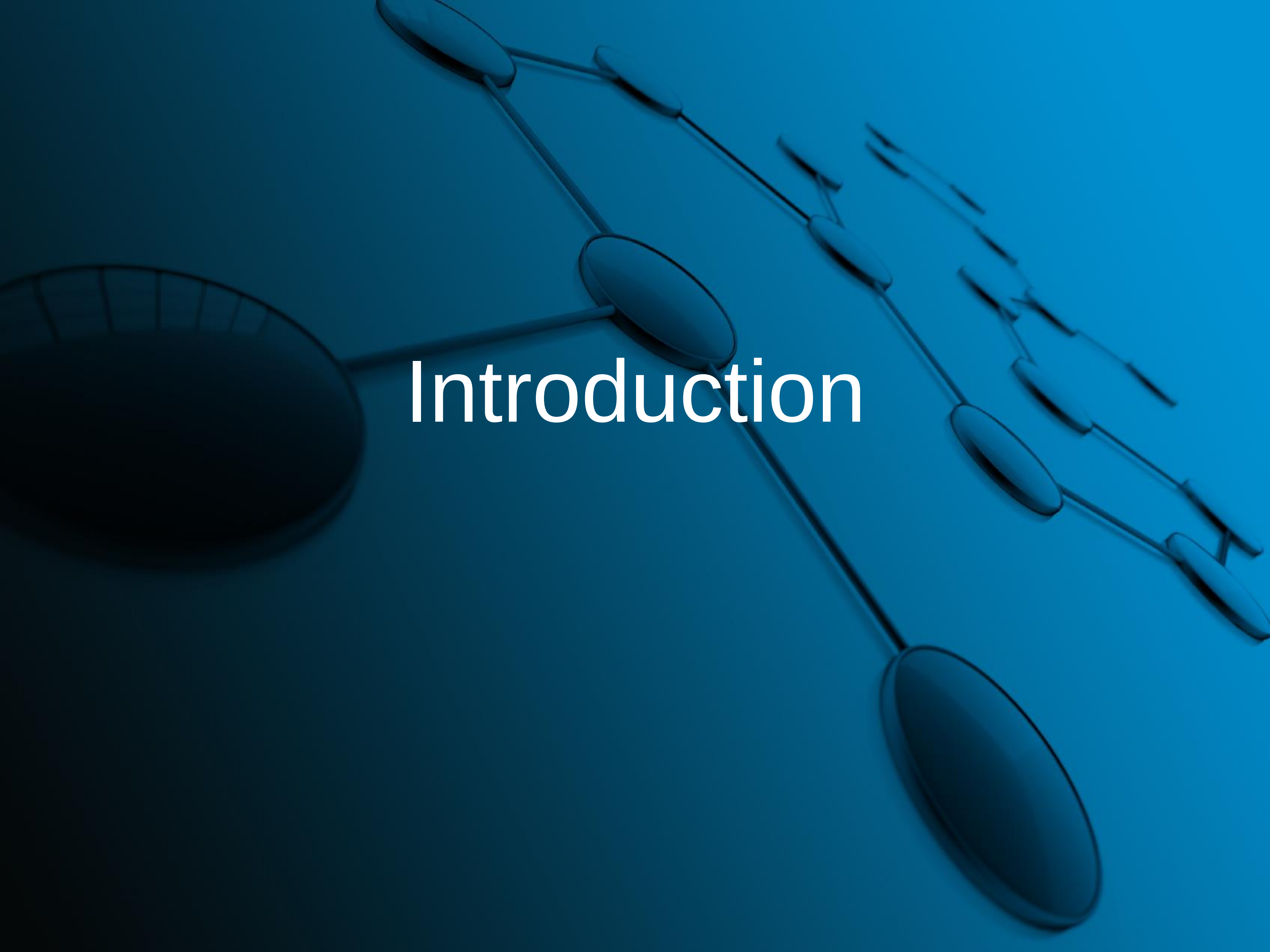
NASTEL®

Driving Business Transaction Performance®

Understanding Queue Manager Performance

MQ Technical Conference V2.0.1.3



Introduction

Introduction

Richard Nikula

- VP of Product Development and Support

Involved in “MQ” since early 90’s

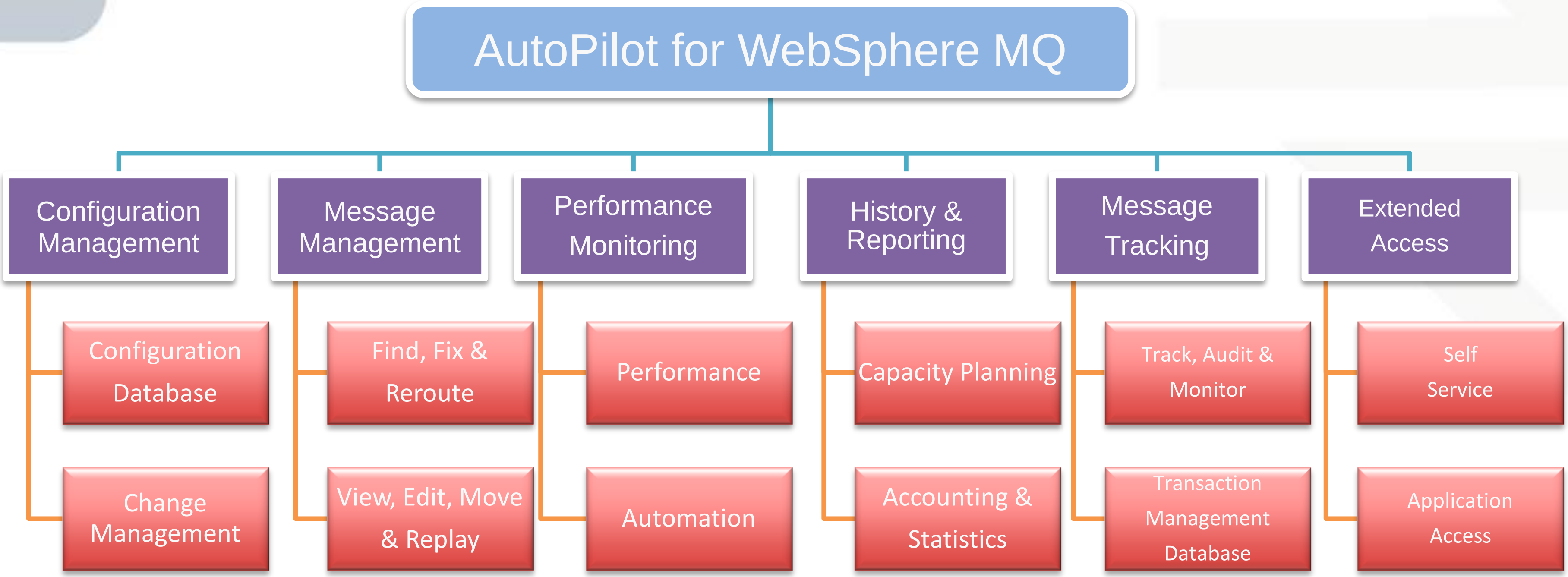
- Primarily at the technology layer

About Nastel Technologies

- Founded in 1994
- Middleware-centric Application Performance Management software supplier
- Core competency : Messaging Middleware, Java Application Servers, ESB's and other SOA technologies

NASTEL®
Driving Business Transaction Performance®

Nastel APWMQ Capabilities



AutoPilot for WebSphere MQ

- Configuration Management
  - Configuration Database
  - Change Management
- Message Management
  - Find, Fix & Reroute
  - View, Edit, Move & Replay
- Performance Monitoring
  - Performance
  - Automation
- History & Reporting
  - Capacity Planning
  - Accounting & Statistics
- Message Tracking
  - Track, Audit & Monitor
  - Transaction Management Database
- Extended Access
  - Self Service
  - Application Access

Overview

In this session, we will demonstrate how to get insight into the behavior of your queue managers using several techniques to determine.

- Do the queue managers perform worse at one time of day compared to another?
- What impact do different options have on message behavior?
- How do the channels perform between different queue managers?
- Where do message waits happen?

Agenda

- Introduction to Benchmarks
- Concepts required for developing a benchmark
- Synthetic versus real measurements
- Benchmark using a simple Ping
- Benchmark using Synthetic Message Tracking
- Tracking actual Messages
- Conclusion

NASTEL®
Driving Business Transaction Performance®

Benchmarking

Benchmark

In computing, a benchmark is the act of running a computer program, a set of programs, or other operations, in order to assess the relative performance of an object, normally by running a number of standard tests and trials against it.

Source: Wikipedia

NASTEL®
Driving Business Transaction Performance®

Benchmark Basics

Provides a baseline to which changes can be observed to determine the impact to the environment

- Using a consistent workload eliminates application changes and usage differences from influencing the tests
- Compare (not explain) differences in results between different configurations

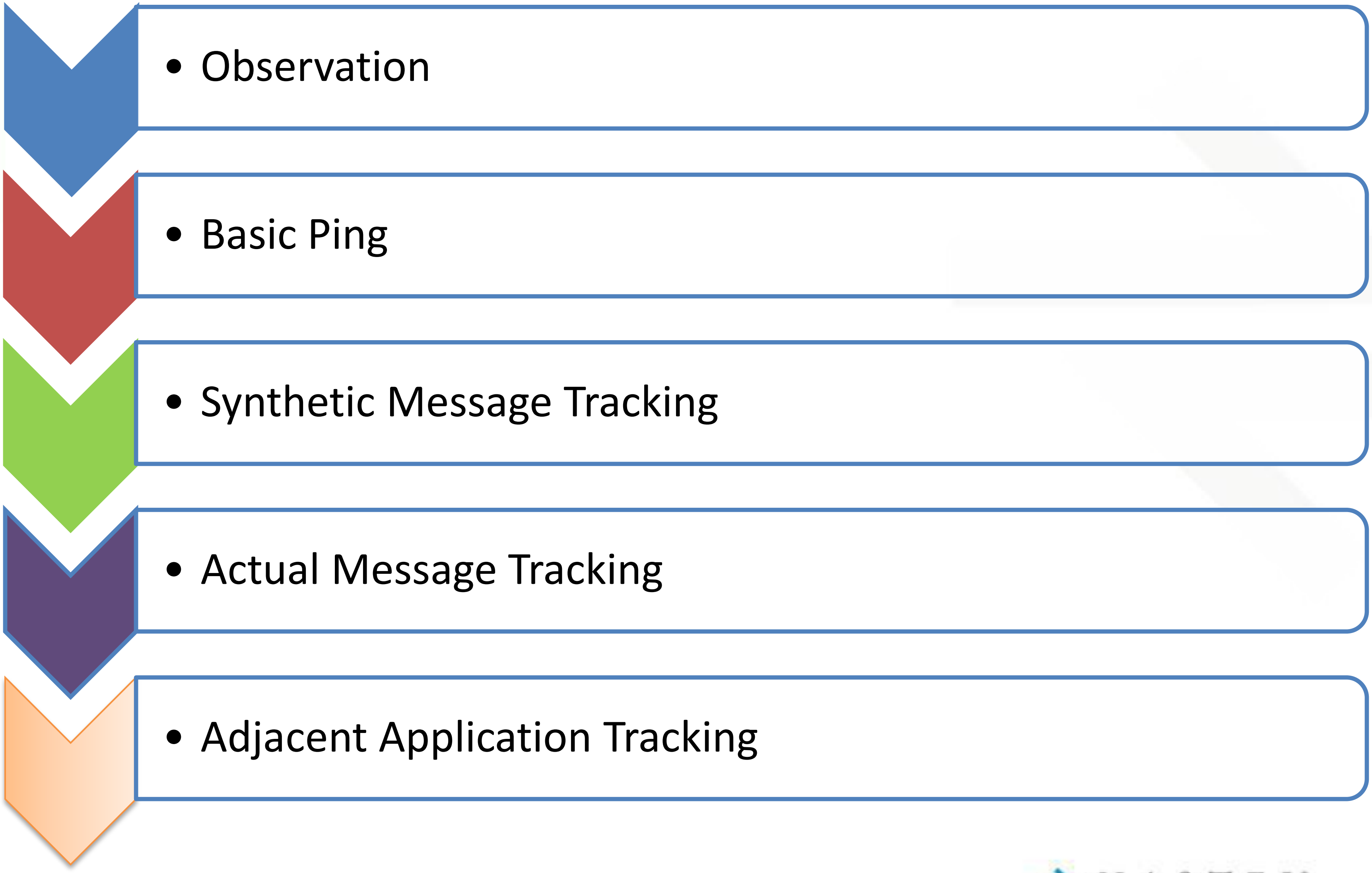
Synthetics Workloads

- Use a standard set of activities
- Do the same thing over and over and should produce a consistent result

