

Won't persistent, six-sigma transactional architecture increase database storage requirements?

Yes. But fundamentally, space should be treated as a non-issue in the database world. Ideally, infrastructure architecture would be tailored to the database requirements. Tables would be indexed and paged across specific disks for each database, etc. But the database architecture plays nicely with the current MSI san architecture as well. Just to be transparent, the storage requirements would need to handle the engine destination tables, the lag tables, and the active tables. In some cases, this could make the dataset up to 3 times the size during load times. Only 2 of the 3 datasets would require index space, though.

Why would MSI allocate resources for this when so many other projects have priority?

It is actually already done. The architecture was put into place as UDM was beginning to be wrapped up and Member Month became formalized. It will be ready to go live into QA on April 30th, 2014.

What if Mercury cannot be accommodated with the current system infrastructure or due to MSI politics?

No problem! The good news is that Mercury is intended to be easily converted into MSIView without disrupting current business processes. The design was built to accommodate fallback to architecture changes required by current business/political/infrastructure constraints. In a nutshell, this means that Mercury, by design, meets every business IT constraint currently in place (with minor modifications), though it would be highly unfavorable to marry such an ideal platform into the existing architecture since this would ultimately only continue to muddy the waters for end users. Sometimes the best way to fix a road is to build a bypass. Mercury is essentially the bypass – not affecting end users during development, but a better option once it goes live.

Let's look at the project design.