APM Best Practices

High Availability

(Failover capabilities and techniques for CA APM, leading to High Availability)

Michael Sydor – Engineering Service Architect

Definitions

- Backup
  - Something we do to facilitate *recovery* of a component and its data
- Failover
  - A characteristic of a *component* within a service, where control is transferred to the next available component, in the same role
- High Availability
  - A characteristic of a *service*, comprised of many components
- Business Continuity (Disaster Recovery)
  - The characteristic of a business to maintain services after rare, catastrophic events

Agenda

- This is a complex topic

- What do we need failover to achieve?
- What does APM offer?
- What are the hard realities?
- What are the best solutions available today?

- Bringing the points to a close

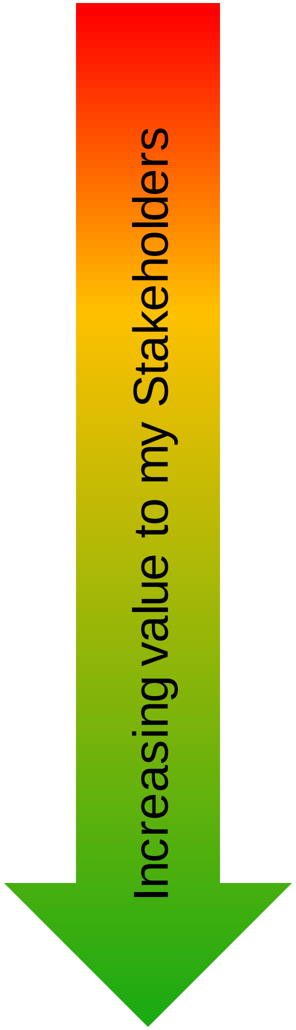
Best Practice Topics

Services Processes Skills Competencies

Skills & Gap Assessment	EM Mgmt Initial Triage Skills	Baseline Processes		Pre-Production Processes		Production Processes	
APM Skills Assessment	Application Assessment	QA Test Plans	Application Audit	Dashboard Strategies	EM Capacity Management	Deployment Planning	Critical App Assessment
Gap and Visibility Assessment	EM Deployment & Mgmt	QA Acceptance	Configuration Baselines	Alert Integration	CMTs and Advanced Tracers	Capacity Management and Planning	Firefighting Practice
Incident Review	Triage with Single Metrics	Quality Review and Escalation		Alert Review and Escalation	Identify and Generate New Instrumentation	Solution Certification	Staffing Strategies
EM Sizing and Capacity Forecast	Remote Analysis Techniques	Agent Validation		Triage with Baselines	Solution RunBook	Triage and Diagnosis	Triage Skills
	Rapid Deployment	Agent Promotion		Reporting	Pre-Production Review		
	Agent Deployment Cookbook	Agent Customization		Failover and Backup Strategies	Post-Production Review		
		Baselines					

