AUGUST 2015

TDWI E-Book

Data Integration and Management: A New Approach for a New Age

Sponsored by:

LIAISON®

tdwi.org

tdwi

Advancing all things data.

Q&A: A DATA-CENTRIC APPROACH TO INTEGRATION AND DATA MANAGEMENT

Integration architecture is shifting from a traditional enterprise service bus architecture to a modular architecture that breaks tasks into independent processes.

Are you using the most appropriate data integration and management strategies for your enterprise? What's the role of schema-on-read and what impact does streaming data have on your environment?

For answers, we spoke to Madhukar Kumar, VP of products at Liaison Technologies, who brings over 14 years of experience developing and managing Web and SaaS solutions at the company. He holds master's degrees in software engineering and mass communications from Kansas State University, and a BA degree from Delhi University.

TDWI: What are some of the biggest mistakes an enterprise makes when choosing integration and data management solutions?

Madhukar Kumar: One of the biggest mistakes is assuming that integration and data management solutions must be separate. We've been conditioned to treat them as two different problems requiring two different vendors, but this model no longer serves the enterprise. By combining integration and data management into one solution, enterprises can solve more business problems with less overhead.

Another big mistake is not recognizing the opportunity cost of self-service integration. In self-service integration, considerable effort and resources are put into maintaining integration's basic extract, transform, and load functions. In a managed services model such as dPaaS (Data Platform as a Service), these integration functions are assumed by the provider, allowing the customer to focus on controlling the data.

A third mistake is ignoring compliance issues. As breaches become more prevalent and security requirements more stringent, enterprises should be clear not only about their security strategy but also about the security capabilities and certifications of vendors.

What is the biggest trend you see in integration architecture patterns?

The biggest trend in integration architecture is the shift from a traditional enterprise service bus (ESB) architecture, which relies heavily on network logic and processing, to a modular architecture that employs microservices to break tasks into many independent processes. Often referred to as "smart endpoints and dumb pipes," this modular pattern mimics that of the Internet, which was necessarily designed with the resiliency to withstand changing endpoints (i.e., websites) and the scalability to accommodate growth. When you consider the proliferation of applications and data sources enterprises must account for today, it makes perfect sense that the resiliency, scalability, and agility afforded by a modular architecture would be highly desirable in an integration context as well.

What is schema-on-read and why is it important in today's world of data integration?

Schema-on-read is the process of applying a schema to raw data at the time it is being pulled out of the data store. In contrast, schema-on-write, which has long been the de facto method of storing data, is the process of applying a schema to data prior to the data being written to the data store.

Although it may seem like a minor shift in approach, schema-on-read, when combined with big data technologies, offers major benefits to enterprises that must integrate ever-expanding varieties of data. One of these benefits is the elimination of the up-front need to painstakingly model tables, table relationships, data types, and so on. Instead, raw data of any format can be set directly into a schema-less repository. This saves effort on the front end of an integration project and eliminates the ongoing model maintenance that would otherwise be required as new data sources and use cases inevitably enter the picture.

Another major benefit of schema-on-read is the ability to define the data's schema—and its output format—on the fly through self-service data management tools. What's more, the raw data persists so you can do it all again tomorrow in a completely new format if needed.

What is polyglot persistence and how does it relate to schema-on-read?

Polyglot persistence is the use of different data persistence technologies to optimally store and assimilate varying types of data. For example, a big data storage repository might consist of a relational database for financial data, a document database for product information, and a key-value database for social media content. Polyglot persistence is a prerequisite for schema-on-read.

How does streaming data processing fit into data integration use cases?

In today's world of big data, more enterprises are looking to process streaming data, which involves using incoming data to trigger real-time action as well as storing it for longitudinal analysis. A use case for this can be found in your wallet. Each time you swipe your credit card, streaming data processing compares the purchase with your overall purchasing pattern to determine whether the transaction will be authorized or will be denied as potentially fraudulent.

