Welcome to the “Dataverse”

Technology Opinion White Paper

Image: pal2iyawit / FreeDigitalPhotos.net

Tim Mangan
TMurgent Technologies
July, 2012

Sidebar

There is a tremendous need for more independent analysis and discussion about the technologies we are faced with these days than is available. The original concept of a "White Paper" was to provide the reader with technology information without marketing blather, a concept long lost by so many vendors that produce a more technical description of their products drenched in a sauce of feature promotion.

Originally, we could count on some trade press organizations to perform and publish research comparisons of like products, but even these have become less useful because of the conflict caused by the advertising side of the same house being beholden to these same vendors.

In this, my second White Paper published through PaperShare, I am hoping to encourage others to fulfill this need. Certainly, this is not the first independent view White Paper I have written, but hopefully the environment the folks at PaperShare are creating will make these papers more widely viewed, and encourage others to follow.

As to my involvement with companies related to this paper, here is my disclaimer. I am an independent consultant not currently engaged in a paid manner with any technology company producing products. My consulting company earns Income primarily helping companies with dealing with applications, which are the primarily generator of data. I also have significant interactions with two companies, Microsoft and Citrix, due to my status as a Microsoft MVP and Citrix CTP.

Introduction

Recently there has been a lot of activity in the trade surrounding handling of "big data". In this white paper we deal with both "big data" and "small data". Small data is far more difficult to deal with than big data. Big data has a well described structure and all we need is lots of storage, IOPS, CPU, and some reasonably smart software to deal with it.

This paper will describe what makes up "Small Data", the ways in which we attempt to corral and organize it today, how the changes in computing affect handling of small data, and what we need to do about it.

I first started talking about this topic about six years ago. Back then, I would get some head shaking (in agreement), but nobody willing to step up to truly address the issue. Oh there were some vendors working on something that would tell me they are about to release the cure, however, in each case they were touching a part of the problem and not the source. So we are still in the same place today as we were five years ago, except that the ways in which we compute have changed (as predicted) and the need is more immediate. Hopefully I don't update this paper in another six years!

It's All About The Data

Ever since I left my previous working life in the Networking world a dozen years ago, I have been living in a space I often describe as being between a rock and a hard place.

"I can reasonably argue that the purpose of all of our computing is to create and manipulate data, and that applications are merely tools that we use for this purpose."

The "rock", is the Microsoft Operating Systems. Fortunately it has lots of nice crevices in which I can do some spelunking. The "hard place" is all of those applications that run on the OS. And there isn't an IT Professional that doesn't consider dealing with applications hard.

What makes dealing with applications so hard is that they use and create lots of data in completely different and unpredictable ways. Microsoft provides a tremendous level of App compatibility, allowing applications built for one version of the OS to work many releases of the OS later on. So the apps that we deal with today are built using several different models and standards for development. Add to this the problem with getting a world of developers to even follow a development standard when there is no formalized training or testing, and we get the world of apps we have today. Every application creates and deals with data in unique and wonderful ways. "The cloud" further complicates our problems with data as we also have to consider non-Microsoft applications and non-Company people that impact or interact with the data.

In 2006 I presented a paper entitled "The Data Problem: It's All About The Data" at the BriForum conference in Darmstadt, Germany. In that paper, I made the argument that data was more important than the applications. Indeed, I can reasonably argue that the purpose of all of our computing is to create and manipulate data, and that applications are merely tools that we use for this purpose. My focus on data at that time was in identifying, characterizing , and corralling data into manageable collections.