Real Workloads

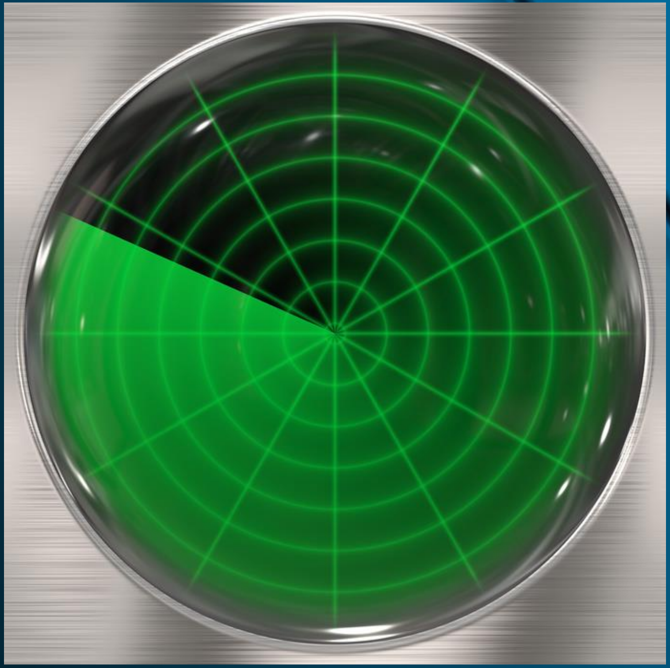
- Measure real messages flowing through the systems
- Subject to application and usage differences

Where to Start?



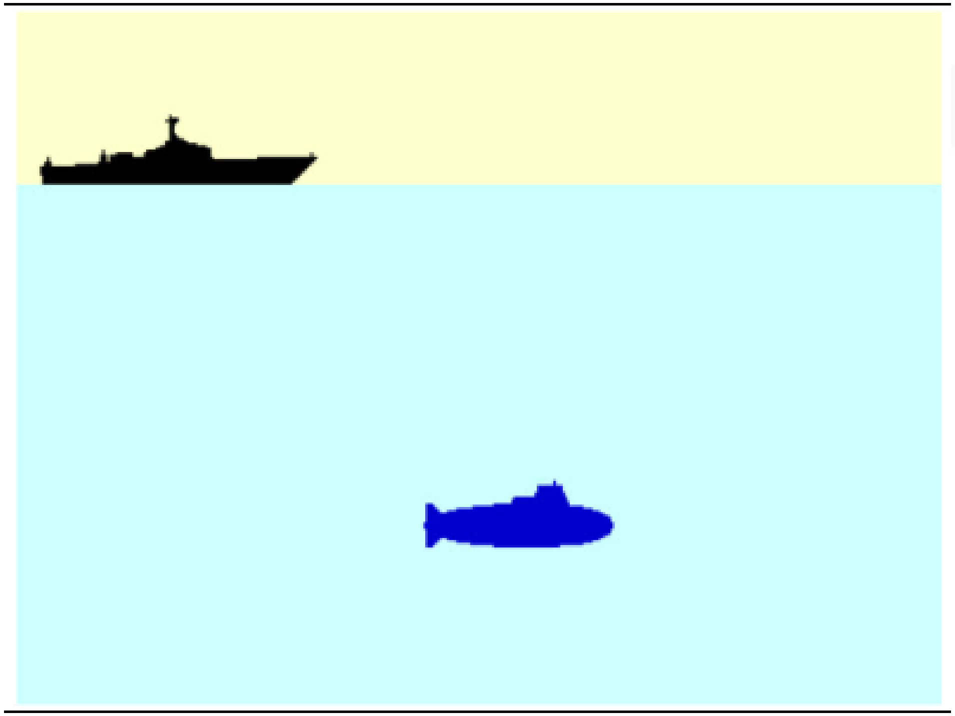
- Observation
- Basic Ping
- Synthetic Message Tracking
- Actual Message Tracking
- Adjacent Application Tracking

Starting with a Basic Ping



Start with a Basic “Ping”

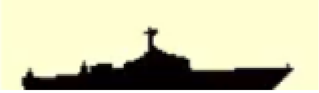
Periodically send a “ping” to a queue manager and measure the component parts



Active sonar. Copyright University of Rhode Island

What do we need?

Ping Component

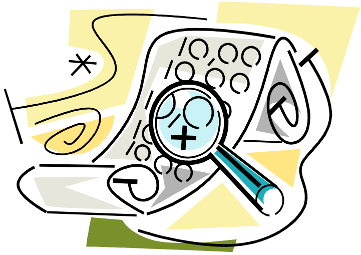


- To Initiate the request
- Configurable to send different size batches and message sizes

Echo Component

- Listen for requests
- Sends a response

Analysis



- Measure the results
- Capture them into repository
- Analyze and produce results
- Alert to anomalous behavior

The Ping Component

In WebSphere MQ terms

- Puts a message to a queue
- Waits for the reply to be sent back
- Reports the results

Options

- A MQ program that you could write
- A script to call an MQSC Script
- 3rd party programs
  - Including (free) AutoPilot® MQSonar™ from Nastel

The Echo Component

In WebSphere MQ terms

- Listens on a queue for a message to arrive
- Puts a reply back to a “reply” queue

The WebSphere MQ “Command Server”

- Listens on Command Queue
- Supports a ping command

Alternatives

- A Program you write
- 3rd party programs
  - Including (free) AutoPilot® MQSonar™ from Nastel

The Analysis Component

Maintains a Historical Record

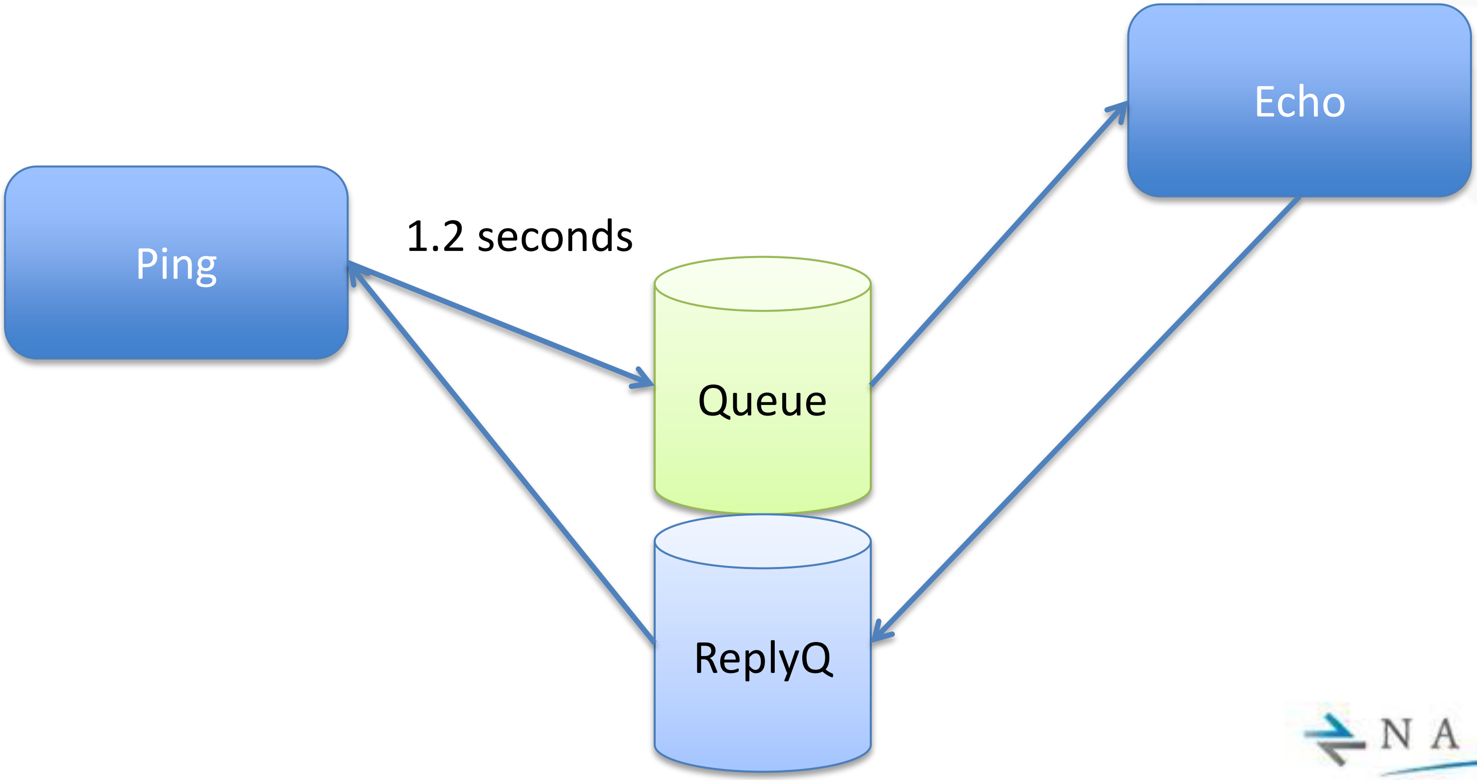
Analyzes the results

Options

- A program that you could write
- Interface with existing tooling
- Spreadsheet
- 3rd party options
  - Including AutoPilot® M6 from Nastel

In Operation

An Example



But Wait! There’s More!

