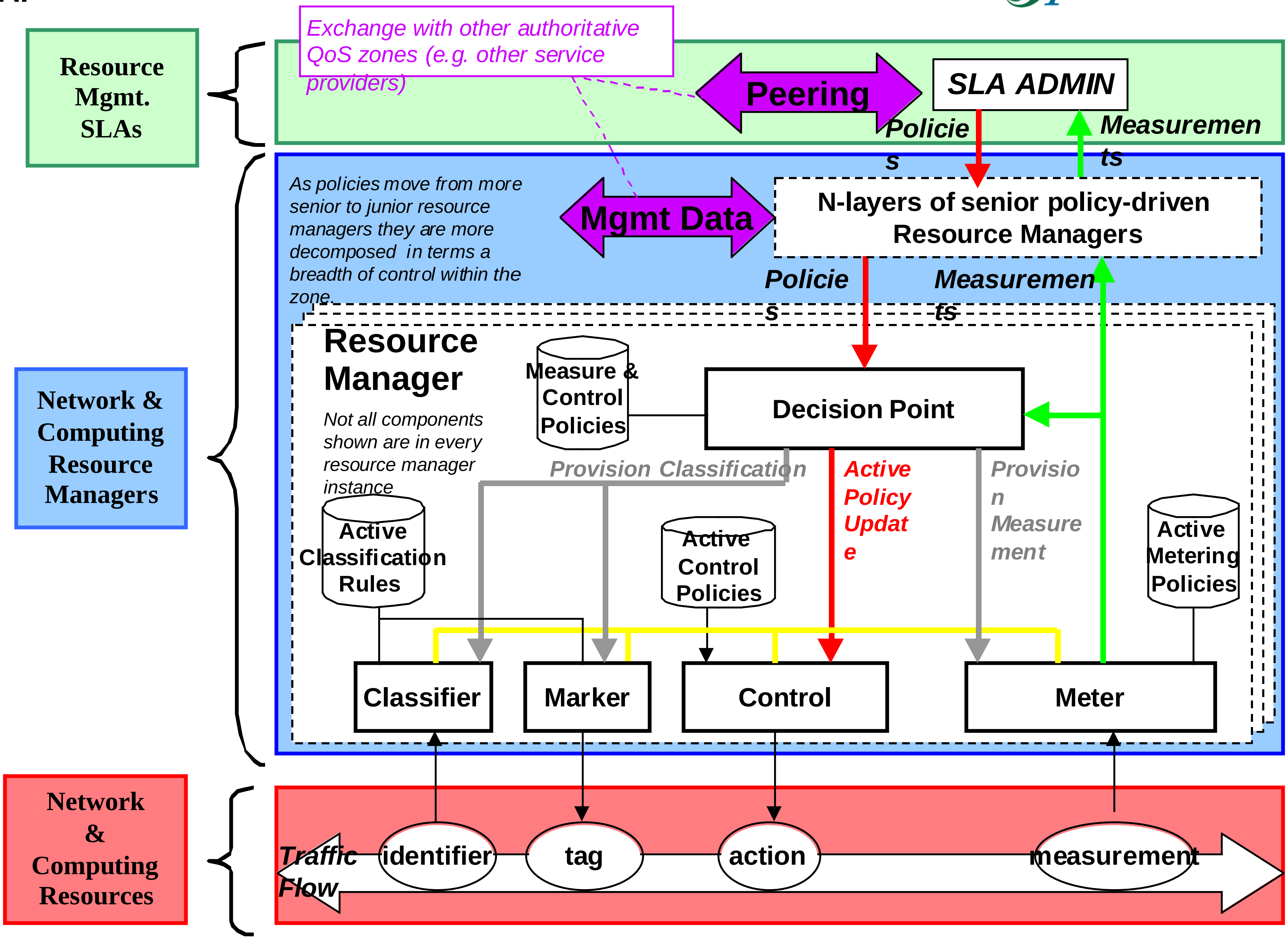
WSOA Middleware Framework



BBN Technologies GTE

THE Open GROUP

QoS Task Force - MID-LEVEL COMPONENT MAP



Certification Services

Certification

- Certification of a product provides formal recognition of conformance to an open standard or specification.
  - Suppliers are able to make and substantiate clear claims of conformance to a standard
  - For suppliers, it is a way to demonstrate that they stand behind their products.
  - Buyers are able to specify and successfully procure conforming products that interoperate
  - Buyers get a vendor warranty of conformance to standards when a product is certified.

Open Group Certification

- The Open Group has developed and operate today provide buyers of certified products a guarantee that:
  - The product conforms to an open standard or specification
  - The product will remain conformant, through modifications, enhancement, fixes and upgrades
  - If there ever is a non-conformance, it will be fixed in a timely manner

- Supported by development and adoption of conformance test suites

Advanced Research

For more information:
http://www.opengroup.org/ar/
Mail: research@opengroup.org