In characterizing the data, I proposed an initial set of characteristics that might prove useful for automated tools to help us deal with the data efficiently. Six years later, that initial set of characteristics still looks somewhat reasonable, although the organization might not. At that time my focus was still on "Application Related Data" (ARD), and how applications generate data, so the organization was based around the perceived need to organize data based on how we might want to create collections based on how tools would want to

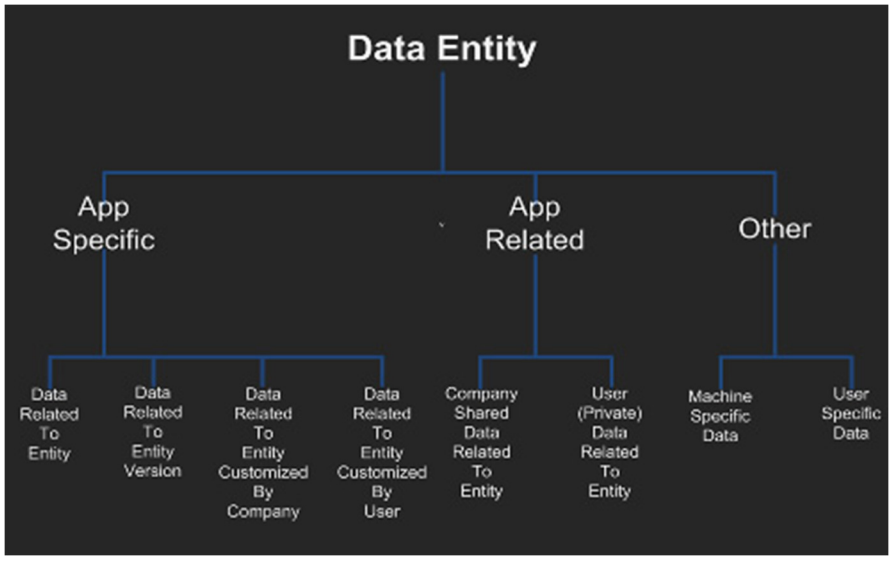


Figure 1 - Data Categorization from 2006 Paper

treat portions of the ARD under certain circumstances. For example, if an application is updated with a newer version, what data should be retained? Or if the user leaves a company, which data belonged to him or her (and should be destroyed) and which belongs to the company (and should be retained)?

At that time, I proposed that it would be best if we always tagged data with Metadata with some sort of characterization set similar to this, but acknowledged that this was unlikely to occur. Instead, I suggested that perhaps we could corral the data using a combination of existing known file system folder locations, registry key locations, and application specific mappings created through a community effort. This was where the head nodding really kicked in, but while the heads were nodding yes, the thought process was more on the order of "It would be great if someone else did that for me, I hope you find some people to help you with that task".

What Has Changed in Six Years?

Rather than working on the problem for the last six years, we have managed to make it worse, in several ways.

First, the separations that used to exist between work and personal computing and data have become much more blurred than in the past. We carry laptops, tablets, and phones wherever we go that might be owned by the company or might be our own personal device, and contain an unorganized mixture of data. In a blog article last year I referred to the typical data organizational of a PC as being "The Blob". If you had to extract out only the company or personal data on the typical storage device of a typical portable device it would be nearly impossible to do so accurately. User Environment Management (UEM) vendors have created products that attempt to redirect much of this data. If you listen to the marketing pitches, these products create an empty external layer that allows software applications to place the data wherever they want, but it really gets magically placed into this external layer based on some smart policies. These products are still quite young and not widely enough used, but perhaps they represent the "someone else to do that for me" that folks seemed to want.