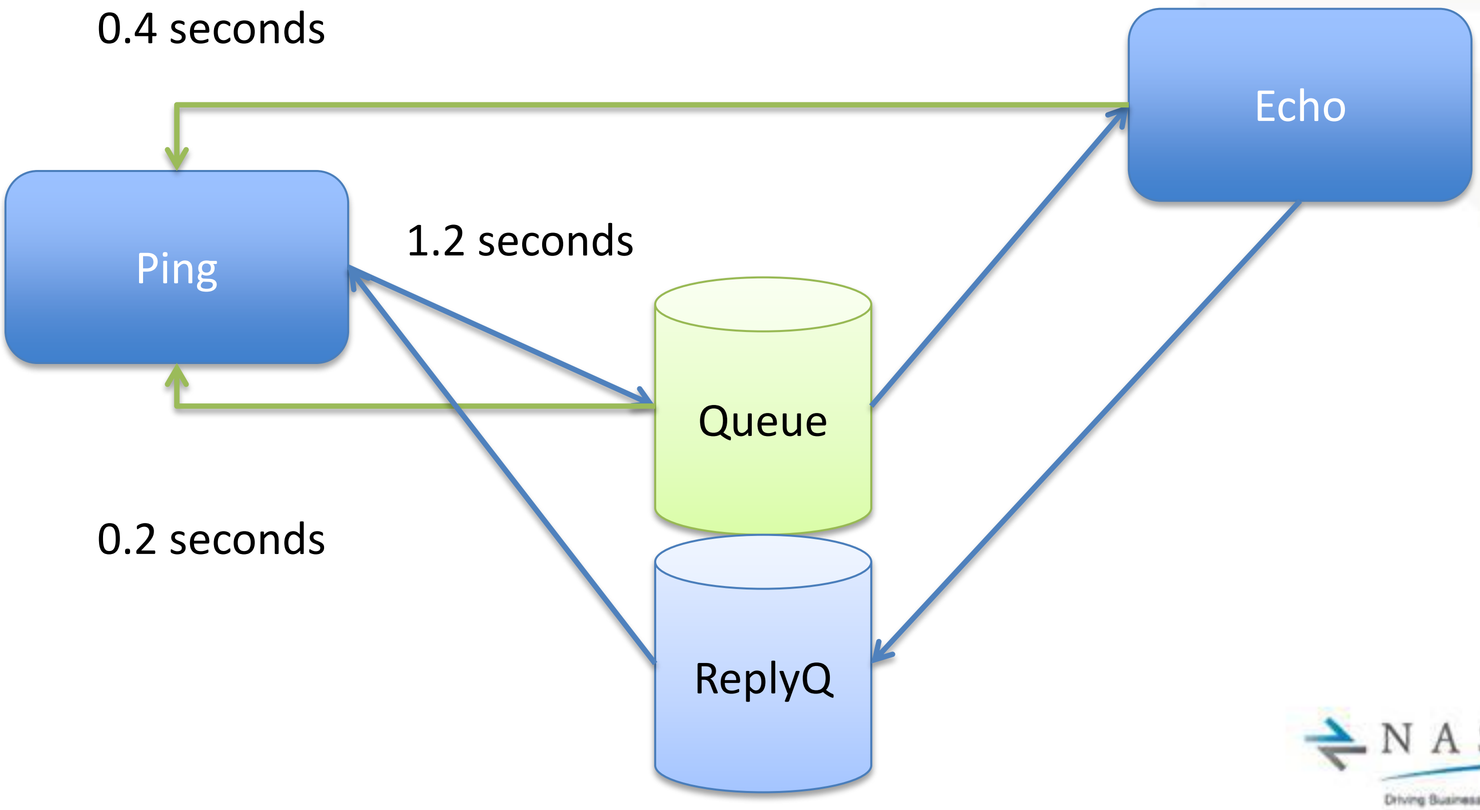
Be

me

Extension #1

With Programming options we can measure more than Total Time

- Confirm on Arrival – a report message when placed on Queue
- Confirm on Delivery – a report message when delivered to application



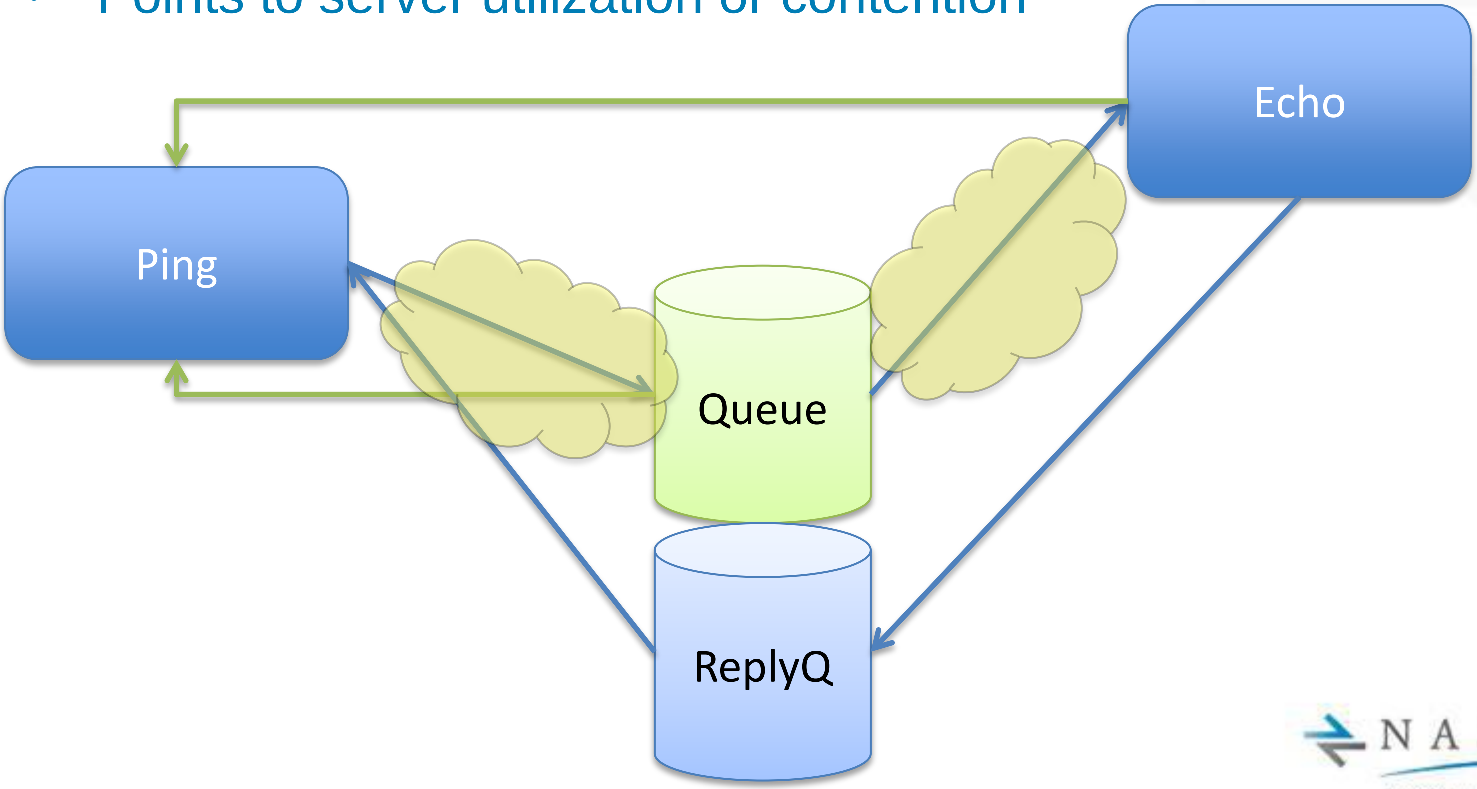
Why this is Valuable

High Time to Confirm on Arrival

- Points to transmission path

Large time difference between Confirm on Arrival and Confirm on Delivery

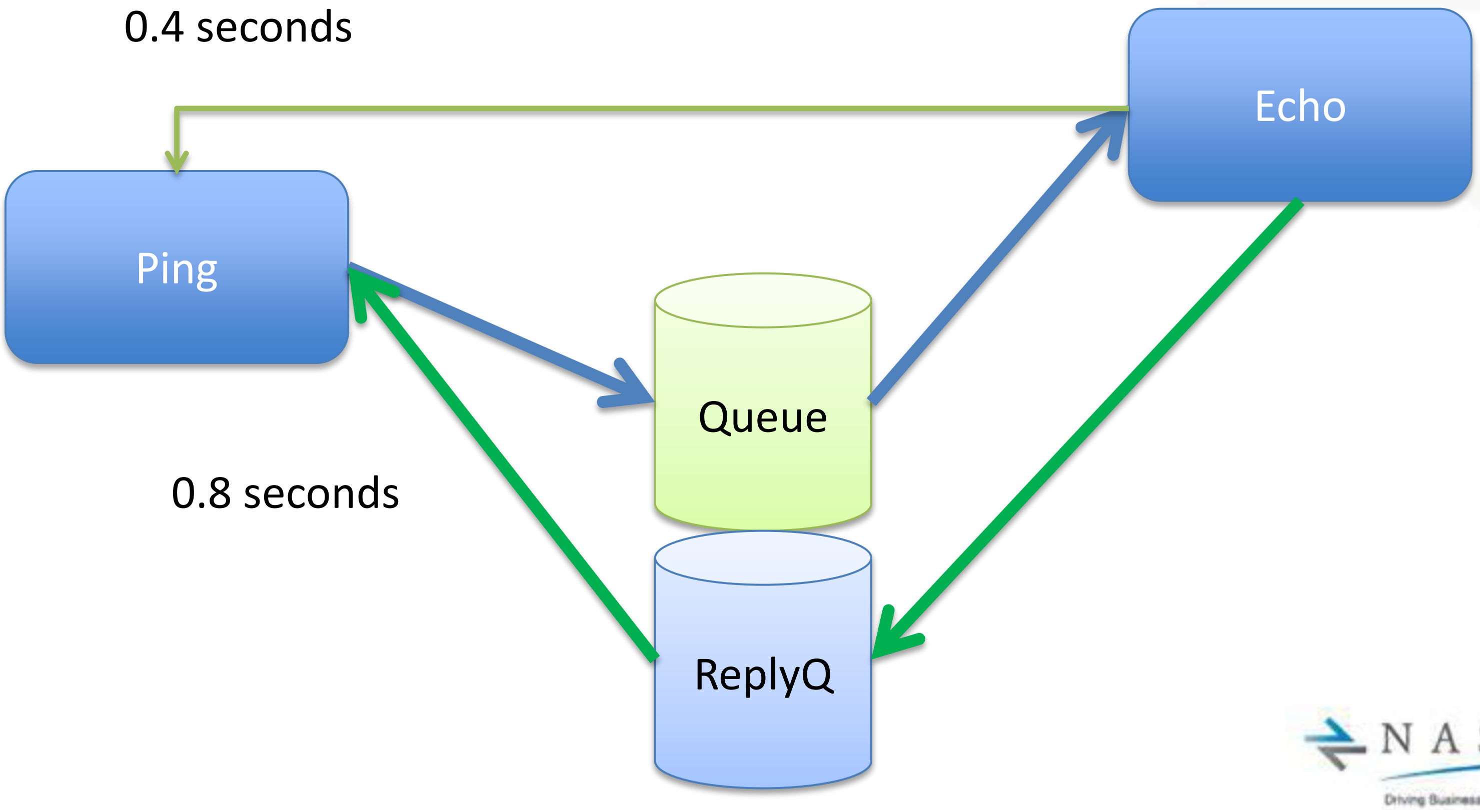
- Points to server utilization or contention



Why this is Valuable

Difference between time spent outbound (propagation) and time spent on the return (reflection)

- Shows which path is contributing factor



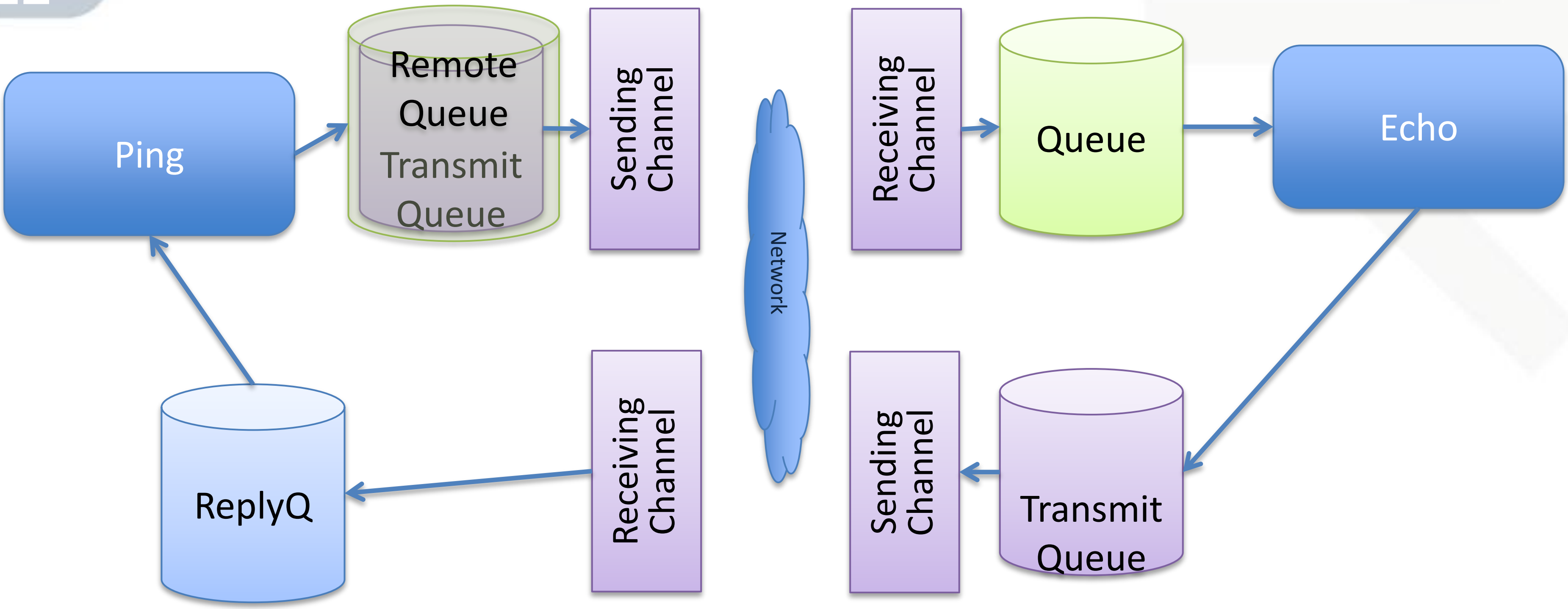
But Wait! There's More!

