
Value-based case study #9

International services provider cuts costs, creates new revenue streams while improving service delivery

Charles Brett, C3B Consulting

Management introduction

Imagine you are an IT service provider to a financial institution like a brokerage. It is 10 minutes to 4PM (when the markets close) and a \$40M transaction has gone missing. If the trade is not completed, the brokerage will not only be liable for regulatory fines but also for interest foregone and, potentially, any price change that occurs the next day. It will hold its service provider liable if it proves to be an IT issue that has caused the problem.

This is a scenario that is all too common. The financial implications are unpleasant. The customer relationship implications are, if anything, worse.

This case study — which reaches across industries like aviation and finance in which sophisticated managed service providers operate — shows how it is possible to break out of the confining jail to which with much of operational IT seems to operate. The outcome of the (true) scenario above will be shared later but before that, consider what happened at a major international airline.

Traditional systems management can be insufficient

This airline; like many others, contracted out much of its IT operation to an international managed services provider. Amongst the 'assets' that became the responsibility of this service provider was an extensive network of IBM's WebSphere MQ (WMQ) running on anything from zOS mainframes to HP's nonStop servers to many flavors and instances of UNIX, Linux and Windows. All these WMQ instances provided mission critical support to, for example:

- reservations, to communicate rate and fare changes to travel agents
- aircraft operations, in the calculation of load factors (for this applications adjust almost in real time as passenger numbers, weather conditions, cargo and fuel loads all continue to change up right up to departure).

In these two situations availability and reliability are critical both to commercial operations (sales, via the travel agents) and to safety for the airline. The service provider knew this when it accepted contract responsibility.

In the past the airline had introduced BMC's Patrol for systems and network management. This had worked reasonably well on its larger systems. But, for management of many of the distributed ones, it was inconsistently applied, not applied or not available. Indeed, some of the WMQ server platforms were being monitored by non-Patrol solutions, or not being monitored at all.

There was, therefore, a broad mis-match, as far as the service provider was concerned, about how it was to be able to meet its service obligations. In addition, there were serious reservations about aspects of the way Patrol worked — for Patrol requires the installation and use of software agents on the platforms it manages. For the service provider using such agents possessed four significant downsides:

- rolling out agents takes time and skilled effort for their successful installation
- software agents necessarily alter each system on which they run, with attendant risks
- upgrading or changing systems where such agents are installed is cumbersome but, worse, using agents mandates extensive retesting to ensure that all is working as expected
- there were thousands of WMQ end points but each Patrol management instance could only handle 250 nodes at that time (a legacy from fewer, larger systems days).

The service provider decided that this situation was too expensive to continue if it was to fulfill its obligations to the airline. It decided to investigate alternatives, and not just related to WMQ but also for Tomcat's ActiveMQ (JMS messaging) which was increasingly being adopted.

Making a choice

The systems provider examined multiple alternatives, including IBM's Tivoli, MO71 (an Explorer-like WMQ SupportPac) and even Nastel. None really satisfied the requirements in the way the service provider felt it needed, especially:

- the scalability to handle thousands, not hundreds, of WMQ and Active MQ nodes
- operational flexibility
- no agents
- the combination of maximizing automation where possible but also improving operational collaboration where automation would not solve everything.

The service provider's objective was, therefore, to reduce costs — in two forms: reducing daily operating expenses while also minimizing the investment required when making changes.

As part of the investigation of alternatives the services provider talked with one of the large airline reservation companies. The latter had faced a similar situation and in its own researches it had come across a product called Infrared 360 (IR360) — developed by Avada Software. From the reservation company the service provider learned that IR360 was apparently capable of high degrees of automation for managing WMQ resources. This included being able to access an WMQ manager even if it had failed — using the IR360's SOAP listener module (which enables remote

restarting, reinstalling, definition and log downloading — all occurring with security preserved).

The service provider then undertook its own research into IR360. The first obvious attraction was that IR360 is agentless. It could also be installed in parallel. One of the issues with adopting Tivoli, for instance, was it would mean de-installing all the Patrol agents, then installing Tivoli agents (for these could not be run in parallel) — and then retesting everything again. In contrast, IR360 could be installed, run in parallel with Patrol and its agents and, only if IR360 was proven, would Patrol then be removed.