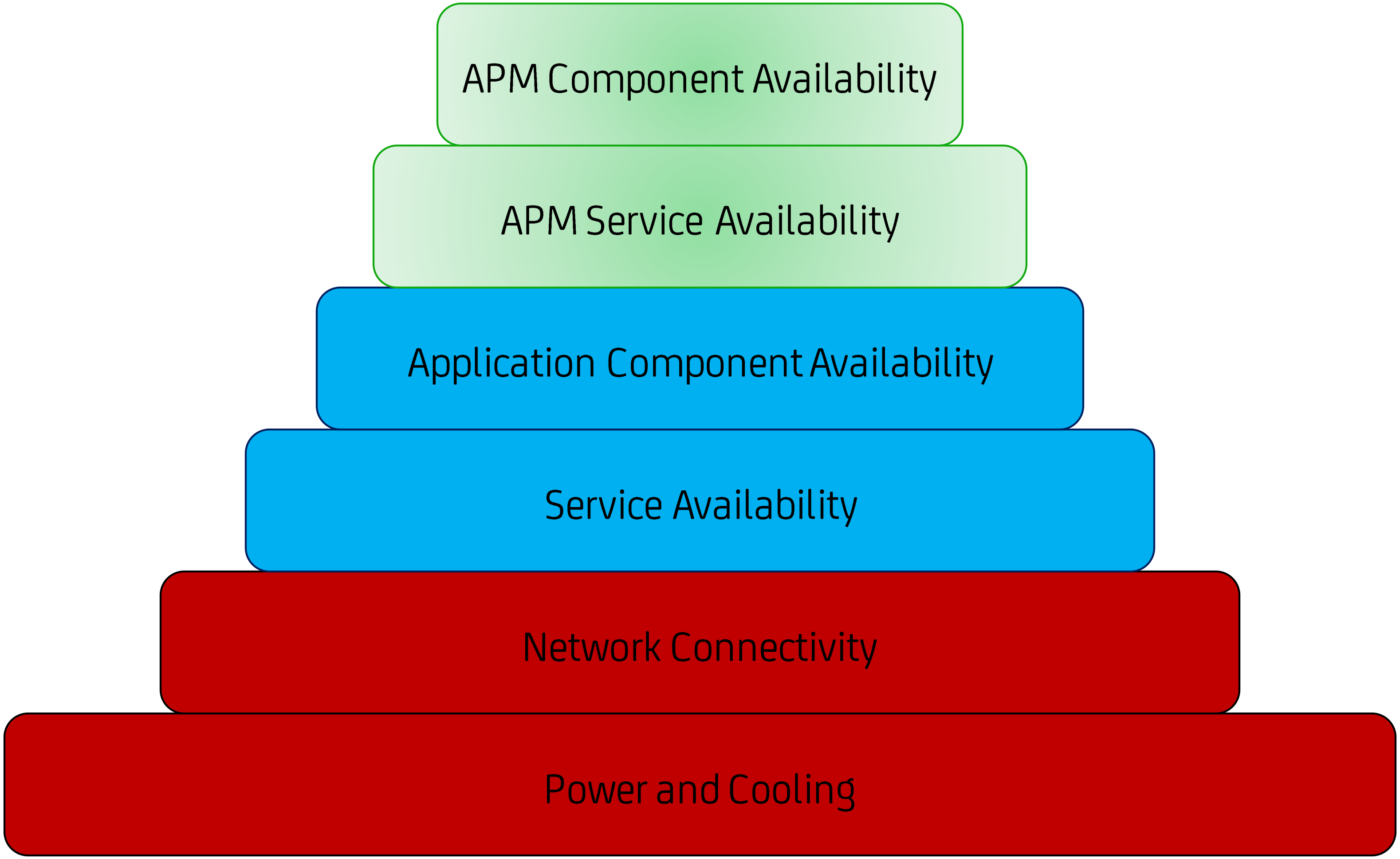
Hierarchy of APM Skills

- I can **deploy** agents rapidly
- I can **tune** agent configurations
- I can **HealthCheck** the APM environment
- I can identify applications **KPIs** and manage thresholds
- I can report **baselines**
- I can assemble and **validate dashboards**
- I can **audit** applications
- I can **plan** and manage follow-on deployments
- I can plan and **manage the APM lifecycle**
  - Technology selection, Training , Architecture, Sizing, Failover
- I can **firefight** unfamiliar applications with APM visibility



What do I need failover to achieve?

Priorities



APM Component Availability

APM Service Availability

Application Component Availability

Service Availability

Network Connectivity

Power and Cooling

Failover Gaps and Business Impact

	Component	Multi-instance	Failover Pair	Owned by IT	Business Impact
	UI	✓			Minor
Network	Electrical Power		✓ (battery, generator)		Game Over
	Service Provider				**Game Over**
	Firewall		✓	✓	**Game Over**
	Switch			✓	**Severe**
	Load Balancer			✓	**Severe**
Application	Web Server	✓		✓	Minor (performance)
	Authentication			✓	**Severe**
	Cache Server			✓	Minor (performance)
	App Server	✓		✓	Major
	Web Service				Major
	Messaging			✓	Major
	Database		✓	✓	**Severe**

Component Realities for the APM Service

Component	Failover Capability	Impact
Network	Minimal	Game Over
Platform	Physical – minimal Virtual – reasonable	Degraded Service
JVM	Minimal	Degraded Service
MOM	Manual	No Alerts No Workstation Access Data preserved
APM DB	None	No Application Map No CEM Defects
LDAP/EEM	Usually distributed	No logins
Collector	None	Historical Data lost up to 24 hours Agents migrate to next Collector
SmartStor	None	Historical Data lost up to 24 hours If corrupt – all data is lost
Agent	Available Collectors	Minimal