The compute power of big data makes streaming data processing possible, but your integration environment plays a critical role, too. The data agility required to move from static point-in-time reporting to real-time insight mining requires timely interactions between integration and data management operations. These operations must be unified on a single platform to avoid offline processes that delay decision making.

What is Kappa Architecture and what are some of its advantages?

Kappa Architecture is an approach to streaming data processing that treats all incoming data as streaming data—whether it's being processed in real time or not. Every piece of data that comes in is treated as part of an event, with relevant metadata (such as sender, time stamp, transformations applied, and so on) being captured. Because the raw data—and the event data of the raw data—are carefully accounted for, the data can then be "replayed" from any angle, in any format, at any time in the future, which is a huge advantage not only for streaming data processing but point-in-time data processing as well.

Liaison offers a data-centric approach to integration. Can you explain this concept?

In a classic integration pattern such as that offered by an enterprise service bus or a hub-and-spoke model, integration is focused purely on moving data from one system to another and the data is left "as is" once it reaches its destination. In other words, the data exists to serve the applications and integration is little more than a wire connecting one application to another.

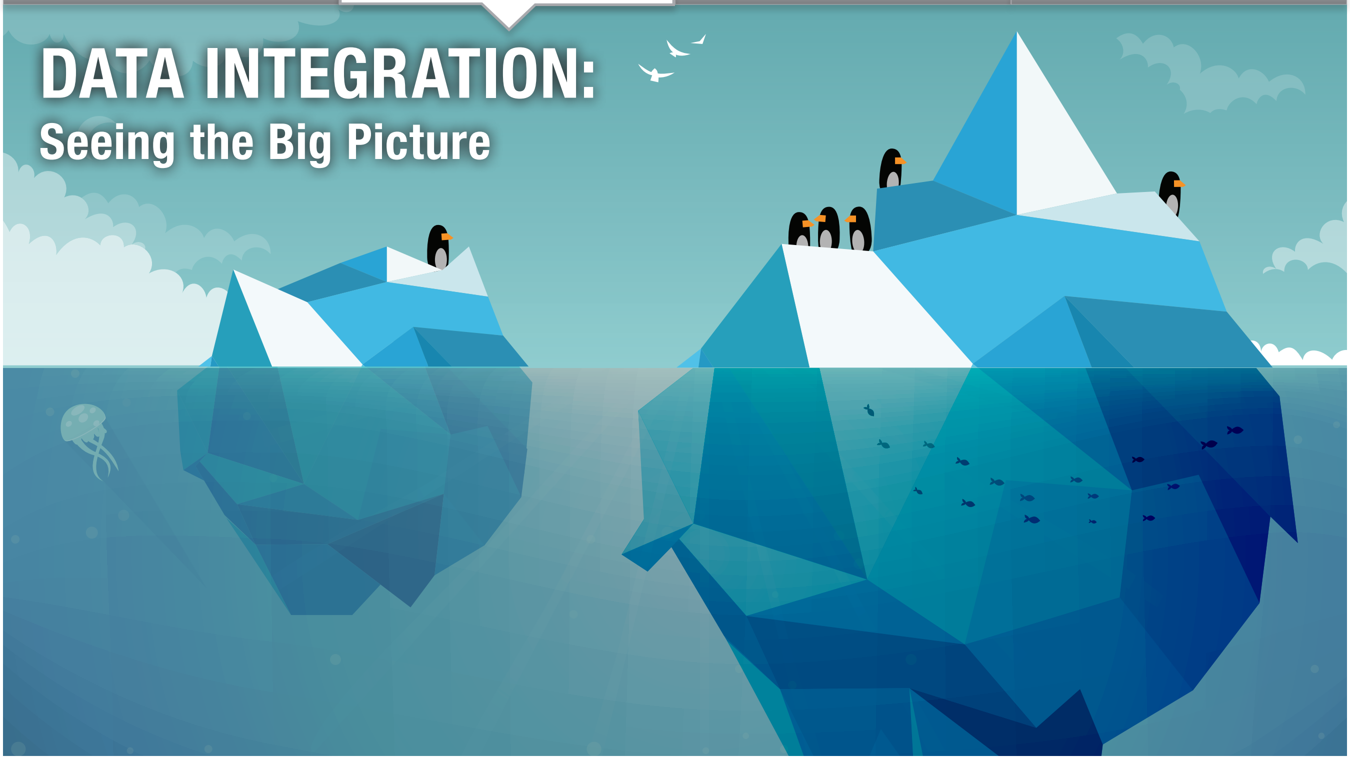
A data-centric approach to integration, on the other hand, recognizes that the data in and of itself has value, and great care is taken to persist all iterations of the data, along with the event data of the data, as it flows across the enterprise. This approach not only moves the data about as needed but also stores it for future materialization in whatever format is required by the use case.