R ng

Extension #2

WebSphere MQ provides remote queuing

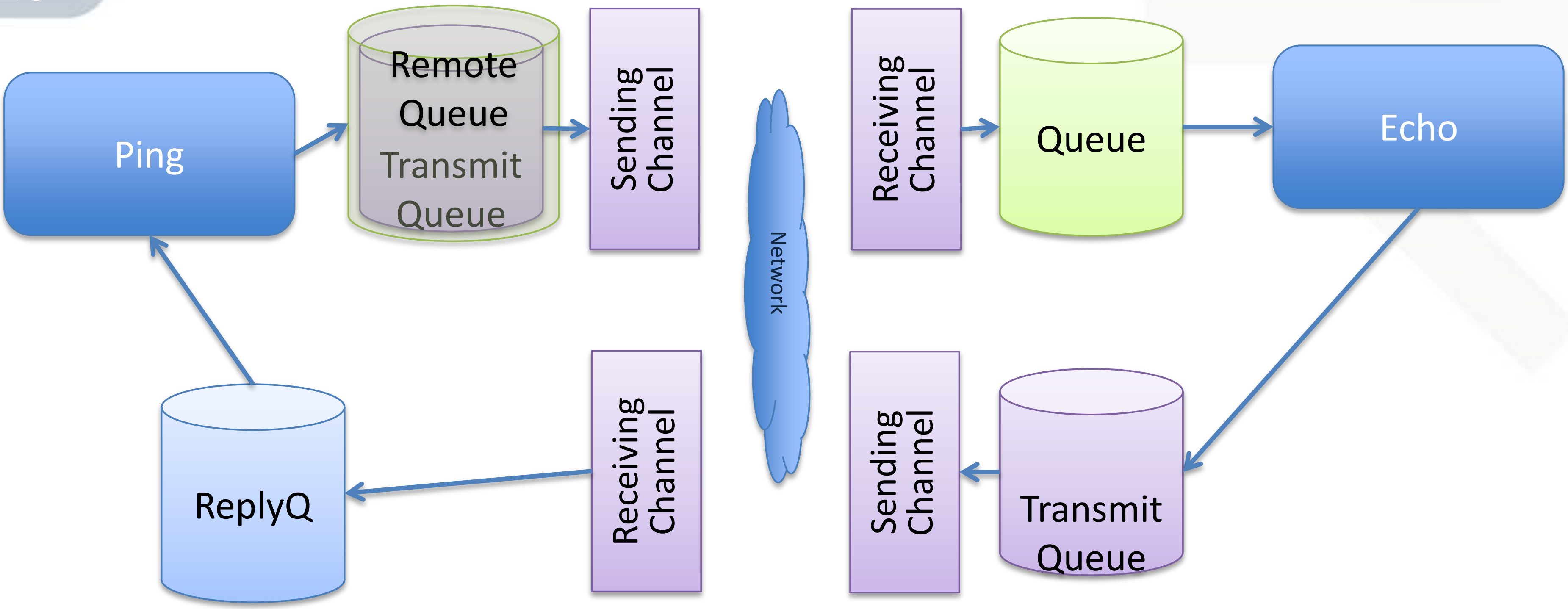
- Ping and Echo can be on different Queue Managers



Why this is Valuable

Same visibility as before across the intercommunication layer

- Can be done with or without Confirmation options



Multiple Queue Manager Scenarios

Identify slowdowns in inter-queue manager communication

Identify queue managers that contribute to delays

Identify differences in behaviors of different queue managers

Validate Cluster configurations

Verify that a path from one sending application to the receiving application is properly configured

- Includes broker, DataPower, and other..
- Requires the Echo component be an application program

Analysis

Statistics Summary

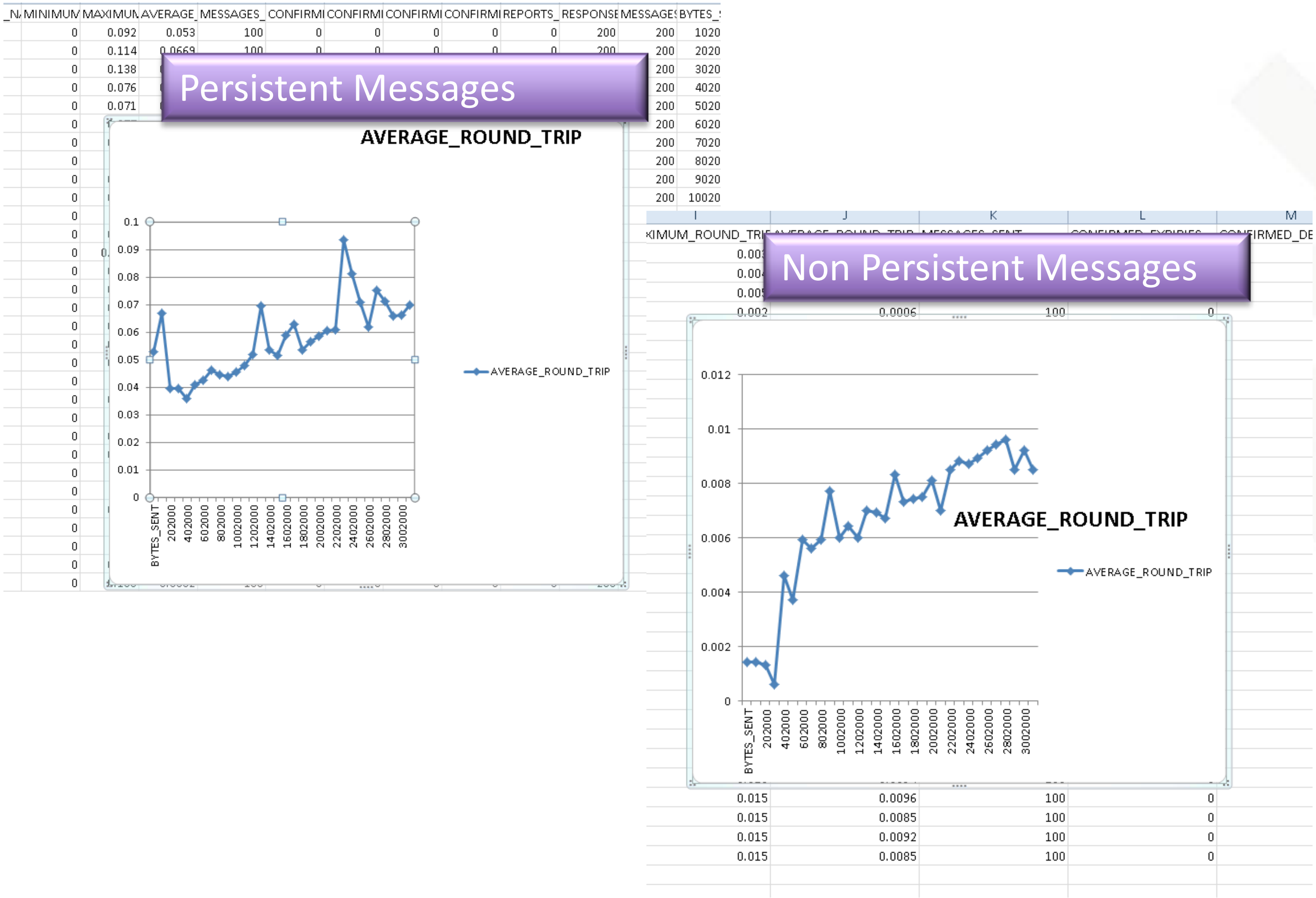
The following statistics can be derived from the samples

- Average and maximum round trip times
- Total time for an entire batch of messages
- Propagation Time (from Ping to Echo)
- Reflection Time (from Echo to Ping)
- Message rates inbound and inbound
    - Actual and Theoretical
- Data rates outbound and inbound

More Statistics that can be produced

MINIMUM_ROUND_TRIP	The minimum time for the reply to be returned for the request in seconds.
MAXIMUM_ROUND_TRIP	The maximum time for the reply to be returned for the request in seconds.
AVERAGE_ROUND_TRIP	The average time for the reply to be returned for the request in seconds for all messages sent in a batch.
AVERAGE_PROPAGATION_TIME	Average time in seconds for the message to propagate from the ping component to the echo component and the reply to be sent. For this statistic to be accurate, the time stamps between the servers sending the ping and sending the echo must be synchronized.
AVERAGE_REFLECTION_TIME	Average time in seconds for the message to reflect back from the echo component to the ping component and the reply to be read. For this statistic to be accurate, the time stamps between the servers sending the ping and sending the echo must be synchronized.
MESSAGES_SENT	The total number of messages sent included in this report.
CONFIRMED_EXPIRIES	Number of messages which expired before being delivered to the echo application (if coe specified).
CONFIRMED_DELIVERIES	Number of messages which were delivered to the echo application (if cod specified).
CONFIRMED_ARRIVALS	Number of messages which arrived to the echo application (if coa specified).
CONFIRMED_EXCEPTIONS	Number of messages which resulted in exceptions (if coe specified).
REPORTS_RECEIVED	Total number of confirmation report messages received.
RESPONSES_RECEIVED	Total number of response messages received.
MESSAGES_RECEIVED	Total number of response and report messages received.
BYTES_SENT	Total number of bytes sent.
BYTES_RECEIVED	Total number of bytes received.
RESPONSE_REQUEST_RATIO	The ratio of responses received to requests sent. A value of 100 means that all messages sent received responses.
General Performance Indicators	
TOTAL_PUT_TIME	The total time in seconds to put all of the messages
TOTAL_GET_TIME	The total time in seconds spent waiting for the responses to arrive
AVERAGE_PUT_RATE	The potential messages put rate calculated based on messages processed.
AVERAGE_PUT_BYTES_PER_SEC	Average put bytes per second
AVERAGE_GET_RATE	The potential messages get rate calculated based on messages processed.
AVERAGE_GET_BYTES_PER_SEC	Average get bytes per second
PUT_GET_RATIO	The ratio of put rate to get rate. A value greater than one means that MQSonar could put messages faster than get responses.
Message Performance Indicators	
GROSS_ROUND_TRIP_RATE	Message throughput rate (request + response) / time
GROSS_ROUND_BYTES_PER_SEC	Throughput (request + response) bytes per second
EFFECTIVE_ROUND_TRIP_RATE	Effective throughput rate (request + response + report) / time
CONFIRMATION_OVERHEAD	Percent of messages resulting from report options
AVERAGE_ARRIVAL_RATE	Average rate messages arrived at the destination (based on coa messages)
AVERAGE_DELIVERY_RATE	Average rate messages arrived at the destination (based on cod messages)
AVERAGE_MSG_LATENCY	Average time between arrival and delivery and number of messages on queue
AVERAGE_MSG_LATENCY_WITH_QDEPTH	Average time between arrival and delivery and number of messages on queue with queue depth
MAXIMUM_MSG_LATENCY	Maximum time between arrival and delivery and number of messages on queue
MAXIMUM_MSG_LATENCY_WITH_QDEPTH	Maximum time between arrival and delivery and number of messages on queue with queue depth
TOTAL_BATCH_TIME	The Elapsed time in seconds to process the entire batch.
TEST_COMPLETION_CODE	Completion code for the test. 0 indicates that all processing was normal.