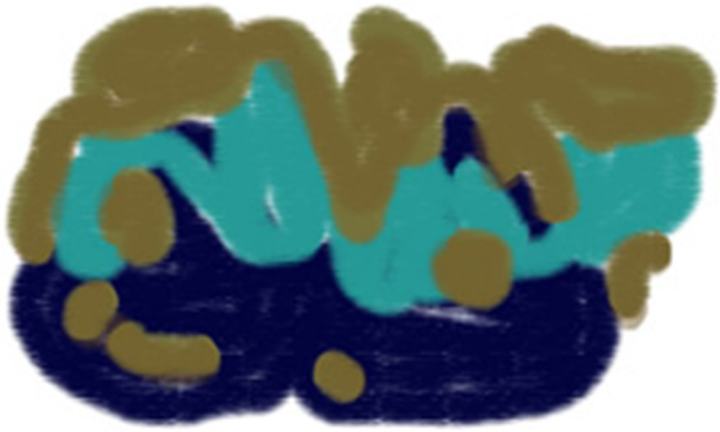


Figure 2 - "The Blob" Model

However, I find that these UEM products really just separate "some potentially useful data" from "absolute junk" and the underlying system. It is great to separate out this data from the junk, but breaking one blob into two blobs does not solve the problem. There also remain issues with incorrect placement (which will improve with time), and concerns over the ability to truly replace the lower layer without any impact to the data layer.

The second way in which we have made the problem worse is through the use of cloud based application services, especially the consumer oriented ones. The data is not only still unorganized, but we place it into the hands of others (whom, rightly or wrongly, we think we can trust) and allow them fuzzily granted rights to do things with our data. For example, if you downloaded this white paper from the PaperShare cloud service, you might have allowed PaperShare to connect to your Facebook, Twitter, or LinkedIn accounts. You probably do not have any clue as to what information *could* be exchanged between these cloud services, in what direction, and if the linkage created the ability for two of those cloud services to connect and exchange data through this service. I'm pretty sure that PaperShare is trustworthy and nothing untoward is going on, but the reality is that if I bothered to read the fine print I still wouldn't know.

The average security officer at any company is rightfully scared to death about this but, except for those involved in very high security environments (certain government agencies, financial institutions, and a few medical ones), they are completely powerless to stop it. Some in the industry have called for corporate IT departments to change their modus operandi from "no you can't" to one of "here is how to do this the right way". Their argument is that the users are going to find a way to perform work in ways they want to anyway, so rather than spend resources detecting and stopping it, companies should spend those resources making sure it is done with the least risks to the company.

The third way in which we have made the problem worse is that users no longer use a single computer at a time. I warned that this issue was approaching in the 2006 paper, and it is coming on strong without much work being done to solve it. This concurrency of computing by a user becomes a large data concurrence, or data coherence, issue. In many cases, computing environments assume that a user will be logged into one device or another, but not both at the same time. These assumptions lead to locked out access, or even worse, data loss due to unexpected "last write wins" rules for data coherence.

On Towards "The Dataverse"

Recently, I have begun to shape the problems we face with data using a different model. This model, which is data-centric, is described with new imagery, some new terminology, and of course new actions that I believe the industry should take in order to better tame the data problem.

Simple imagery is often helpful to the reader so that he or she can build a mental model to encapsulate their understanding. Today, I believe that most of us have a mental image that, while reasonable, leads us to believing that we are impotent to solve the data problem. In this image, and the one I will soon propose to replace it, we have sets of data to be managed.

> "I use the term "***dataverse***" to refer to the universe as seen by a particular data set. This includes not only the data, but rules for interaction and mappings of who can do so."

These data sets can be collections of data from files in certain locations, or some formalized container. A particular set of data can be considered more than just the data itself, but we can think of

that data set from the perspective of the data itself; I use the term "*dataverse*" to refer to the universe as seen by a particular data set. Each data set will have its own dataverse, but if we can construct a good general model of a dataverse perhaps we can build the tools we need to manage these data sets.

The image that we seem to be thinking of looks a bit like the image below. I refer to this as the "Entity Interaction Model" for the Dataverse, as the focus of attention is on entities that interact with the data set.