What sets Liaison's platform and offerings apart from its competitors?

All of the above! Every trend, concept, and technology discussed thus far—microservices, self-service data management tools, polyglot persistence, schema-on-read, Kappa Architecture, data-centric integration—represents the most innovative methods available today for integration and data management. All of these are encapsulated by the Liaison ALLOY Platform, the first to unify these best-of-breed approaches in one compliant, cloud-based solution. It is also the first platform to deliver integration and data management as managed services, freeing customers from the significant burden of designing and maintaining their own solutions.

DATA INTEGRATION:

Seeing the Big Picture



When it comes to data integration, think big, as in making data integration a key part of your data management Big Picture, not something separate and distinct.

Data integration (DI) is never an end unto itself. No business integrates data simply and solely in order to *integrate* data. To do so would be madness. Integration is always adjunct to a Big Picture.

Unfortunately, this Big Picture almost always gets lost in traditional approaches to DI.

This is because DI is traditionally practiced—and DI products and services developed and marketed—as ends unto themselves. Thus the all-in-one DI platform treats data integration as primarily a function of accessing, moving, reducing, and transforming data. To the degree that DI platforms address priorities such as data consistency and cleanliness or metadata standardization, they typically do so as problems that are ancillary to the primary tasks of data access, movement, reduction, and transformation.

Think of the process by which DI players incrementally added—almost always by acquiring best-of-breed vendors—data quality, master data management (MDM), and other data management (DM) features to their core DI stacks.

Core DI, usually in the form of batch extract, transform, and load (ETL), came first. It's their bedrock.

This isn't to say that entrenched DI players don't take data quality or MDM seriously; it's rather to point out that their primary focus is on the problem of integrating data (typically in the context of a batch ETL paradigm)—and that issues of quality, consistency, or governance take a back seat to this focus. Instead of treating DI as a component of a Big Picture—of an e-commerce or multichannel retail application; of an analytic model or application in life sciences; of a loan-approval process in financial services; or *of a data platform itself*—traditional tools treat DI as if it were completely separate and distinct.

Think of it in terms of food preparation, which is almost always teleological (i.e., directed to a goal or end: the meal that one eats). There's a lot that goes into producing a good meal, especially during the prep phase, and while it's true that one might separately prepare individual items or ingredients, one rarely, if ever, undertakes food prep for its own sake. The same is true of data integration, which is just one ingredient in a project, application, or service deliverable—or, at a higher level, in an enterprise data platform. Unfortunately, the traditional paradigm treats DI like the deliverable itself.

That's a problem.

"Today, most companies purely focus on just one step [of the data integration process]. In the case of [companies such as] Informatica or a TIBCO, this is the 'integration' itself—putting [data] into a[n end-stage] repository," says Madhukar Kumar, vice president of products with Liaison Technologies, a cloud integration specialist.

"The problem with this is that when the customers go and purchase a 'solution,' they have to go and find something else or somebody else—or, what many [customers] choose to do, build their own homegrown systems—to do the other steps."