Acceptable Failover Intervals

Failover Task	Duration	Relative Cost
Install new server and restore backups (Physical)	8 hours	$
Restoration of lost resource from backup, with a pre-configured server (cold)	4 hours	$$
Install new server and restore backups (Virtual)	1 hour	$$
Continuous replication to a (warm) secondary server	20 minutes	$$$
Replication (for backup) and clustering	2 minutes (latency)	$$$
Total duplication of processing path	0 minutes	$$$$

What does APM offer?

APM Failover Capabilities

Failover Point	Target	Release Available
Agent + list	Next Collector	All
Agent + Cluster	MOM load balanced	7.X and greater
Collector	MOM load balanced	7.X and greater
MOM	MOM manual	7.X and greater
MOM	APM DB	-traditional replication-
MOM	MOM shared FS	9.X and greater
MOM	MOM lock file, shared	9.5 and greater
MOM	Something much better	Someday

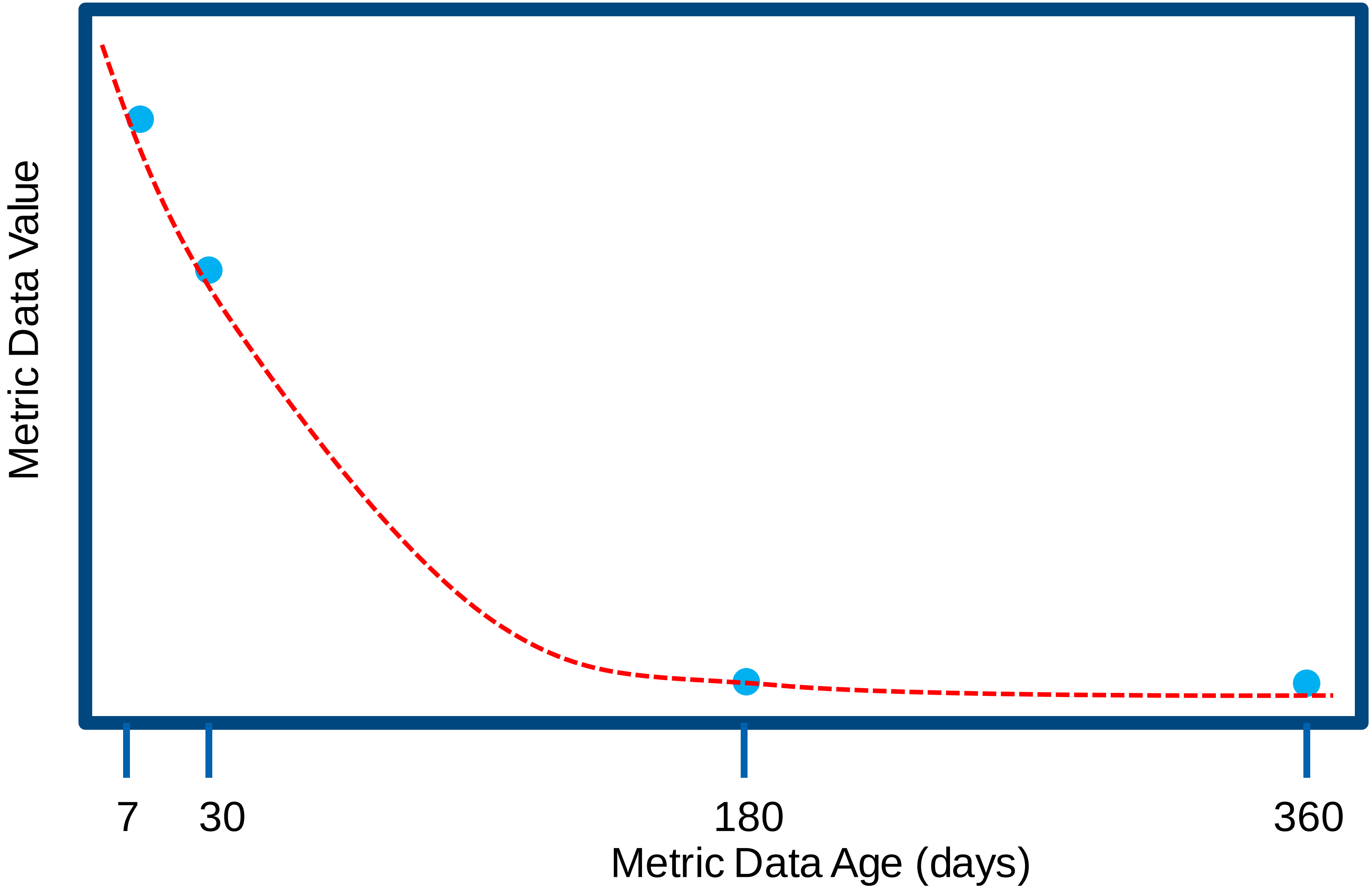
Things to think about

- First step in any MOM/collector failure is to try to fix that EM