Example Comparison using Excel



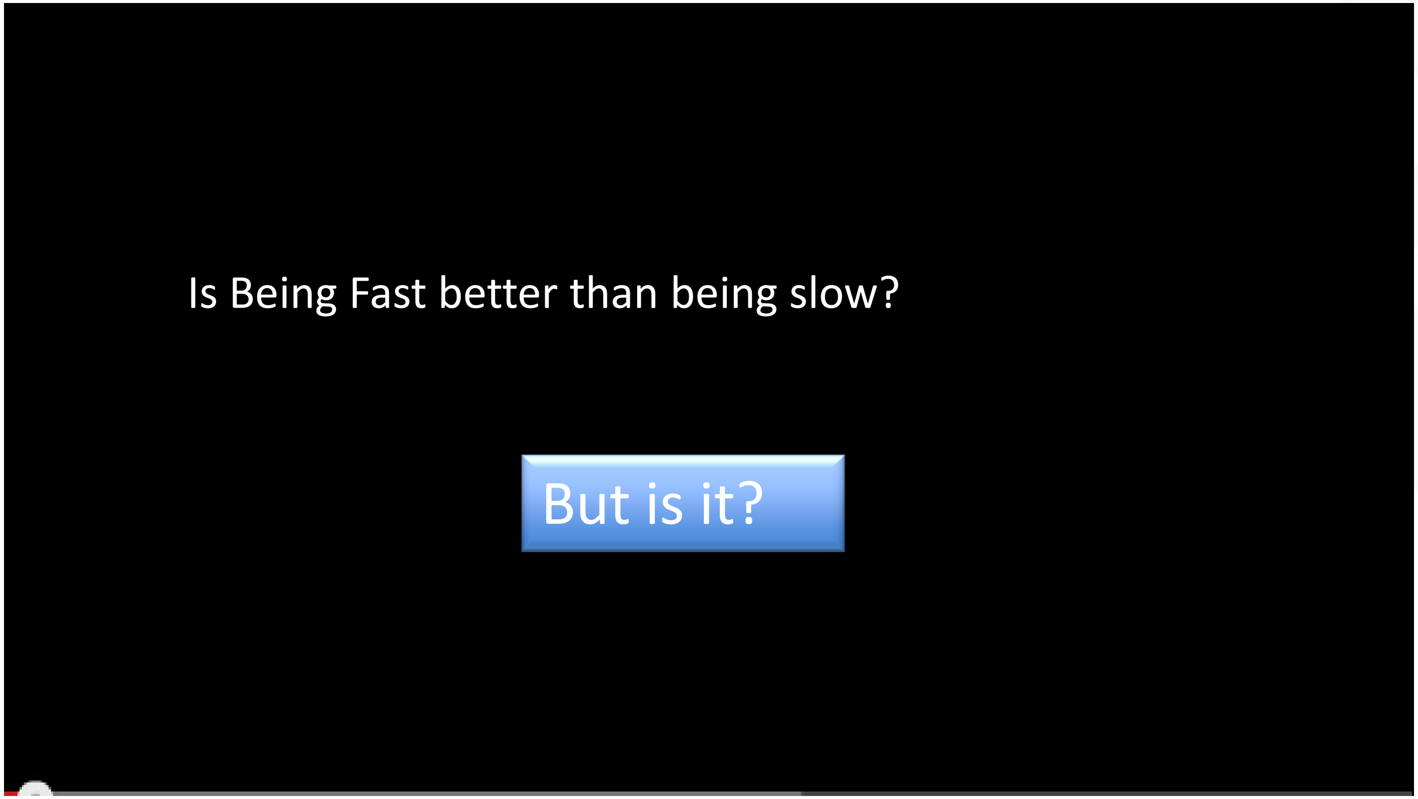
Persistent Messages

_N	MINIMUM	MAXIMUM	AVERAGE_	MESSAGES_	CONFIRMI	CONFIRMI	CONFIRMI	CONFIRMI	REPORTS_	RESPONSE	MESSAGES	BYTES_
	0	0.092	0.053	100	0	0	0	0	0	200	200	1020
	0	0.114	0.0669	100	0	0	0	0	0	200	200	2020
	0	0.138									200	3020
	0	0.076									200	4020
	0	0.071									200	5020
	0										200	6020
	0										200	7020
	0										200	8020
	0										200	9020
	0										200	10020
	0											
	0											
	0	0.										
	0											
	0											
	0											
	0											
	0											
	0											
	0											
	0											
	0											
	0											
	0											
	0											
	0											
	0											
	0											
	0											
	0											
	0											

Non Persistent Messages

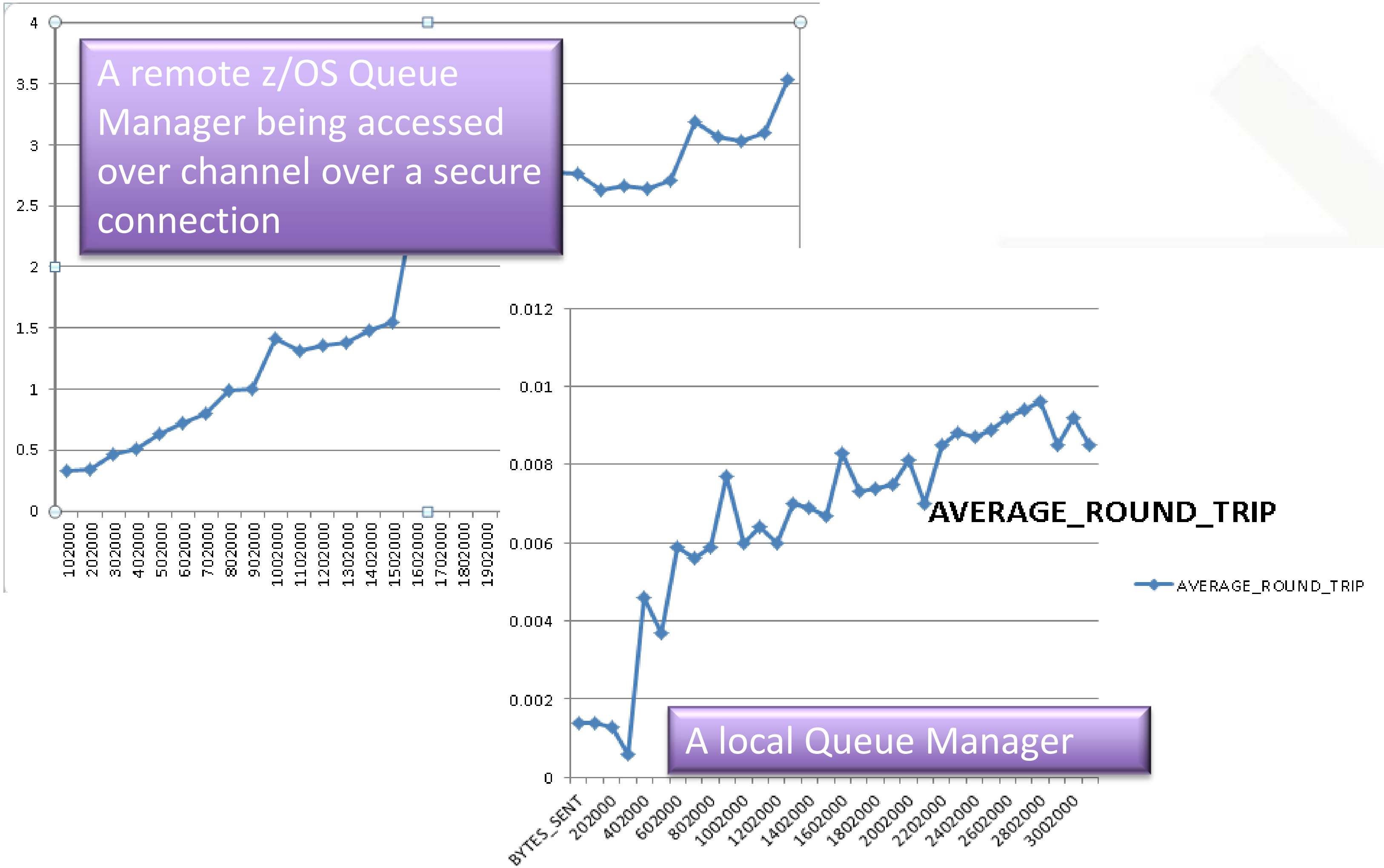
I	J	K	L	M
XIMUM_ROUND_TRI	AVERAGE_ROUND_TRIP	MESSAGES_SENT	CONFIRMED_EXPIRIES	CONFIRMED_DE
0.003				
0.004				
0.005				
0.002	0.0006	100	0	
0.015	0.0096	100	0	
0.015	0.0085	100	0	
0.015	0.0092	100	0	
0.015	0.0085	100	0	

Is Faster Better than Slower?



Is Being Fast better than being slow?

Another Comparison



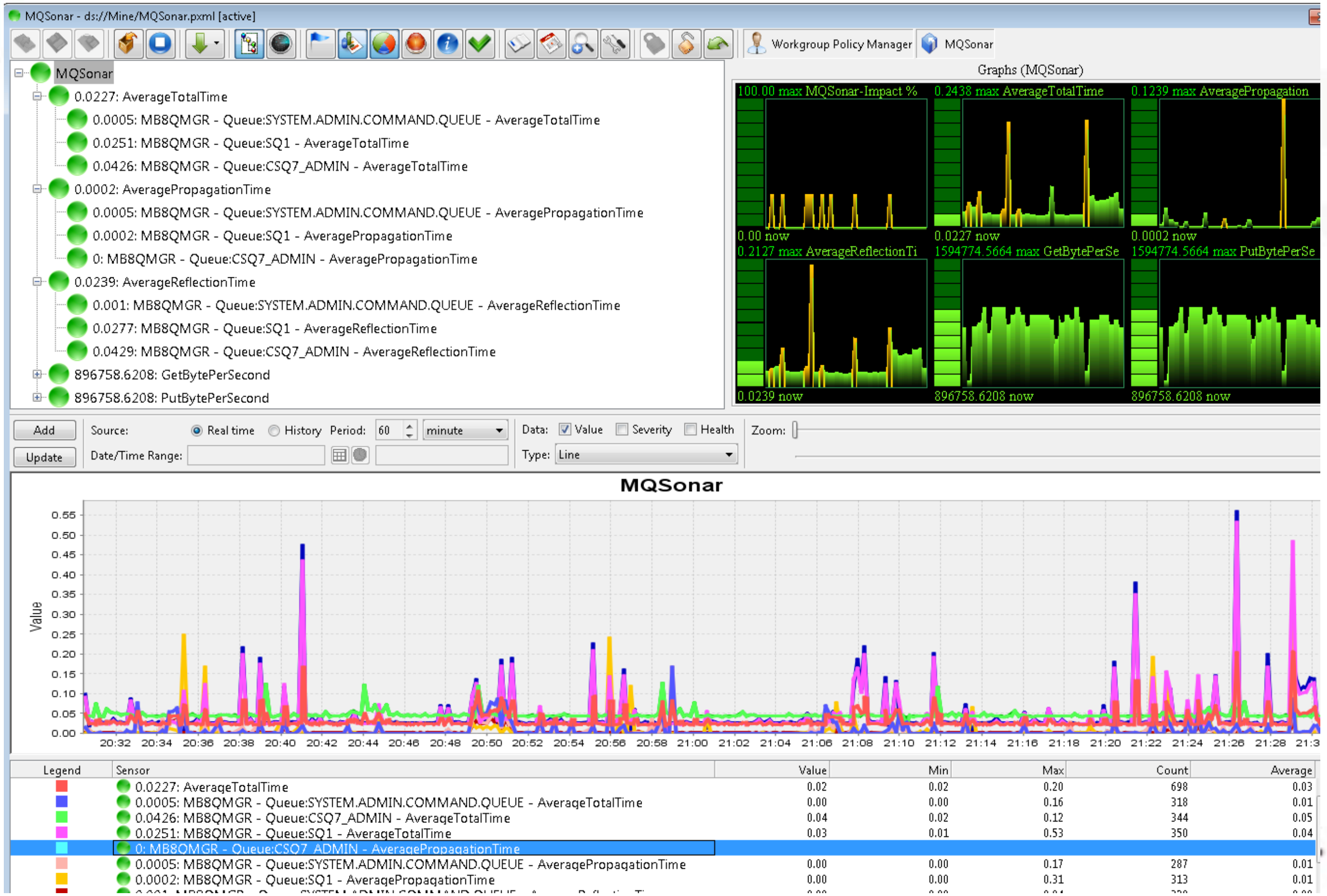
A remote z/OS Queue Manager being accessed over channel over a secure connection