At a high level, these "other" steps involve the acquisition and conditioning of data, in addition to the requisite reductions and transformations that constitute the more familiar work of data integration, says Kumar. Just as there's no such thing as generic data integration (data is always engineered to address a specific set of business requirements), the means by which data is acquired and conditioned must likewise address different vertical- or domain-specific business needs.

Yes, it's possible to cobble together an all-encompassing "data platform" by mixing and matching products or point offerings from different vendors, and some vendors do sell "integrated" offerings that purport to mix and match DI, data quality, and MDM. What's lacking in all cases is a solution that encompasses the Big Picture: the specific requirements of an e-commerce or multi-channel retail app, a loan-approval app, or other business deliverables.

This becomes especially apparent once one gets down to the nitty-gritty, Kumar argues.

"The other big problem with this … is that with something like master data management, the models are always going to be so specific to the domain that you're going to need domain experts to create them. If I go and try to create a product master for healthcare, I need a product expert for healthcare and life sciences. They're the ones who understand the different codes—and there are thousands of different codes," he points out. "You need an expert for that, and the experts are truly the customers [i.e., the people who use the software], nobody else. However, when the MDM vendors sell their products, they don't sell MDM expertise. They say, 'Here's an out-of-the-box model for a product'; then you have to go hire somebody and take that model and customize it to meet your needs."

DI realists like to say that no prefabricated data model, no matter how "industry-specific," will ever survive contact with the physical world. This is especially true of canned data quality and MDM offerings: at a high level, across all industries, the physics of cleansing, matching, and de-duping data might be roughly similar, but for a data quality solution to be authentically "baked" into a business app, it must effectively address itself to the idiosyncratic and occasionally counterintuitive requirements of a particular industry, domain, or business.

Take master data, for example. Most MDM offerings are designed for one (or both) of two domains: product and/or customer data. What about MDM in HR—where records such as "job title," "role," and "responsibility" replace familiar product or customer records—supplier relationship management, vendor relationship management, and so on? How easily can existing product or customer data MDM offerings—with their requisite models—be customized or retrofitted to address these and other domains?

In the context of data integration, and particularly with respect to *big data integration*, a cloud architecture can be hugely advantageous.

"There are MDM solutions that are very specific to [customer, product, and other] domains, and a lot of MDM [vendors] will say, 'We're also multi-domain,' which means that they claim to do both product and customer data. But in practice, it's always much more complicated," Kumar indicates. "You always have to customize. If you're a large company, you will always have to do that. The [key considerations are] first, how easy is it to customize [the data access, conditioning, and integration components] using the solution, and second, the sophistication and flexibility of the solution [itself]."

Admittedly, Kumar isn't a disinterested observer. His company, Liaison, markets what it calls a managed Data Platform as a Service (dPaaS) that combines a design and development environment with a cloud-based data management platform. And by "data management," Kumar contends, Liaison takes a holistic approach: its Liaison ALLOY Platform encompasses data access, conditioning, and integration/final loading. "We're not experts in the model—you, the customer, are, so our approach is to give you tools that you can start with as a base—you can quickly create your own model. We're also a managed service based in the cloud, and we offer a full suite of services so that you can build out a Data Platform as a Service yourself," he points out.

"Cloud" has become something of an all-purpose *deus ex machina*—Got a thorny problem? The Cloud will fix it!—but in the context of data integration, and particularly with respect to *big data integration*, a cloud architecture can be hugely advantageous, Kumar argues.

"The traditional data integration vendors haven't truly embraced the cloud. Their biggest challenge is that they're designed primarily for [an] on-premises [world]—for relational data, for relational databases. When you're dealing with relational data, you are constrained by the limitations of that [relational] model," he says, citing the RDBMS and its predefined schema. Relational data management has relaxed to some extent (e.g., some RDBMSs now offer at least limited support for [late binding, which achieves something like schema-on-read, albeit with functional and performance limitations]), but the relational database still puts data (or the kind of "world" that can be modeled and represented by data) in a figurative straitjacket, at least compared with a schema-optional platform such as Hadoop.