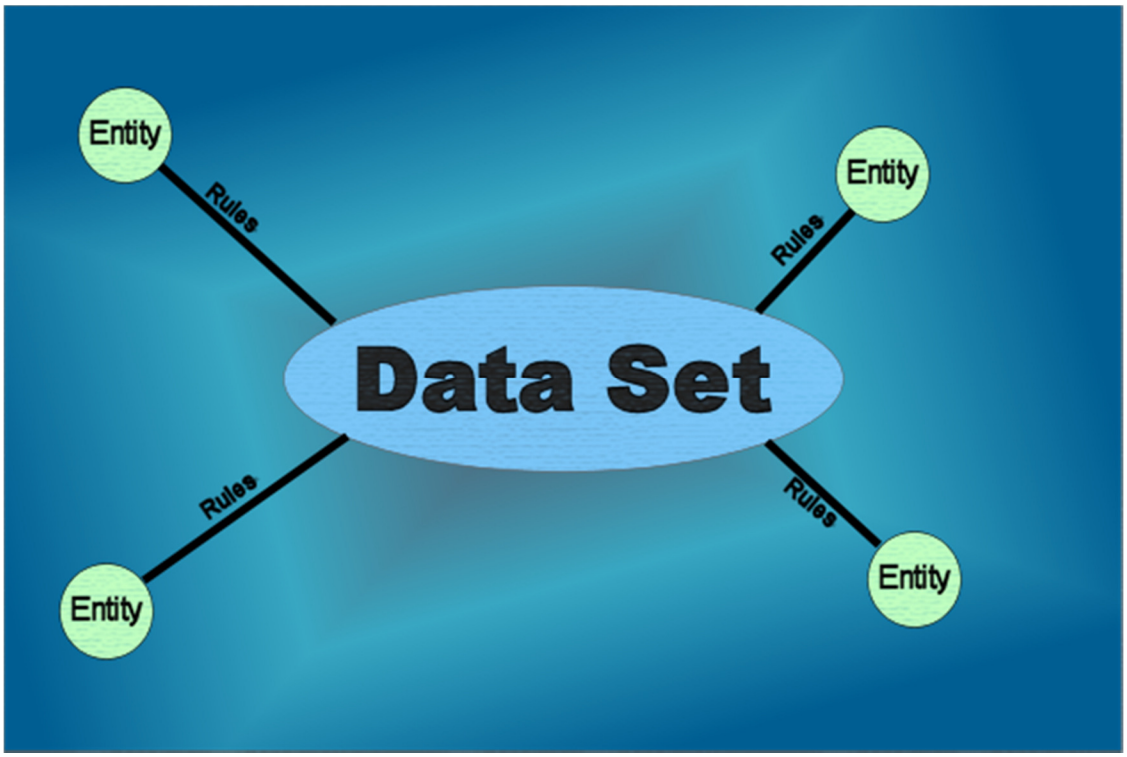


Figure 3 - Dataverse (Entity Interaction Model)

In this model, we identify entities (defined using some kind of credentials) and then apply a set of rules to the interaction between the entity and the data set. The set of rules for one entity might be different for another entity. For example, the entity that is the "owner" of this data set might have full privileges or permissions on the data within the data set, while another might have "read only" permissions to view the data.

Typically, we usually prefer to assign the rule set (permissions) to a group of entities (such as an active directory user group, or people in our "friends list"), but ultimately the rule set is applied to the individual entity within that group or list. While this is a useful image and model for the dataverse, it hides the additional complexity that we need today.

Currently, I prefer a different image and model for the dataverse. This model, which I call the Rule Interaction Model, essentially flips the attention to defining the rules first and applying them to the entities.

In this model, we create a set of well-defined rules, and the interactions are assigned to entities which dictates what they may or may not do with the data. There are several reasons I prefer this model today:

- Rules that are defined by the Cloud Service Entities today are not clearly written. They tend to be written by the lawyers at the Cloud Service to prevent you from suing them; they are impossible to understand. They are not negotiated and may be changed at the whim of the service, leaving you little recourse but to stop using the service.
- Entities also impose additional rules that allow for poorly explained "cloud to cloud" exchange that isn't handled by the original model. Undefined "Cloud to cloud" exchange holes open up when Cloud A interfaces with Cloud B which interfaces with Cloud C. Can Clouds A and C interact with your data? Who knows?