A second immediate attraction, again because IR360 is agentless, was that the roll out to all platforms — from zOS to Linux or Windows — was uniform. There would be no discrepancies or differences in WMQ management tools or approach. In part this was delivered because IR360 is a J2EE application (Figure 1) acting as a portal which installs and runs on any App. Server (WebSphere, Apache, JBoss, etc.) as well as work with most databases (DB2, Oracle, SQL Server, Informix, etc.). This portal approach would then link to (and from) all the instances of WMQ. Nothing would need to be deployed out into the field instances of WMQ. Furthermore, updating or upgrading IR360 itself would be as simple as dropping a new WAR file onto the chosen IR360 application server: there would be none of the complexity and cost associated with Patrol and the previous WMQ management approach.

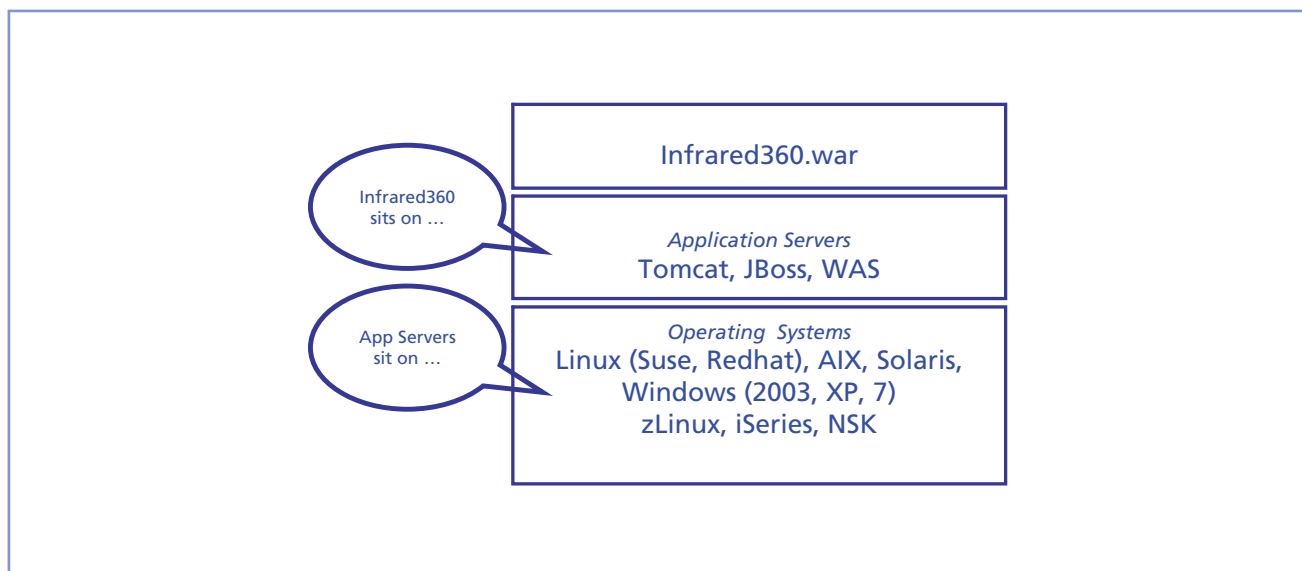


Figure 1: WAR files runs on an App. Server

A further attraction of this approach was that it could run on and exploit existing App. Servers in the airline. This meant that scalability (and other positives like failover and disaster recovery) could be 'inherited' from the airline's already robust App. Server environment without additional work.

The next attraction was on the manager side: everything is managed through a browser (IE, Mozilla). The practical implications of this were considered far reaching because it would:

- remove the need for fat clients to be deployed on designated management workstations as well as the need for specific security to be applied to these designated management workstations
- extend the principle of management consistency across the whole of the WMQ 'estate' (which had not been possible before)
- mean that access to the WMQ environment would be available from anywhere specialists were located or where they needed to be
- improve flexibility both for employees and their tasks.

For this services provider these attractions were compelling. It had signed the management services deal with the airline. Any costs — like those described above — that could be taken out (of the cost of providing the services to the airline) would drop straight through to the service provider's bottom line. These were sufficient reasons to decide to introduce IR360 (and practice has since proven these to be accurate).

Practice proves to be beyond effective

IR360 installed smoothly and in parallel, as planned. It started to produce the anticipated benefits. At the same time, as its capabilities were better understood and as practice made the services provider more effective in its use, further abilities emerged that had not been sought or expected. These 'extras' included:

- improved management of alerts
- new service and revenue opportunities
- better QA and SLA delivery
- secure delegation of user authorities
- automated problem fixing.