"The kinds of things you can do with data [in Hadoop] are almost limitless: you can have a graph-based database, you can find out relationships, do attribution, do segmentation, out of box. These [kinds of capabilities are] not available in any of the point MDM solutions. We built out [the] MDM [component of the ALLOY managed service] using the MapR distribution of Hadoop, so not only do we have persistence, we have key value and we have time series in addition to graphing."

Kumar concedes that a managed dPaaS isn't a prerequisite for doing Big Picture data integration—it's conceivable, he allows, that a vendor could replicate something similar in an on-premises context—but argues that it's difficult or impossible to trump the flexibility, scalability, cost, and convenience that function to differentiate the cloud paradigm.

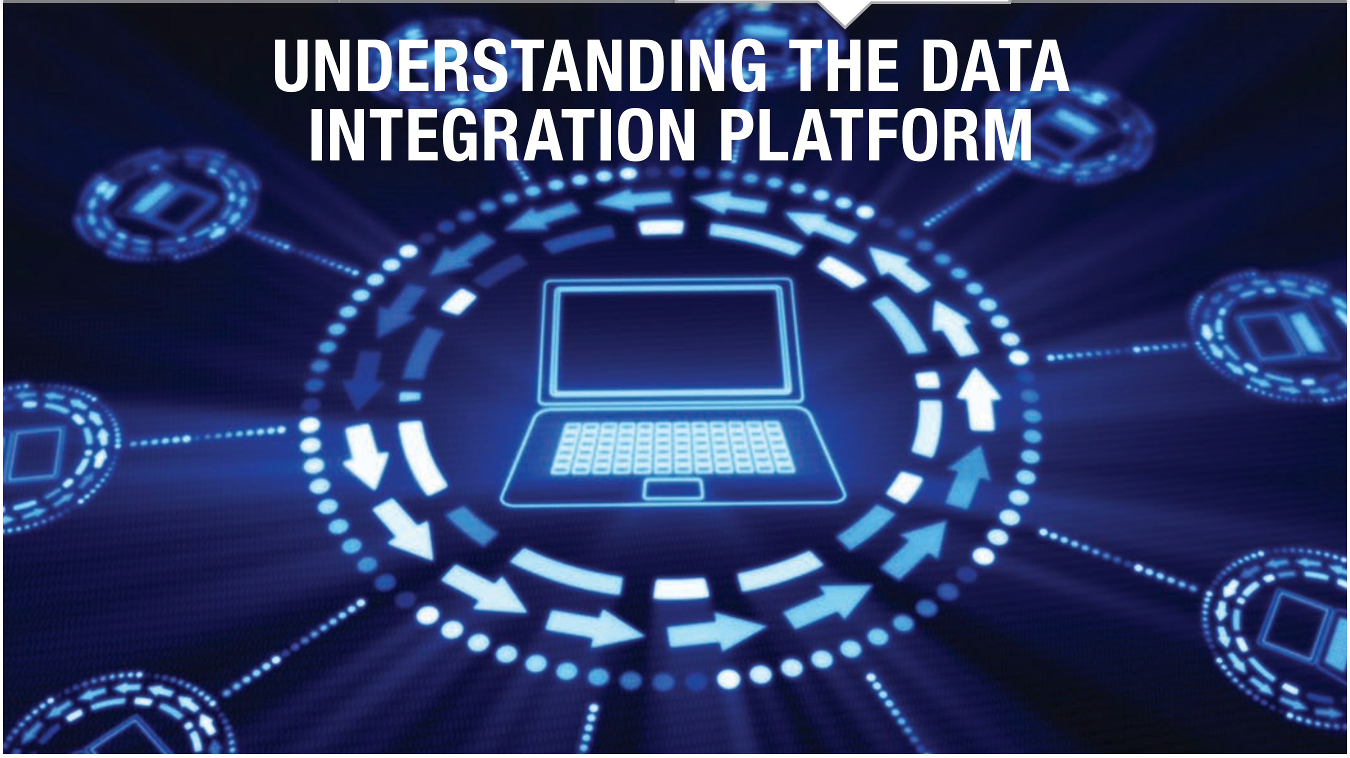
What's more, he argues, any comparable on-premises offering must be flexible enough to (cost-effectively) manage data of all shapes and sizes (i.e., strictly structured, semi-structured, and multi-structured data types) and must be as conversant with APIs (and especially with Web services APIs) as with more conventional (SQL-driven) mechanisms for data access and manipulation. This effectively rules out traditional DI platforms, along with most DI, data quality, and MDM point solutions, Kumar claims.