The image for this model is shown next:

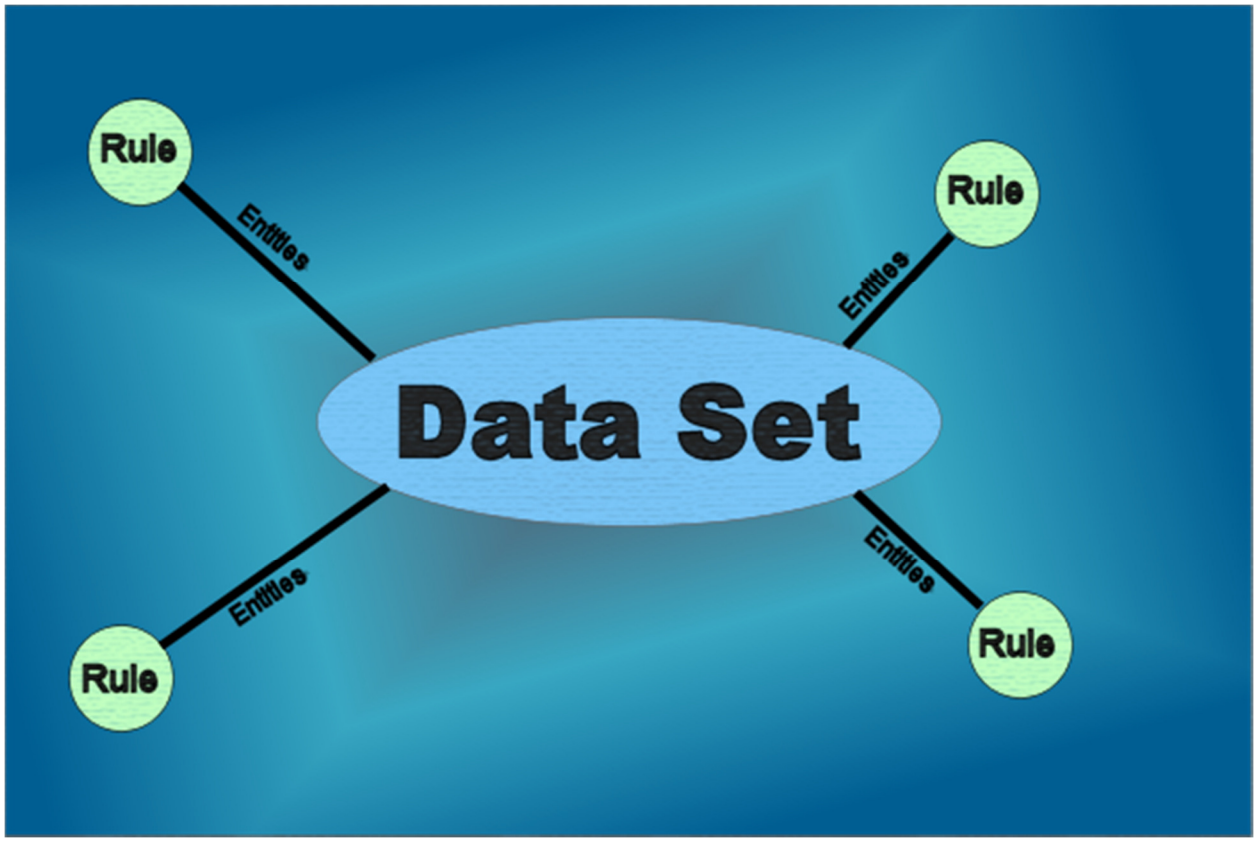


Figure 4 - Dataverse (Rule Interaction Model)