A local Queue Manager

Integration with AutoPilot (optional)

Statistics Collected are published to AutoPilot Server

- Alert on abnormal results
- Historical trending



MQSonar - ds://Mine/MQSonar.pxml [active]

- MQSonar
  - 0.0227: AverageTotalTime
    - 0.0005: MB8QMGR - Queue:SYSTEM.ADMIN.COMMAND.QUEUE - AverageTotalTime
    - 0.0251: MB8QMGR - Queue:SQ1 - AverageTotalTime
    - 0.0426: MB8QMGR - Queue:CSQ7_ADMIN - AverageTotalTime
  - 0.0002: AveragePropagationTime
    - 0.0005: MB8QMGR - Queue:SYSTEM.ADMIN.COMMAND.QUEUE - AveragePropagationTime
    - 0.0002: MB8QMGR - Queue:SQ1 - AveragePropagationTime
    - 0: MB8QMGR - Queue:CSQ7_ADMIN - AveragePropagationTime
  - 0.0239: AverageReflectionTime
    - 0.001: MB8QMGR - Queue:SYSTEM.ADMIN.COMMAND.QUEUE - AverageReflectionTime
    - 0.0277: MB8QMGR - Queue:SQ1 - AverageReflectionTime
    - 0.0429: MB8QMGR - Queue:CSQ7_ADMIN - AverageReflectionTime
  - 896758.6208: GetBytePerSecond
  - 896758.6208: PutBytePerSecond

Source: ● Real time ○ History Period: 60 minute

Data: ☑ Value ☐ Severity ☐ Health

Type: Line

Legend	Sensor	Value	Min	Max	Count	Average
🟥	0.0227: AverageTotalTime	0.02	0.02	0.20	698	0.03
🟦	0.0005: MB8QMGR - Queue:SYSTEM.ADMIN.COMMAND.QUEUE - AverageTotalTime	0.00	0.00	0.16	318	0.01
🟩	0.0426: MB8QMGR - Queue:CSQ7_ADMIN - AverageTotalTime	0.04	0.02	0.12	344	0.05
🟪	0.0251: MB8QMGR - Queue:SQ1 - AverageTotalTime	0.03	0.01	0.53	350	0.04
🟦	0: MB8QMGR - Queue:CSQ7_ADMIN - AveragePropagationTime					
🟥	0.0005: MB8QMGR - Queue:SYSTEM.ADMIN.COMMAND.QUEUE - AveragePropagationTime	0.00	0.00	0.17	287	0.01
🟨	0.0002: MB8QMGR - Queue:SQ1 - AveragePropagationTime	0.00	0.00	0.31	313	0.01

Tracking Messages

Concepts

Requirements for the ping technique:

- “Special” messages
- “Special” message paths

Using MQ Exit points you can capture the actual MQ calls and messages without changing them

- Capture synthetic message flows
- Capture real message flows using actual application logic
- Richer detail on the message flow

Message Intercept

Captures details about the message

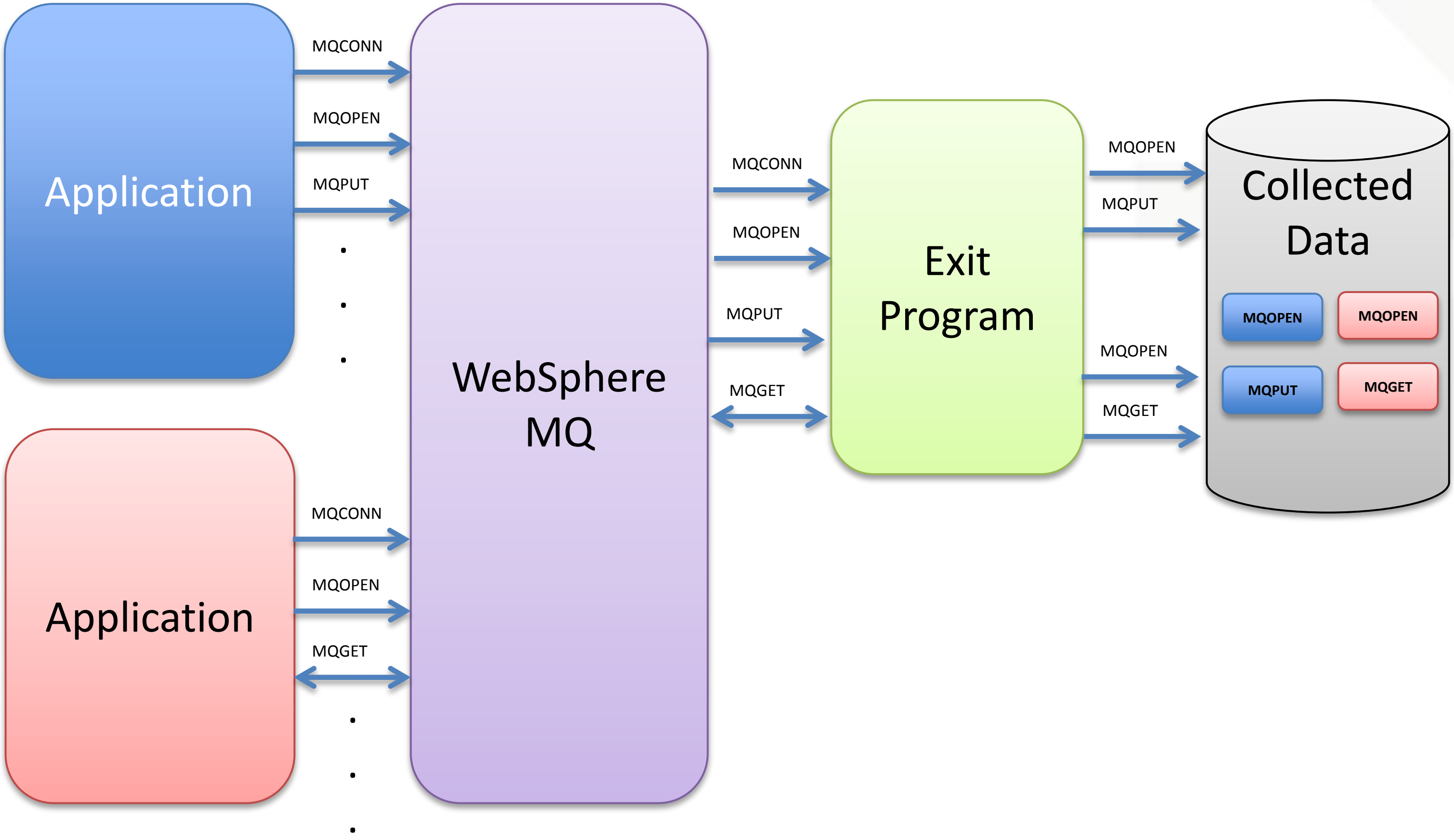
- Header details (time sent, size, …)
- Application details
- All or part of payload

Options

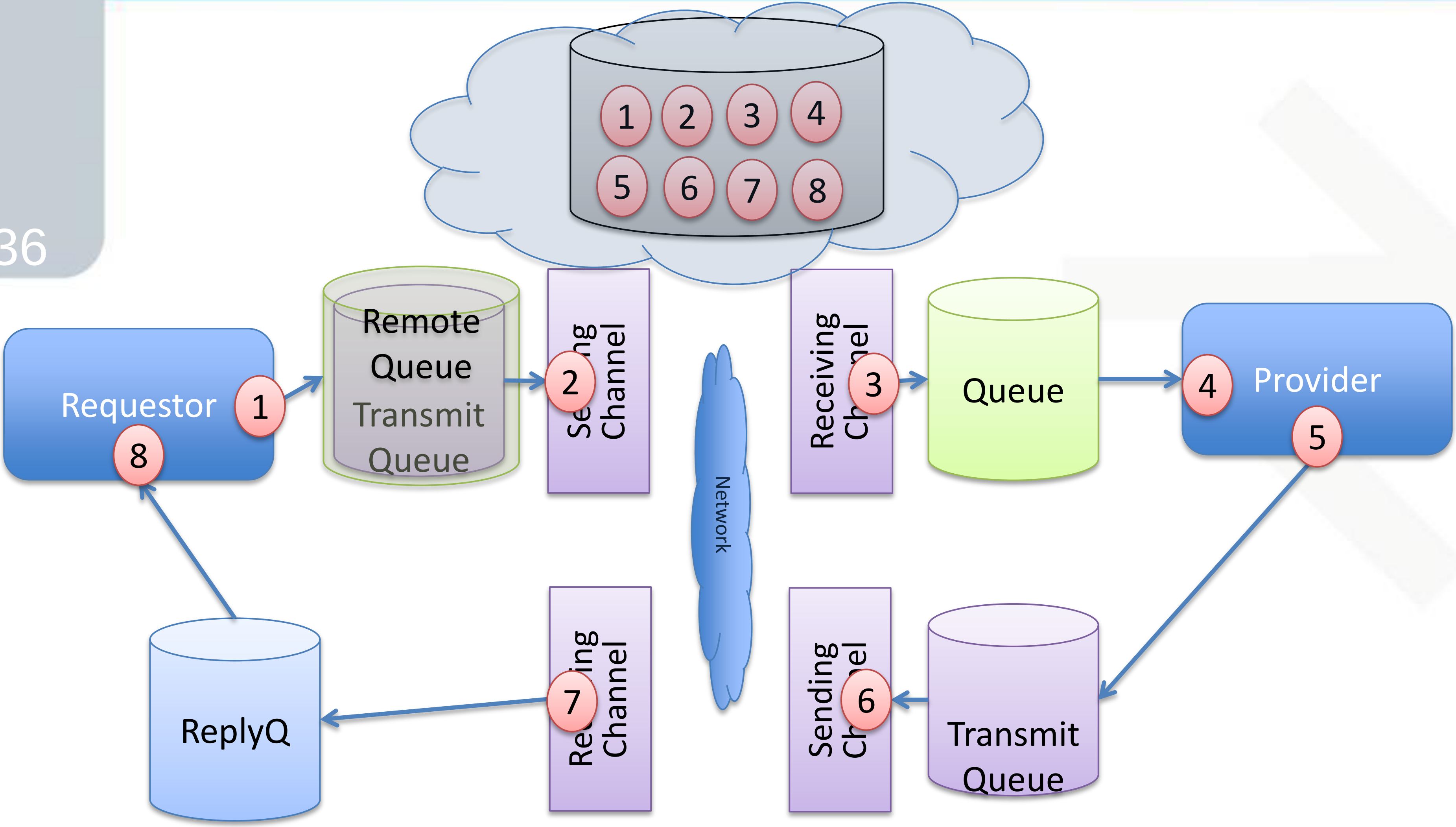
- A MQ program that you could write
- IBM Support Pac (mirrorq, MA0W)
- 3rd party programs
  - Including AutoPilot® TransactionWorks ™ from Nastel