There's also the question of expertise. If it's true—as Kumar himself concedes—that you're always going to have to customize, to the extent that you can draw on an extensive knowledge base (with expertise that spans different verticals as well as the domains within those verticals), you're going to have to do that much less customization. This, he maintains, is Liaison's trump card.

"We have our Contivo Analyst, which uses machine learning to look at the data maps that we have created—we have more than 15,000 of them—then come back and make recommendations based on what it finds. That's our integration DNA. A conventional data integration vendor, on the other hand, is purely looking at this problem from an ETL perspective. That ETL process is inherently very different from business-to-business or application integration," he concludes.

"The integration piece is the plumbing piece. Do you really want to be squandering time and money building plumbing? We take the approach that we're doing business-to-business integration, only we encapsulate that in our managed service. If you're a customer, we're taking away your biggest pain—the plumbing."

UNDERSTANDING THE DATA INTEGRATION PLATFORM



Don't trap yourself into thinking that most data integration technologies address just one component of data preparation and engineering: namely, the data movement and transformations that comprise the bulk of ETL. Instead, consider data integration from the perspective of a data platform.

The best way to look at data integration is by looking at it from the perspective of the data platform.

Data integration (DI) needs and challenges vary widely across different industries. So, too, do government or industry regulatory requirements, which always serve to complicate DI. This is one thing traditional approaches to data integration—which focus primarily on an extract, transform, and load (ETL) paradigm—get fundamentally wrong.

They get something else wrong, too: they treat DI like the Main Thing, not like the plumbing that it is.

The truth is, most data integration technologies address just one component of data preparation and engineering: namely, the data movement and transformations that comprise the bulk of ETL. A better way of looking at DI is from the perspective of a data platform (i.e., a conceptual architecture that addresses data access, conditioning, standardization, and persistence). Traditional DI subordinates core functions such as data quality and master data management (MDM) to its own ETL-centric paradigm. (See "Data Integration: Seeing the Big Picture" in this publication.) A data platform treats data quality (DQ), master data management, and governance like the core components they are, not as supplemental to the activity of ETL.

Madhukar Kumar, vice president of products with Liaison Technologies, which markets the Liaison ALLOY Platform, a cloud-based Data Platform as a Service (dPaaS), describes what such a platform might look like. "Imagine if I drew you a picture where what we think of as data integration consists of three components. The first is the acquisition piece, but instead of hooking up to a relational database or a traditional [on-premises] data source, we're also hooking up into Salesforce, NetSuite, or maybe even Workday in the cloud. Simultaneously, we're enforcing a bunch of data quality and consistency rules on top of that [data acquisition]—cleansing and identifying incorrect data. The third [component] is we're actually putting that [cleansed and consistent] data into a repository—you take it, with the appropriate metadata, and drop it in a repository," he says.

This last item is the MDM component, says Kumar. "From MDM, it can feed into websites through an API call, where it might generate dashboards or generate a report ... or it may feed back to a data warehouse where you might be doing your other analytics," he says. "With a data platform, you consolidate all of this. Not only are you able to do [the] mastering of [the] data, but acquisition and cleansing, too. This is similar to what happens in a business-to-business integration process. Compare this to a traditional [ETL-centric] process, which is inherently very different. ETL is often a batch job, or it's as simple as implementing a Cron job that schedules your extraction, your transformation routines, and your loading. You may or may not have mapping [to a data model]."