  - Agents automatically switch to back up collector, but automating a Mom fail-over may cause issues; manual procedures provide a decision point; more intelligent control
- Stage Introscope license files for Backup EMs
- Backup Mom(s)
  - Creating new Mom instance from scratch is very fast and SmartStor data is tiny, so copying is fast
  - Must have access to outside resources

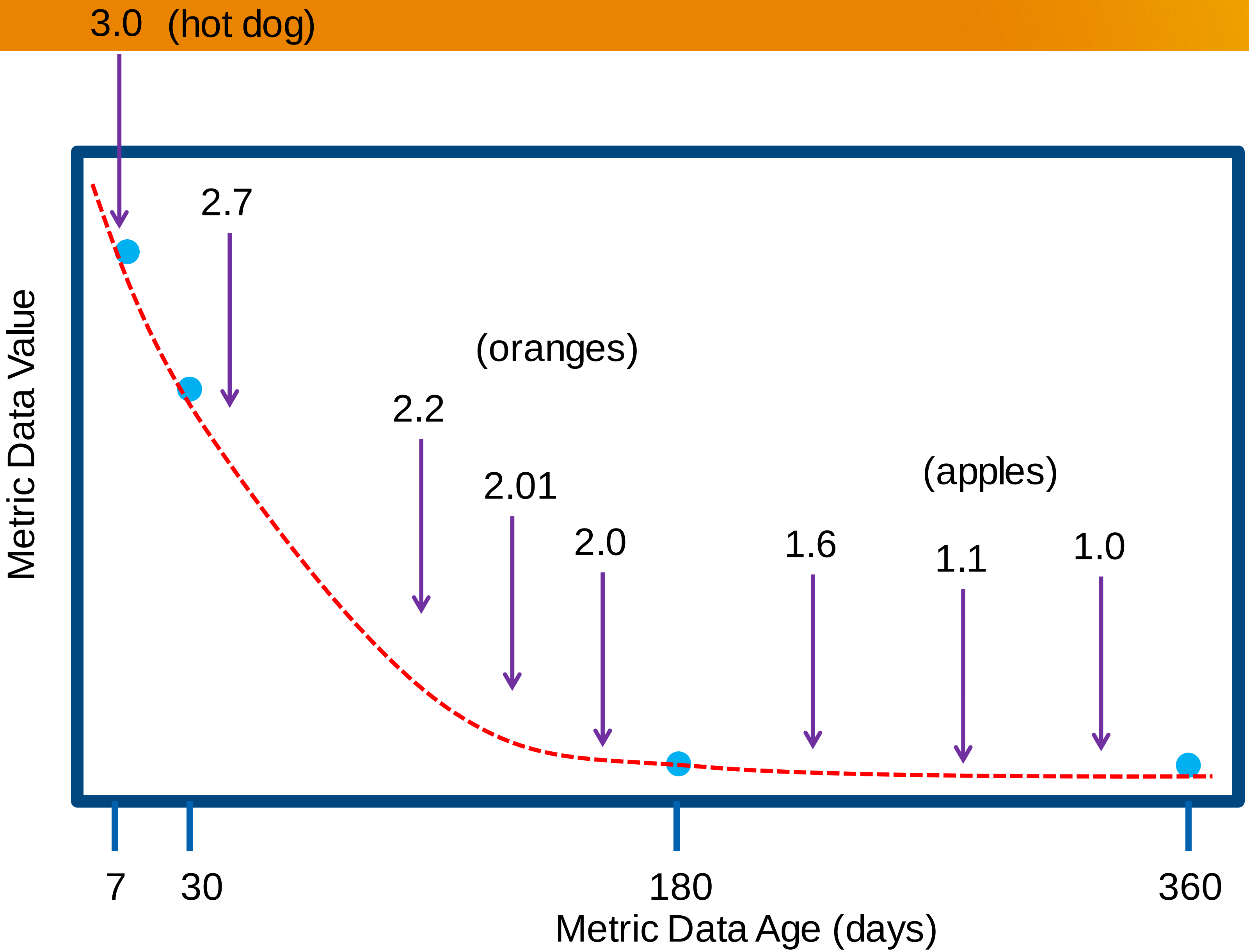
    Outside resource must recognize/accept backup Mom(s)
    - Frameworks (Tivoli, OVO, …), LDAP, SNMP, SMTP
  - Firewall rules must allow network access; outside resources & collectors
  - Collectors can be in different physical location, be careful of the Mom performance issues
- Backup Collectors
  - Agents must have firewall access to backup collectors
- You cannot combine SmartStor data from multiple collectors
  - But you could have collectors that only house SmartStor
    - with no new data this SmartStor will eventually shrink down to zero, as it ages out of the tier strategy
      - Tier strategy can be halted by setting long tiering duration (99999 days each)