Take, for example, the first of these — management

of alerts. Alerts are the bedrock of systems management. When something goes wrong or a threshold is crossed (say that a WMQ queue depth had exceeded a given percentage, an alert is issued to (say) Patrol, Unicenter or a Tivoli Console — so that someone investigates and remedies the outstanding issue. In principle this works well. It does, however, possess a downside. Because the alert can go a large number of people it can cause one of two effects:

- the first is that nobody does anything (thinking that someone else is dealing with it)
- the second is where too many people become involved — and too many are 'contributing' to the resolution (in the experience of the service provider this was extremely wasteful of precious skills and resources).

Using IR360 the service provider discovered it could generate cascading alerts that would only go to the people that mattered. For example, if the WMQ depth exceeded 20% and appeared unthreatening, then this would be dealt with at a routine low level of responsibility. If, however, the queue depth continued to rise to (say) over 30%, then the new alert could be escalated to management within the service provider with the authority to ensure resolution. If it then exceeded (say) 50%, a broader warning might be sent out, including to the airline so that the latter might know in advance that there might be a problem. The net effect of exploiting this approach was that SLA observance improved, as well as relations with the airline: the latter could prepare rather than react to a problem.

Unexpectedly, IR360 threw up the possibility of new revenues for the services provider. For example, in non-production areas — like development, testing and QA — there was a constant need for clearing out of queues and resetting queue managers as part of daily activity. Previously there had been no automated tools to accomplish this. Using IR360 the service provider created a new income stream by selling access to the control service to do what had not been possible before. Interestingly this was delivered from the same portal that ran the WMQ production systems, because IR360 possesses the security segmentation to enable the various different environments (development, testing, QA and production) to coexist but still remain distinctly and securely separate.

A further benefit came with test harnesses and message generation (IR360 is script-less). These can be

prepared in advance, for example for initiating a new WMQ instance with the correct queue manager and queue values. These same harnesses/message formats and content can, however, be re-used. The services provider is now using this capability along with the IR360 scheduler to do regular automated re-certification. For example, at the end of the month at the airline there was a procedure to empty queues and restart them. Before, this was at best semi-automated. Now it is fully automated.

It goes further, however, by including a retest of the communications and infrastructure as well as bringing back up the 'refreshed' WMQ resources — all without user intervention. This has not only become part of ongoing QA; it now provides data points and reporting to the airline about how SLA commitments are being met. Indeed, so extensive has this function become it is now run on a backup portal (but still using the common database) so as not to impact the operational portal plus the airline has access so as to be able to see in real time what is happening within its own WMQ estate (thereby removing another area of people involvement).

Related to this was the issue of 'secure virtual delegated authority' (Figure 2) and the opportunity to improve collaboration across the service provider's own business. Most IT shops have multiple logical separations of activities (the most obvious are those related to development, testing, QA, operations, etc.). These tend to be distinct and to exist as their

own microcosms, with minimal 'cross-border' communications between the people responsible in each. This service provider discovered that IR360's security approach facilitates collaboration between logical groups, yet without prejudicing security. The result has caused barriers to reduce. For example, at the airline this has even been added to production environments themselves where certain applications — such as Cargo, Crew, Reservations, Fares, etc. — involve distinct separation of duties but also require that both support and business units be able to collaborate in order to resolve problems.

Perhaps, however, the most impressive extension for the airline is the automated fixing of problems. The service provider now uses 'compound alerts' to anticipate and resolve issues. For example, a queue may need to be restarted; it may not initially be apparent that the source application or service at the far end has also failed and needs restarting. The traditional solution sequence was to solve each issue as the operator encountered it. Using IR360 the service provider now defines in advance 'compound alerts' which can then complete the whole correctly sequenced restart process (queue manager, channel, queue, application, etc.). Using the services interface within IR360, there is now no user intervention if the criteria for a given compound alert are satisfied. The mean time to fix failures:

- has dropped from a minimum of 2 hours to minutes (or less)