Traditional, ETL-driven DI isn't just outmoded, Kumar insists; it's broken. It was optimized for an era of comparatively scarce data and—more important—of mostly structured data sources, such as hierarchical databases, relational OLTP platforms, or the data warehouse itself. It's ill-suited for data of different shapes and sizes (e.g., semi-structured data, such as logs or some kinds of JSON files, or multi-structured data such as text and multimedia files), and it's fatally biased in favor of stateful, direct connections between and among data sources. This makes it a poor fit for the REST-based world of the cloud.

"The vast majority of enterprises today start off at least one data integration project every year. Of these, a small fraction of projects are actually completed on time, and an even smaller percentage actually [realize] ROI on top of that," he comments. "Enterprises are now spending *four times* as much on integration because the complexity has increased. There are enterprises that have giant teams of integration zombies, because every time they have to change something, [these teams] have to go in there and do a massive heart surgery to your [data integration infrastructure.]"

The Regulatory Trap

Industry-specific needs—and, particularly, industry- or region-specific regulatory requirements—also reveal the shortcomings of the traditional, ETL-driven DI paradigm, Kumar argues. He cites two industry-specific standards: changing Payment Card Industry (or PCI) security standards and the inordinate complexity of the Health Information Portability and Accountability Act (HIPAA). "With HIPAA, not only do you need to understand the audit log of the data ([i.e.,] did somebody touch that data at any point in that entire integration process), but you ... have to maintain very strict, immutable logs of the overall state and journey of the data. What was the inception of the data? Who touched it? Did it go through some transformations? If so, [were these transformations performed by] a machine or a human being? Did that human have the rights to go and make changes to the data? What changes were made?"

In a sense, traditional, ETL-driven DI is a totalitarian paradigm: it subordinates a process flow (e.g., a payment card transaction, or [in Kumar's example] a data integration flow between and among healthcare applications) to its *own* process. It *brings the process to itself* and can't easily be brought to (or implemented in-band with) the process. Kumar contrasts this model with that of an ideal data platform—or with Liaison's ALLOY dPaaS.

"Our managed [dPaaS] has three layers: the first, a Hadoop layer, provides polyglot persistence. The second, a 'plumbing' layer, is the integration layer; the third is a visibility layer. What we've built provides full transparency of data [in the context of] two 'buckets': the first, data about data; the second, the actual data itself," he explains.

The takeaway, according to Kumar, is that a dPaaS such as ALLOY is closely aligned with what he dubs the data "journey." This journey can vary distinctly across different industries, regions, or business domains, he argues; to attempt to legislate the flow or pace of this journey—which is what traditional DI does—is madness.

"The data journey … will tell you where did [that data] originate, what was its [original] time stamp? Did it go through certain transformations? Did somebody apply rules to it? Did it change? What was changed? Who changed it? Was it cleaned? If so, what rules were applied? But this [model] also lets you do other things, like [perform] data heuristics at a batch level," he maintains.

"It isn't that batch [ETL] is a bad way to do [data integration]—it's that it isn't the *only* way, and with more demand for real time, its [usefulness is] decreasing. In our [dPaaS], when you do an actual ETL bulk transfer job, you're able to profile the data as part of that process."

No More Pipefitting

In a data platform paradigm, data integration itself is implemented as plumbing. This isn't a bad way to think about it, says Kumar, who argues that established DI players are in the business of selling customers the equivalent of *bespoke* plumbing—the pipes of which must be laboriously custom-fit for each and every implementation. There's absolutely no reason for this, he maintains.