What are the hard realities?

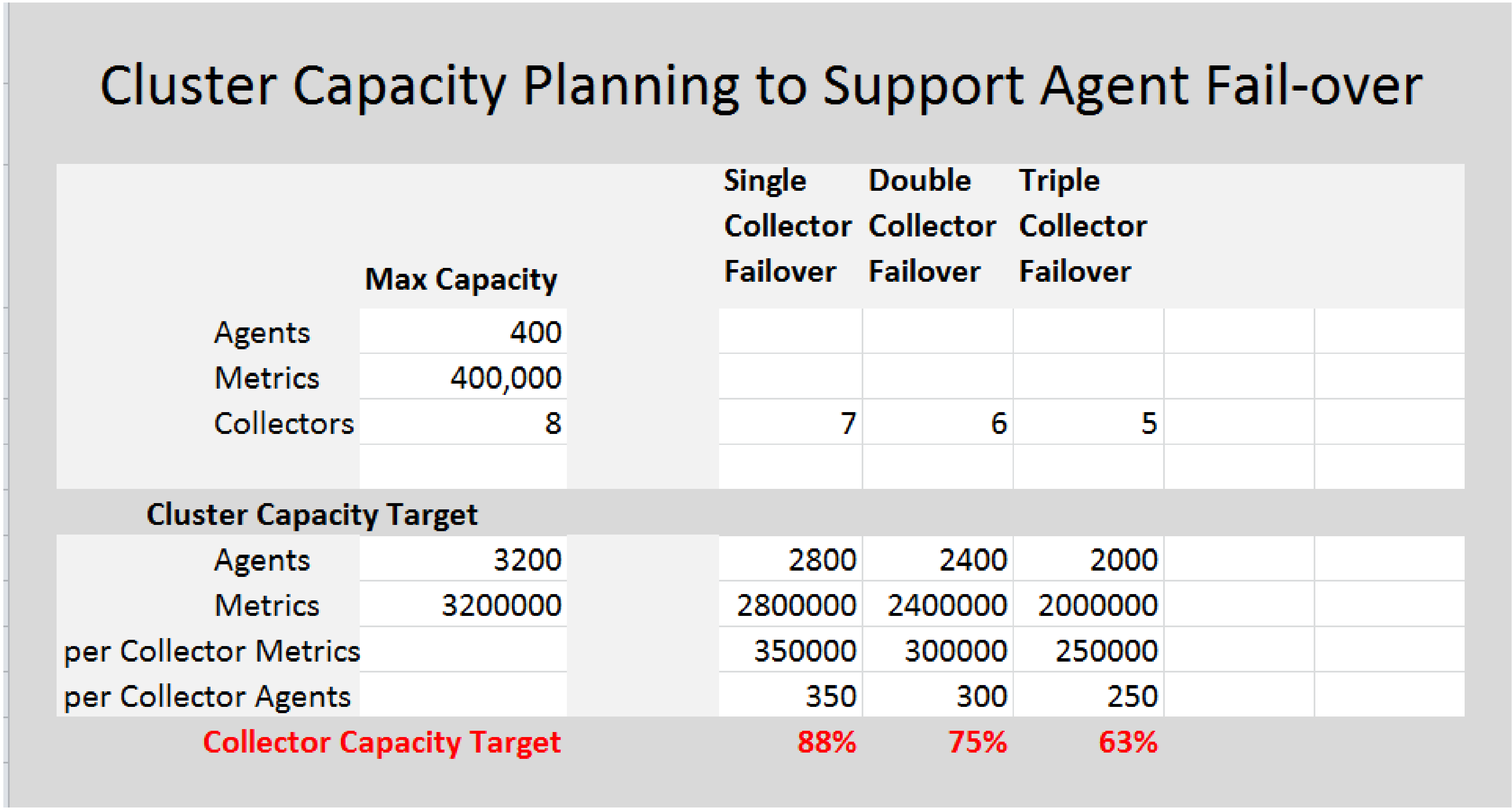
SmartStor Data Value



SmartStor Data Value – why is this so !!!



Cluster Capacity



Cluster Capacity Planning to Support Agent Fail-over

	Max Capacity	Single Collector Failover	Double Collector Failover	Triple Collector Failover
Agents	400			
Metrics	400,000			
Collectors	8	7	6	5
Cluster Capacity Target				
Agents	3200	2800	2400	2000
Metrics	3200000	2800000	2400000	2000000
per Collector Metrics		350000	300000	250000
per Collector Agents		350	300	250
Collector Capacity Target		88%	75%	63%

Why is Cluster Capacity Important?

- A Cluster operates as fast as the *slowest* Collector
- An Overloaded Collector does not fail outright – it *degrades* service in an attempt to survive
  - A degraded service will:
    - drop data <- can invalidate alerting
    - drop MOM connection <- puts strain on MOM
    - Cause agents to thrash <-puts strain on MOM and other Collectors
- When the Collector fails, to agents are quickly re-assigned, bringing the next Collector to failure in a domino affect
- If Collectors and Agents are thrashing, the MOM will degrade first Workstation Access, then Alerting, then crash or recycle.