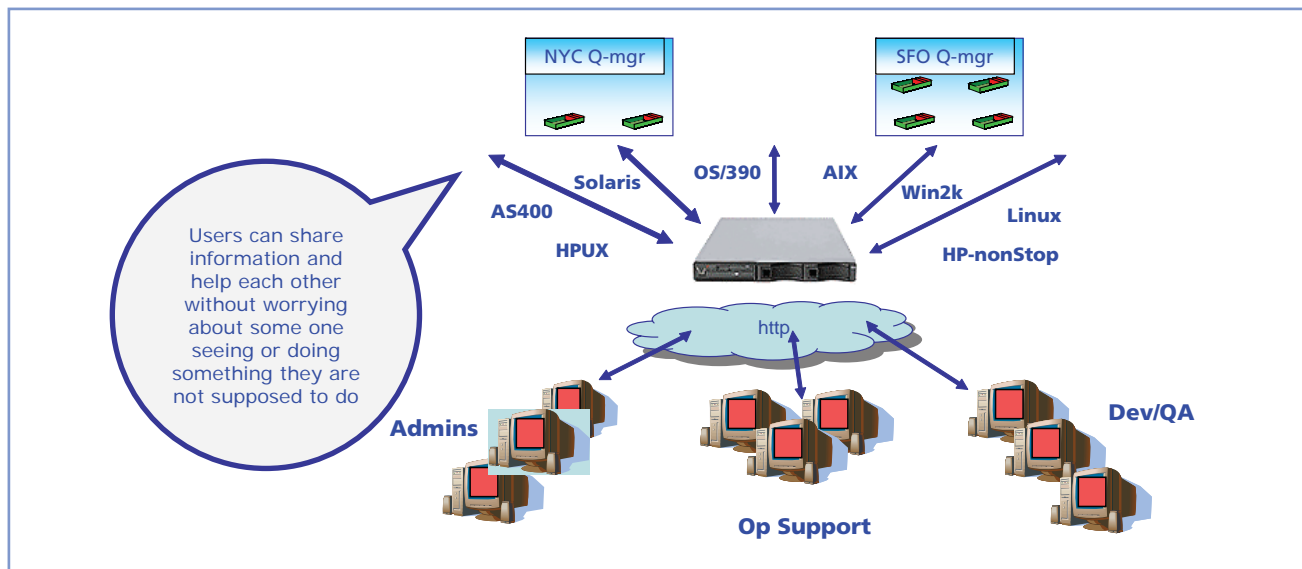


Figure 2: Collaboration which preserves security

-
- is accurate
 - generates a notification of what has happened — for the reporting described above.

This continues to save a fortune in time. It also continues to improve the service to the airline.

Business impact

For this service provider the business impact of IR360 has been extensive, and goes well beyond the airline example above. The capabilities of IR360 have started to generate wholly new opportunities and revenue streams.

Sales teams are, today, able to be much more responsive to potential new customers. In the past the service provider would take weeks if not months of learning and preparing before it could start to introduce its management and techniques, and before it would accept responsibility. This delay is no longer necessary. Because IR360 is non-invasive, it can run in parallel. The effect, in the sales instance, is that the service provider can start to work earlier at prospects and in so doing convince potential customers that it can deliver. This has proven to be a powerful sales tool (and another unanticipated benefit).

While the airline support was focused on WMQ as its principle objective, this service provider is now looking to extend the same IR-360-based conceptual approach to App. Servers, whether WebSphere, Oracle, Tomcat, JBoss or others. There is no reason not to do this, and the limits may not stop there. Customers are already asking about the relevance of IR360 to managing TIBCO infrastructure and even PeopleSoft (and similar) applications.

Back to the missing \$40M transaction

Finding a missing transaction with only minutes to go before the market closes is not simple in today's complex financial systems. If the transaction is not where it is expected, then it could be anywhere (just like your car keys, except that transactions are intangible and exist within an equally intangible infrastructure).

One attribute of IR360 is that it can search data within a message (beyond just the transaction ID, if

this is even known) — for example looking for the name of the brokerage's customer, the destination account or even the transaction amount. Using this capability a search for the missing transaction was started. Within seconds it was found, its header was modified, it was resubmitted and it was processed before the market closed.

Where was that transaction? It was in the development environment, sitting on a development queue. Why? Someone had forgotten to change queue manager designations (from Dev. to Production) and so the \$40M trade had gone to the wrong place. It happens. It happens all the time, if not so often for a combination of such a large amount just before the markets close. There was extreme relief all round.

Management conclusion

The airline that this service provider supports has obtained substantial operational benefits through the introduction of IR360 — from improved reliability through to improved SLA observance and vastly better insights into what is happening before it happens. For the service provider, deployment has not only enabled it to meet its obligations but it has been able to go further — reducing costs, opening up new revenue streams as well as embracing capabilities that can attract new customers.

The importance of the agentless approach continues to prove itself. Not only is it non-invasive (which matters to most large IT organizations that fear to 'alter' anything without extensive retesting) but it is flexible and can run in parallel. With an application that runs on common App. Servers, this service provider has simplified its operations while improving its quality. Its business model, and its customers (like the airline), now depend on IR360. To remove it has apparently become unthinkable (and could only be for something that was significantly better again).

Charles Brett
C3B Consulting
www.c3bcconsulting.com