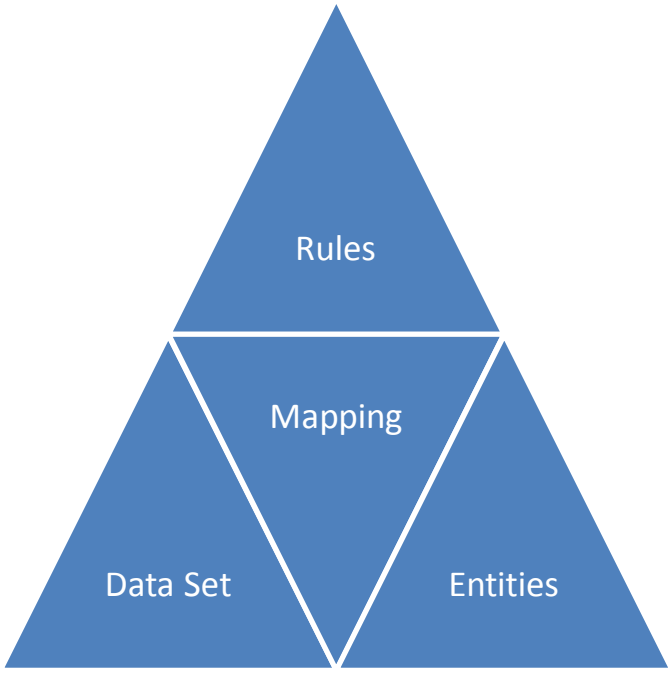
The key to making the dataverse work is extracting the definition of rules from any one single implementation. Rule definitions will need to be short, concise, and specific; and there should be only one set of rule definitions for all of the various dataverses. Each stakeholder can apply a rule or not, as needed.

However, there will be challenges in completing the dataverse definition.

Creating the Dataverse

To be useful, this model will require considerable efforts, but with luck those efforts are achievable in a reasonable time frame. In this final section of the white paper I lay out the work ahead for the community as I see it.

The definition of the Dataverse consists of three main components. Each of these components will need additional work in order to complete the definitions, but once complete we can use mapping that will enable entities to apply a rule to the data in the data set.



Part 1: The Data Set

We still have much work to do to define what the data set is. After six years, no-to-little headway has taken place in the organization of our data into data sets.

We have learned how to implement data redirection. Many fine products use redirection as a means to not only decouple data from the device, but to modify the unstructured behavior of systems and applications. To some extent, we are still learning how to manage the interaction of several redirection based technologies when used together (such as application virtualization with UEM).

The rubrics used to create these containers will need more work. The quality of these in many cases are not measured or even looked at; and what is “good enough” today will not be good enough for the future.

Ultimately, we don't want "add-on" technology to apply itself to unstructured systems in order to create the data sets. What we have today is largely the result in the incumbent vendor's shortcomings. Microsoft appears to be waking up to these issues, but rather late[1]. Implementing a native solution to data set management involves not only the OS vendor, but the independent application vendors. Microsoft to date has also shown little leadership in helping their Independent Software Vendors organize and identify the data they produce/interact with.

But it is not too late. With appropriate leadership to solve the data set identification issue for newly developed applications, we can use the retrofit approach to handle the legacy stuff.

To some extent, the cloud is currently helping us to organize our data. But I fear that this is only because the cloud is new and tends to support services on a data set. To some extent, many cloud services form a logical partition for data that forms a data set for their purposes. But we can see how quickly this breaks down with use when we look at how consumers are using cloud based storage. The user may at least start out with some location based segregation (using folders within the storage for different purposes), but even after a few months use most users quickly end up with as disorganized location based structures as they have on their PC's hard drive.

Location based data sets are one possibility, however, as I discussed six years ago the use of "self-describing data" or "tagging" might also be viable ways to define data sets.

But these are just techniques that we might use to define a data set. Work is still needed to define what the different sets should be. To some extent, the number and types of data sets will be endless, but there should be room for some standardization into typical types.

Part 2: The Entities

In my view, there are two basic kinds of entities, and each need work to complete their definitions.

First, there are people. Much work has been done on Identity and Authentication over the years and we are mostly in good shape here, in the form of registered and verified email addresses. The biggest problem I see is that individuals have multiple identities, and we (generally) cannot link those identities. An example might help here. Let's say I want to share some data with members of one group (an social/professional organization I am part of), but not with another group (people that work for a competitor of my company). If Joe has two electronic identities, one associated with his work life and another used with the other organization, sharing with Joe using the organization identity also shares with Joe the competitor. We need to be able to link these identities.