Inconvenience

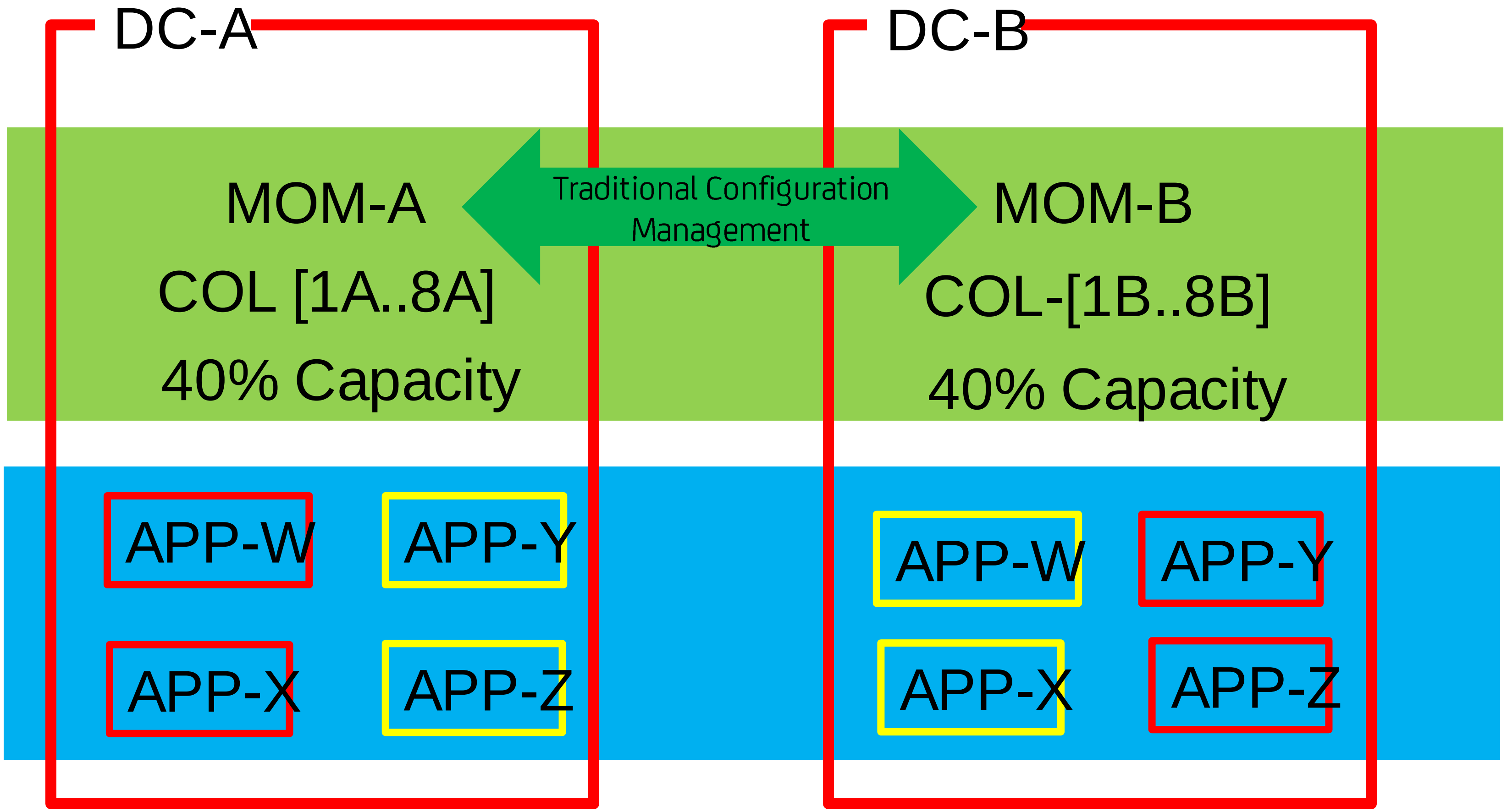
Component	Strategy	APM Impact
Agent	Built-in	None – Application instance is lost anyway
Collector	Cold Spare	Lose historical data
Collector	Full Backup	Lose up to 24 hours
APM DB	Replicate	Lose historical data, app map
MOM (spare)	Cold Standby	5-20 minutes loss of alerting, workstation connectivity
MOM (active-passive)	Lock File	5-20 minutes loss of alerting, workstation connectivity, additional failure of file system possible
MOM (active-active)	Manual	5-10 minutes loss of alerting Workstation always connect – data may be missing for certain applications for 1-5 minutes
MOM (automatic)	Doesn’t exist !!!	5-20 minutes loss of alerting, workstation connectivity, additional failure of file system possible

What are the best solutions available today?

What Works Today

- Stand-alone Collector Failover
  - Usually a Pair
    - Active-Active @ 49% load each
  - Three or More
    - Active-Active-Standby, to 100%, single Collector failure capability
- MOM Only
  - Usually a Pair
    - Active-Standby via shared lock file
- Cluster (MOM and the kids)
  - Active-Active @ 49% load each (Collectors)
  - Traditional Change Control and Replication

Active-Active



DC-A

MOM-A

COL [1A..8A]

40% Capacity

APP-W

APP-Y

APP-X

APP-Z

Traditional Configuration Management

DC-B

MOM-B

COL-[1B..8B]

40% Capacity

APP-W

APP-Y

APP-X

APP-Z

Active

Spare *Ready-To-Run but inactive*

Active-Active Summary

- Application failover is manual
- Agent re-assignment (Collector failure) is automatic
- Collectors are sized for all active applications but each data center is running half load (have the apps)
- APM Database is traditional daily replication
- MOM failover is manual
  - All MOM startup scenarios are scripted
- All MOMs have the same management modules
  - Agents and application instances remain unique
- Disaster Recovery is manual

Bringing the points to a close…

Late Show©
Top 5 Reasons… for an APM Failover Initiative

#5 – Alerts from APM have become the primary source of application status

#4 - Consumers of APM information have grown tremendously – we need to ensure ready access to the data, across the application lifecycle

#3 – We have made significant investments to build a data landfill and we are worried that data might be missing when someone actually starts to look for it – and they will be unforgiving!