How it works

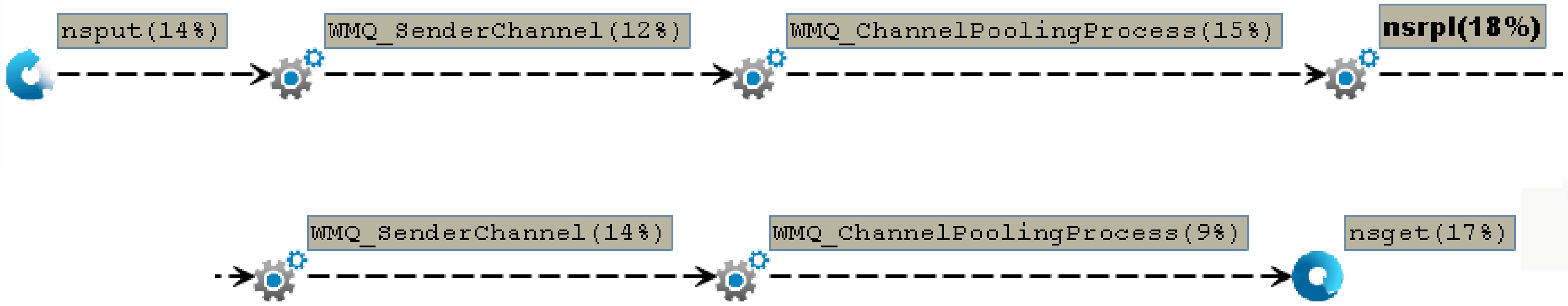
Message Intercept

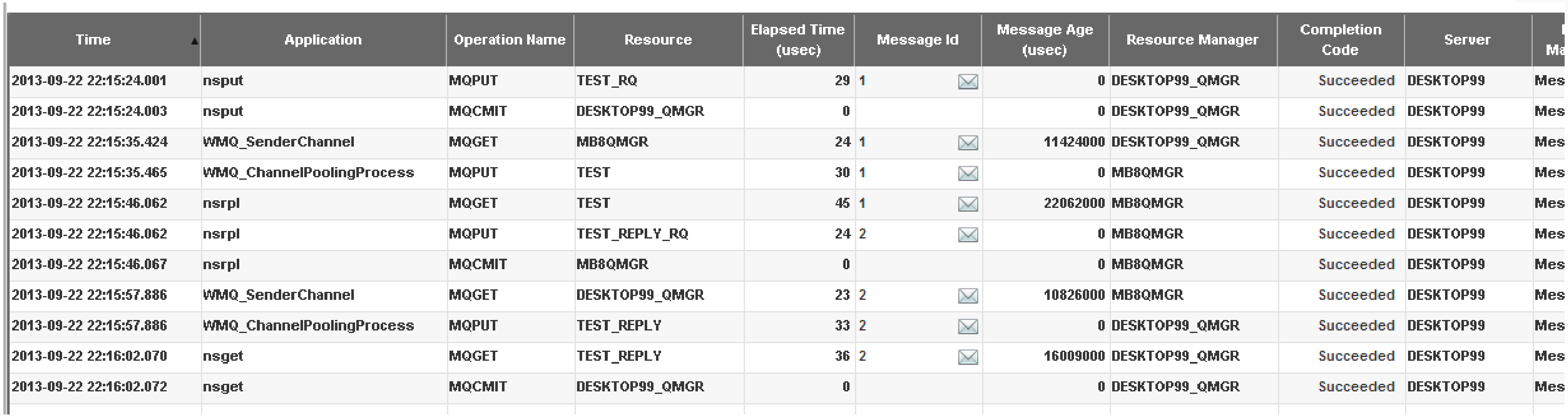


With Remote Queueing



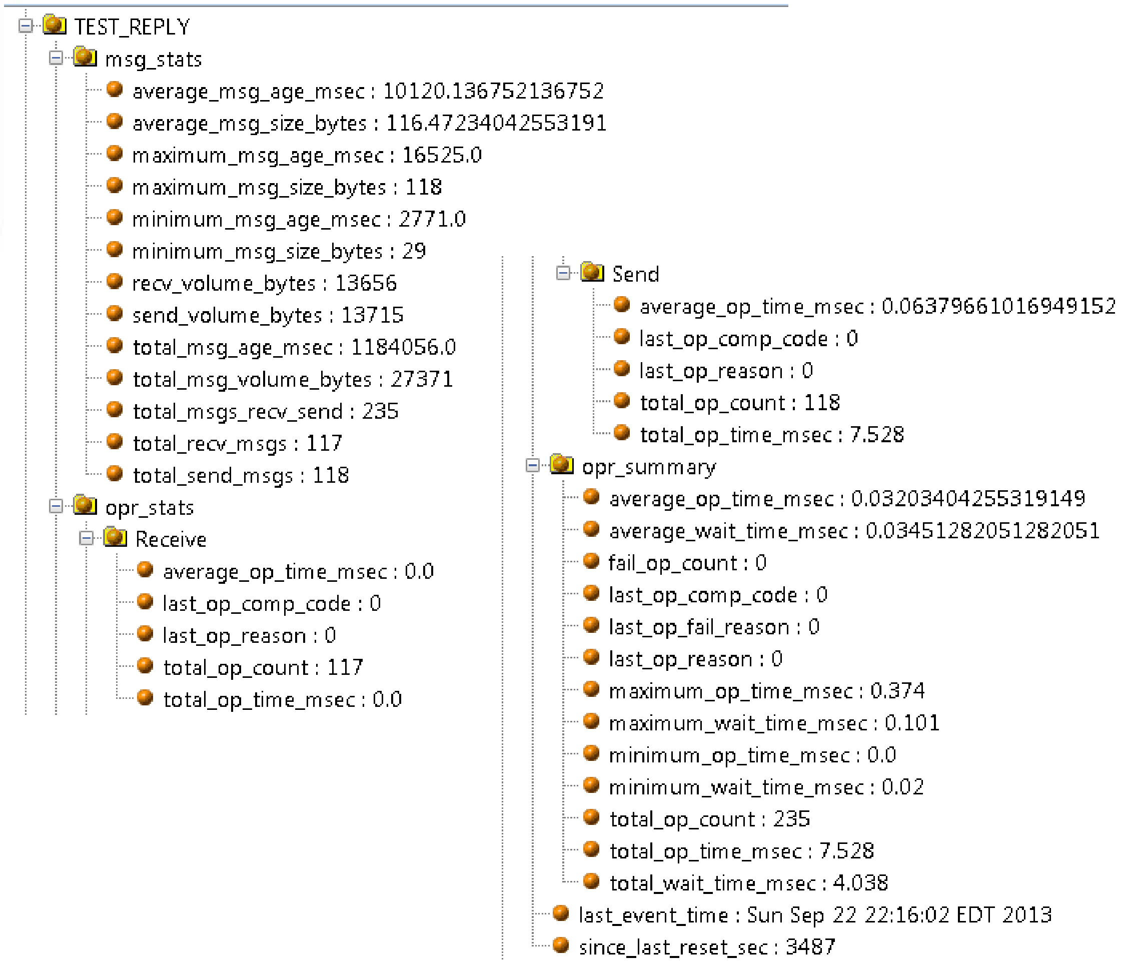
Simple Message Flow





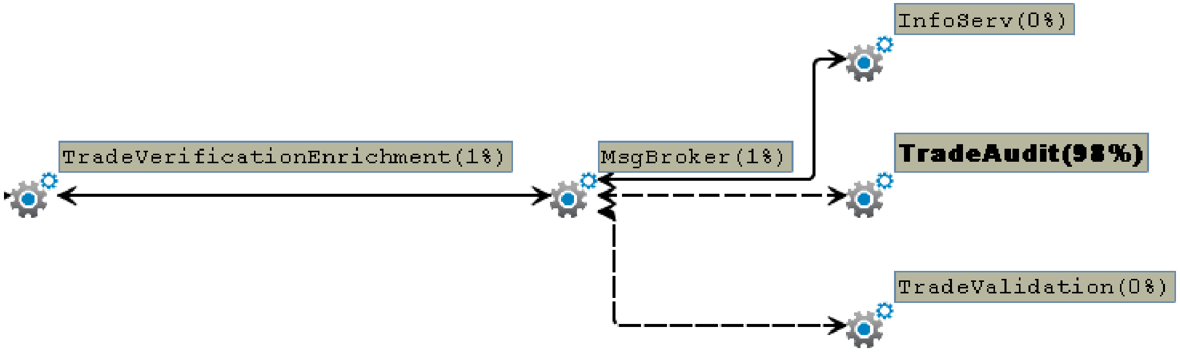
Time	Application	Operation Name	Resource	Elapsed Time (usec)	Message Id	Message Age (usec)	Resource Manager	Completion Code	Server	Ma
2013-09-22 22:15:24.001	nsput	MQPUT	TEST_RQ	29	1 ✉	0	DESKTOP99_QMGR	Succeeded	DESKTOP99	Mes
2013-09-22 22:15:24.003	nsput	MQCMIT	DESKTOP99_QMGR	0		0	DESKTOP99_QMGR	Succeeded	DESKTOP99	Mes
2013-09-22 22:15:35.424	WMQ_SenderChannel	MQGET	MB8QMGR	24	1 ✉	11424000	DESKTOP99_QMGR	Succeeded	DESKTOP99	Mes
2013-09-22 22:15:35.465	WMQ_ChannelPoolingProcess	MQPUT	TEST	30	1 ✉	0	MB8QMGR	Succeeded	DESKTOP99	Mes
2013-09-22 22:15:46.062	nsrpl	MQGET	TEST	45	1 ✉	22062000	MB8QMGR	Succeeded	DESKTOP99	Mes
2013-09-22 22:15:46.062	nsrpl	MQPUT	TEST_REPLY_RQ	24	2 ✉	0	MB8QMGR	Succeeded	DESKTOP99	Mes
2013-09-22 22:15:46.067	nsrpl	MQCMIT	MB8QMGR	0		0	MB8QMGR	Succeeded	DESKTOP99	Mes
2013-09-22 22:15:57.886	WMQ_SenderChannel	MQGET	DESKTOP99_QMGR	23	2 ✉	10826000	MB8QMGR	Succeeded	DESKTOP99	Mes
2013-09-22 22:15:57.886	WMQ_ChannelPoolingProcess	MQPUT	TEST_REPLY	33	2 ✉	0	DESKTOP99_QMGR	Succeeded	DESKTOP99	Mes
2013-09-22 22:16:02.070	nsget	MQGET	TEST_REPLY	36	2 ✉	16009000	DESKTOP99_QMGR	Succeeded	DESKTOP99	Mes
2013-09-22 22:16:02.072	nsget	MQCMIT	DESKTOP99_QMGR	0		0	DESKTOP99_QMGR	Succeeded	DESKTOP99	Mes