The second kind of identity is applications (or perhaps companies, such as Cloud Services). We need to develop electronic identities for these as well, in order to map to the rules. The closest that we have for this would seem to be digital signatures. These are quite different than a registered email address.

---

[1] In essence, I laid this problem to the feet of Microsoft six years ago and they still only have the same roaming profiles and folder redirection solution for released solutions. Although they appear to be starting to address this with Park City/UEV, they are not only late but appear to be interested in only putting in a minimal effort.

Whether two different kinds of digital identities will prove viable, or if we ultimately will need everyone to have a personal digital signature is to be seen.

By expanding the entities to include authenticable software applications/services, we can expand our permissions based model to the cloud of platforms, making it possible better control the cloud-to-cloud interactions that are beyond our control today.

Part 3: The Rules

We badly need consistent definitions for the rules for use in our dataverses. With different computing platforms and architectures in use, proprietary rules of one platform do not always cleanly map to those of another. The rules we right should be concise and universal. A well written set of succinct rules can easily be used by any data set to create the rights and privileges it needs.

We can define new rules that define how, where, and possibly when, the entity may cache the data, use the data for their own purposes, or share it with other entities.

An example of a well-defined rule might be that assigned entities may view, but not copy the data. Or another might be that copying is allowed as long as no modifications are made. Or copy with modifications is allowed, but that the original may not be altered.

This step is quite doable without massive software development. It would require a cross industry group of individuals to think through what is needed and reach consensus on a defined rule set, as well as establish the procedures needed to extend the rule set in the future.

I am unaware of any existing organization appropriate to house this effort, and perhaps the community can suggest one and encourage them to take on this effort, or create a special purpose organization for this purpose. My prior background in the Communications Industry leads to me to believe that such industry-wide cooperation is possible[2], but the Computer Systems Industry lacks this history completely. Perhaps the closest groups that exist fall under the umbrella of "open source" efforts, although these are primarily interested in software development frameworks.

Conclusion

Data-centric computing has the potential to replace the styles of computing that we use today with something much more powerful and flexible. While the ideal might be to start with completely new systems and data build around a data-centric architectures, it is unlikely that a fork-lift approach will work. We most likely need to address this data problem on the existing systems, and retrofit our way to this future.

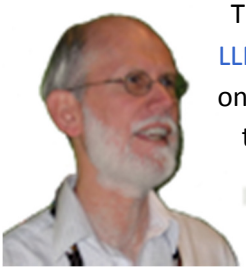
---

[2] Groups like the IETF, IEEE, and ISO are great examples of such cross industry cooperation. The Author also was Chairman of the Frame Relay Forum, a consortium of vendors and users responsible for developing interoperability specifications for their area.

We still need better ways to containerize the sets of data. Whether these are isolation containers or just metadata lists pointing to data may not be that important. Indeed, I may not really care *where* my data is if I know that I can completely control who (in the entity context) can do what with it.[3]

If this document should drive others to work on only one portion of the three components of the dataverse that I described, please let it be the rules. Even without the other work we sorely need some consistent definitions.

About the Author



Tim Mangan holds the position of "Kahuna" at his company, TMurgent Technologies, LLP based in Canton Massachusetts. He explains that at a small company, there can only be one Kahuna. He is also the President of Virtualization Boston, a User Group in the Boston area. Tim is a Microsoft MVP for App-V, a Citrix CTP (I know, redundancy). He is a sought after speaker at a number of international conferences every year, including The Experts Conference, Synergy , and BriForum. He is also the author of several books related to Microsoft App-V and System Performance. Read more at his home blog or website.

[3] I can't help but to think that there is some marketing expert somewhere now thinking "we can create a product that does this and call it "Data Virtualizations". Ugh!