#2 – APM Availability is poor – for reasons we cannot explain or prefer to ignore

#1 – APM is a tractable system on which to practice failover concepts, so that we can be selected to implement them on more critical systems

Things to think about

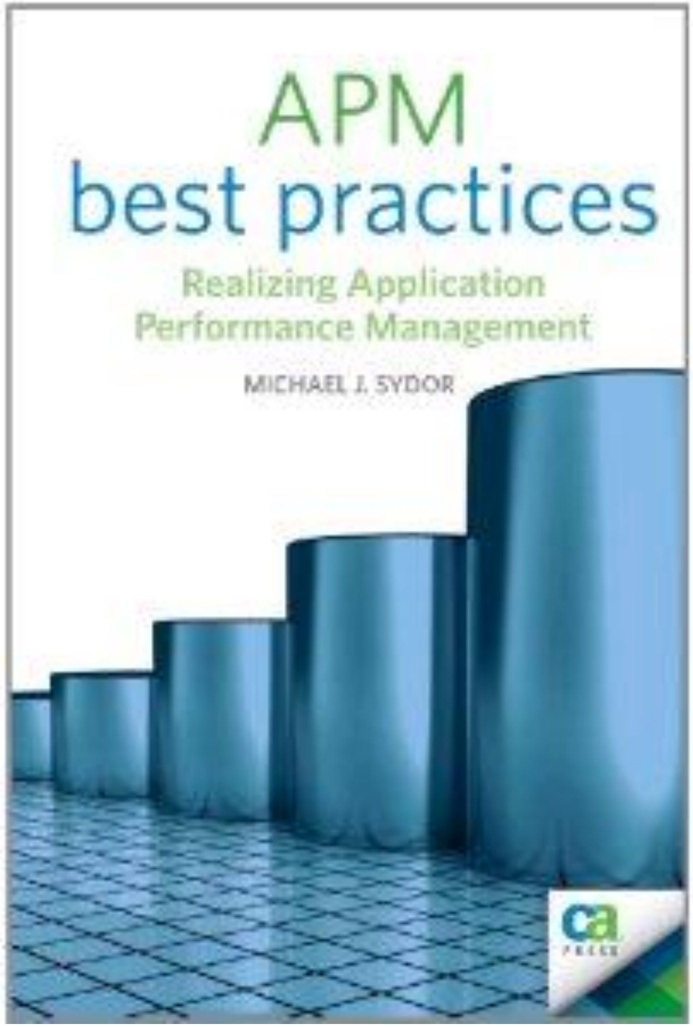
- Major guiding ideas / thoughts
  - Value of the performance data lost – Really?
    - Use it when it is fresh
  - Corporate exposure if no performance alerts triggered
    - Do I even have the right KPIs to alert on?
  - Cost and time to rebuild Introscope environment versus a permanent backup investment
    - Either in same physical location on different hardware
    - Or in different physical location (or both)
  - During major crises
    - Is Introscope expertise even available (not doing something more important)?
    - Getting production applications restored is more important than performance monitoring - ALWAYS
    - No configuration changes in production environment
      - Don’t try changes to Agent configurations
        - Test the monitoring configuration prior to production
      - But do allow the thresholds to be adjusted to improve alerting accuracy

Reality Check

- Losing alerts is more significant than losing performance data
  - Provided that you are regularly digesting the data in the form of summary reports/HealthChecks
    - HealthChecks or Baselines from the basis of effective triage
    - Real-time data makes it even better.
    - Identifying KPIs, and generating Baselines eliminates the need for historical data anyway
  - You can't validate alerting if you don't know how to identify and manage KPIs
- Restoring real-time visibility into key systems is the typical client priority
  - Get agents deployed quickly on the new instances
  - Restoring historical data is a distant second

Questions

You can do it yourself.



Available today on Amazon.com