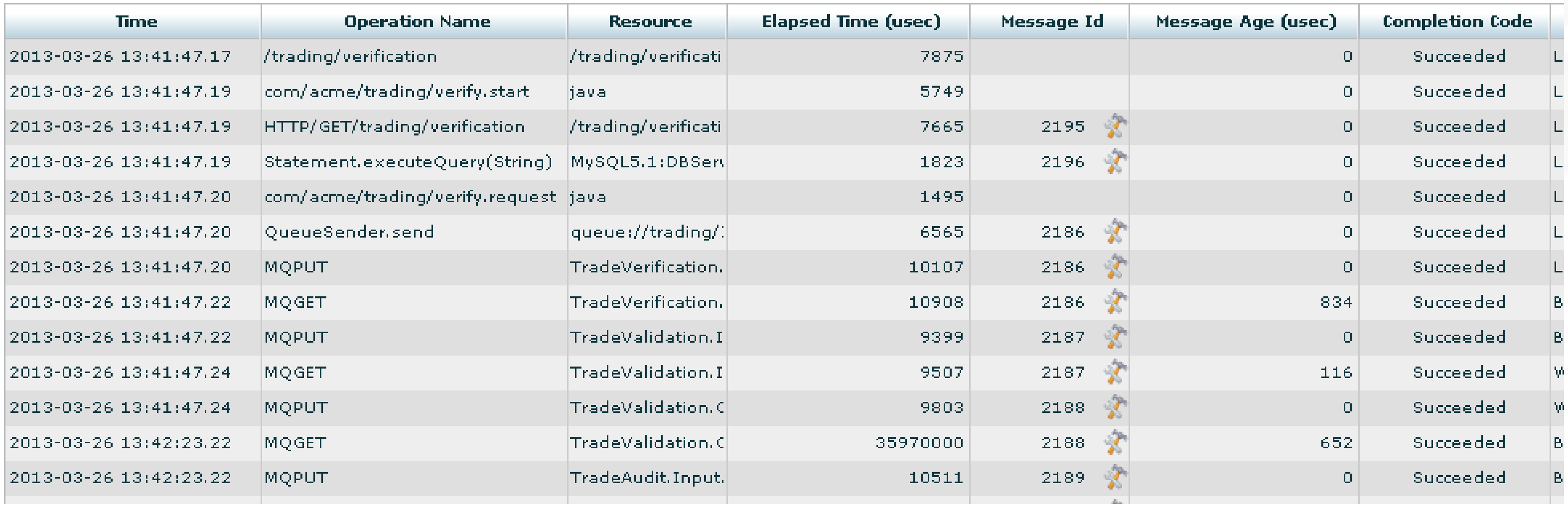
Increased Statistics Available



```
TEST_REPLY
  msg_stats
    average_msg_age_msec : 10120.136752136752
    average_msg_size_bytes : 116.47234042553191
    maximum_msg_age_msec : 16525.0
    maximum_msg_size_bytes : 118
    minimum_msg_age_msec : 2771.0
    minimum_msg_size_bytes : 29
    recv_volume_bytes : 13656
    send_volume_bytes : 13715
    total_msg_age_msec : 1184056.0
    total_msg_volume_bytes : 27371
    total_msgs_recv_send : 235
    total_recv_msgs : 117
    total_send_msgs : 118
  opr_stats
    Receive
      average_op_time_msec : 0.0
      last_op_comp_code : 0
      last_op_reason : 0
      total_op_count : 117
      total_op_time_msec : 0.0
    Send
      average_op_time_msec : 0.06379661016949152
      last_op_comp_code : 0
      last_op_reason : 0
      total_op_count : 118
      total_op_time_msec : 7.528
  opr_summary
    average_op_time_msec : 0.03203404255319149
    average_wait_time_msec : 0.03451282051282051
    fail_op_count : 0
    last_op_comp_code : 0
    last_op_fail_reason : 0
    last_op_reason : 0
    maximum_op_time_msec : 0.374
    maximum_wait_time_msec : 0.101
    minimum_op_time_msec : 0.0
    minimum_wait_time_msec : 0.02
    total_op_count : 235
    total_op_time_msec : 7.528
    total_wait_time_msec : 4.038
  last_event_time : Sun Sep 22 22:16:02 EDT 2013
  since_last_reset_sec : 3487
```

Cross Application Tracking





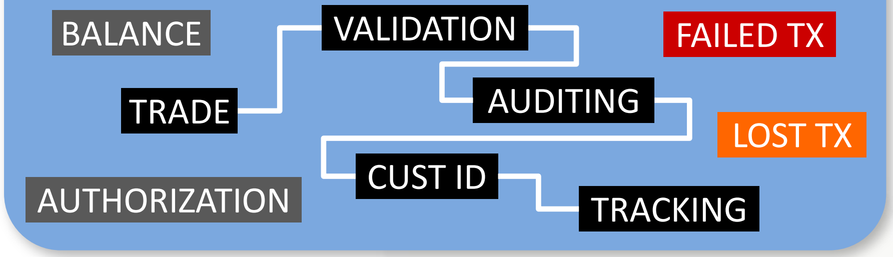
Time	Operation Name	Resource	Elapsed Time (usec)	Message Id	Message Age (usec)	Completion Code	
2013-03-26 13:41:47.17	/trading/verification	/trading/verificati	7875		0	Succeeded	L
2013-03-26 13:41:47.19	com/acme/trading/verify.start	java	5749		0	Succeeded	L
2013-03-26 13:41:47.19	HTTP/GET/trading/verification	/trading/verificati	7665	2195	0	Succeeded	L
2013-03-26 13:41:47.19	Statement.executeQuery(String)	MySQL5.1:DBSer	1823	2196	0	Succeeded	L
2013-03-26 13:41:47.20	com/acme/trading/verify.request	java	1495		0	Succeeded	L
2013-03-26 13:41:47.20	QueueSender.send	queue://trading/	6565	2186	0	Succeeded	L
2013-03-26 13:41:47.20	MQPUT	TradeVerification.	10107	2186	0	Succeeded	L
2013-03-26 13:41:47.22	MQGET	TradeVerification.	10908	2186	834	Succeeded	B
2013-03-26 13:41:47.22	MQPUT	TradeValidation.I	9399	2187	0	Succeeded	B
2013-03-26 13:41:47.24	MQGET	TradeValidation.I	9507	2187	116	Succeeded	W
2013-03-26 13:41:47.24	MQPUT	TradeValidation.C	9803	2188	0	Succeeded	W
2013-03-26 13:42:23.22	MQGET	TradeValidation.C	35970000	2188	652	Succeeded	B
2013-03-26 13:42:23.22	MQPUT	TradeAudit.Input.	10511	2189	0	Succeeded	B

Middleware-Centric Application Performance Monitoring

CEP Policy Engine

OPERATIONAL MONITORING

TRANSACTIONAL MONITORING


BALANCE
TRADE
AUTHORIZATION
VALIDATION
AUDITING
CUST ID
TRACKING
FAILED TX
LOST TX

APPLICATIONS

Trading Equities

Funds Transfers

Claims Processing

Order Handling

Payments Processing

WebSphere. TIBCO JMS

Java Microsoft® .NET

ESB

IBM DataPower Solace Systems®

Messaging Middleware

Application Servers

Enterprise Service Bus

SOA Appliances

INFRA STRUCTURE

Servers

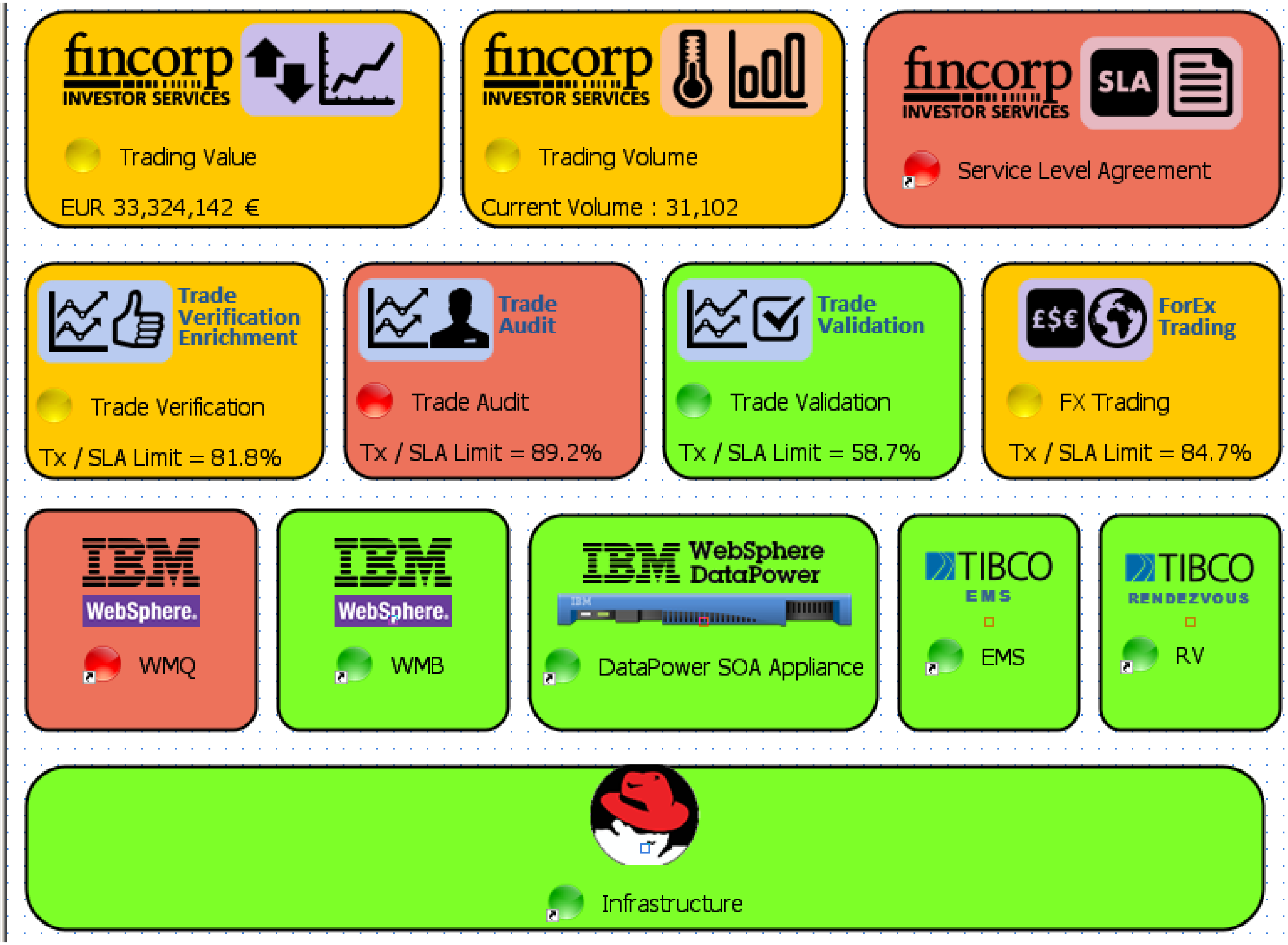
Network

SQL Databases

Storage

NASTEL®
Driving Business Transaction Performance®

Application Dashboard



fincorp INVESTOR SERVICES

Trading Value

EUR 33,324,142 €

fincorp INVESTOR SERVICES

Trading Volume

Current Volume : 31,102

fincorp INVESTOR SERVICES

SLA

Service Level Agreement

Trade Verification Enrichment

Trade Verification

Tx / SLA Limit = 81.8%

Trade Audit

Trade Audit

Tx / SLA Limit = 89.2%

Trade Validation

Trade Validation

Tx / SLA Limit = 58.7%

ForEx Trading

£$€

FX Trading

Tx / SLA Limit = 84.7%

IBM WebSphere.

WMQ

IBM WebSphere.

WMB

IBM WebSphere DataPower

DataPower SOA Appliance

TIBCO EMS

EMS

TIBCO RENDEZVOUS

RV

Infrastructure

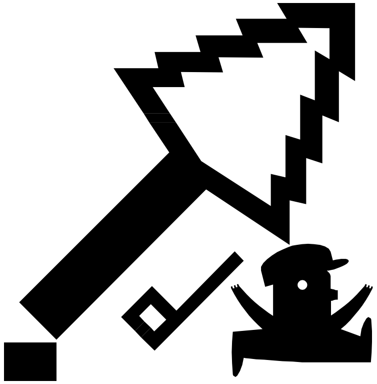
Using the MQ Recovery Log Files

Log files can be used to get insight into MQ message traffic

Automatically generated so no special setup required to collect them

But some challenges

- Capture recoverable messages only
- Don’t capture error conditions (queue full, not authorized)
- Not easily to expand to other applications
- Lack of real time awareness of problems



Summary

Summary

- Starting with a Simple benchmark to periodically ping a queue manager can be valuable
- Combining with other options, such as Confirm on Delivery increases the information that can be determined
- Performing this regularly provides insight into the changes in behavior of the queue manager
- Tracking actual messages increases the data available about the queue manager operation

Thank You!

Questions:
info@nastel.com

NASTEL®

Driving Business Transaction Performance®

www.nastel.com