"The biggest mistake a lot of enterprises make today ... is to try to do data integration on their own," he contends. "Bear with me; I'm not saying this because it's self-serving. If I'm Wal-Mart, my core business is to sell products, but I'm spending about $100 million on data integration software that's just fraught with complexity. It doesn't make sense. [This model] hasn't tracked with how [the] applications [we use] have [themselves] changed. What used to be a giant monolithic system like an SAP ERP has been deconstructed over time: Salesforce has taken over [the] CRM component; Hybris has taken over product information management. Every time this happens, you have more [of a] proliferation of apps—and these apps have APIs, not all of which are auto-discoverable."

He sees this as part of the bigger, root problem of broken DI—of DI that prioritizes its own process at the expense of other critical data management processes, to say nothing of the process flows between and among applications and services. "A lot of customers try to keep integration completely separate from data management. This happens whether they realize it or not. It's [a function of] how these [data integration] tools are designed to be used," he comments.

"We're working with a very large biotechnology company. They have one full team for ETL, one full team for MDM, and another team just for integration that manages the enterprise service bus. All of these [teams] behave as if they're from three different companies. Every time a business analyst wants to find some data, they have to go to three different teams."

Flexibility's the Thing

The lesson, as Kumar sees it, is that IT needs to get out of the integration business.

"IT has to figure out how to get out of doing open heart surgery [on its data integration infrastructure] every year. [IT] should look for a solution [in which] the integration is assumed—or part of a managed Data Platform as a Service—and has service-level agreements attached to it," he says. "[Traditional] data integration is also mostly monoglot: it's focused on SQL, on relational data. Ours is a polyglot world, [involving] not only different types of data, but different APIs. IT needs to consider polyglot persistence—to either consider building something out that can store all different kinds of data, or taking advantage of a managed service like ours. Data should come in in any format, but when you read the data, it should be in the format and model that you want."

This is why Liaison's ALLOY dPaaS supports true schema-on-read, courtesy of its polyglot Hadoop-based data store. "In the traditional [relational] model, you define the schema of the data first and that's where all of your governance is supposed to be. Then you say, 'The data has to go in and fit this model.' The approach we've taken is very much in line with the way Hadoop is designed to work. You persist the data [in Hadoop], but you define the governance rules *at the time of read*, so what we have is a governance workflow that is very user-self-serviceable.

"The user can actually use our visualization [layer] to go in and say, 'I'm trying to filter out patient billing history, so ... give me data plus the context that I need—billing. This has a whole workflow associated with it: relationships, attributes, and associated rules. The [point is that the] data could be sitting in any format that you want, but the governance is enforced depending on who you are or what you're trying to do with the data."

www.liaison.com

The future is now at Liaison Technologies. As a leader in cloud data solutions, our holistic and tailored approach allows businesses to meet today's toughest data challenges while building a robust foundation from which to tackle tomorrow's.

From complex data integration to data management to the brave new frontiers of big data, our secure solutions break down data silos, reduce inefficiencies, and uncover actionable insights.

Founded in 2000, Liaison serves over 7,000 customers in 46 countries, with offices in the United States, The Netherlands, Finland, Sweden, and the United Kingdom.

- The Liaison ALLOY Platform Brochure
- The Liaison ALLOY Platform White Paper

Advancing all things data.

tdwi.org

TDWI is your source for in-depth education and research on all things data. For 20 years, TDWI has been helping data professionals get smarter so the companies they work for can innovate and grow faster.

TDWI provides individuals and teams with a comprehensive portfolio of business and technical education and research to acquire the knowledge and skills they need, when and where they need them. The in-depth, best-practices-based information TDWI offers can be quickly applied to develop world-class talent across your organization's business and IT functions to enhance analytical, data-driven decision making and performance.

TDWI advances the art and science of realizing business value from data by providing an objective forum where industry experts, solution providers, and practitioners can explore and enhance data competencies, practices, and technologies.

TDWI offers five major conferences, topical seminars, onsite education, a worldwide membership program, business intelligence certification, live Webinars, resourceful publications, industry news, an in-depth research program, and a comprehensive website: tdwi.org.