



# Defining *SolutionScape* and *Digital Fluidity*

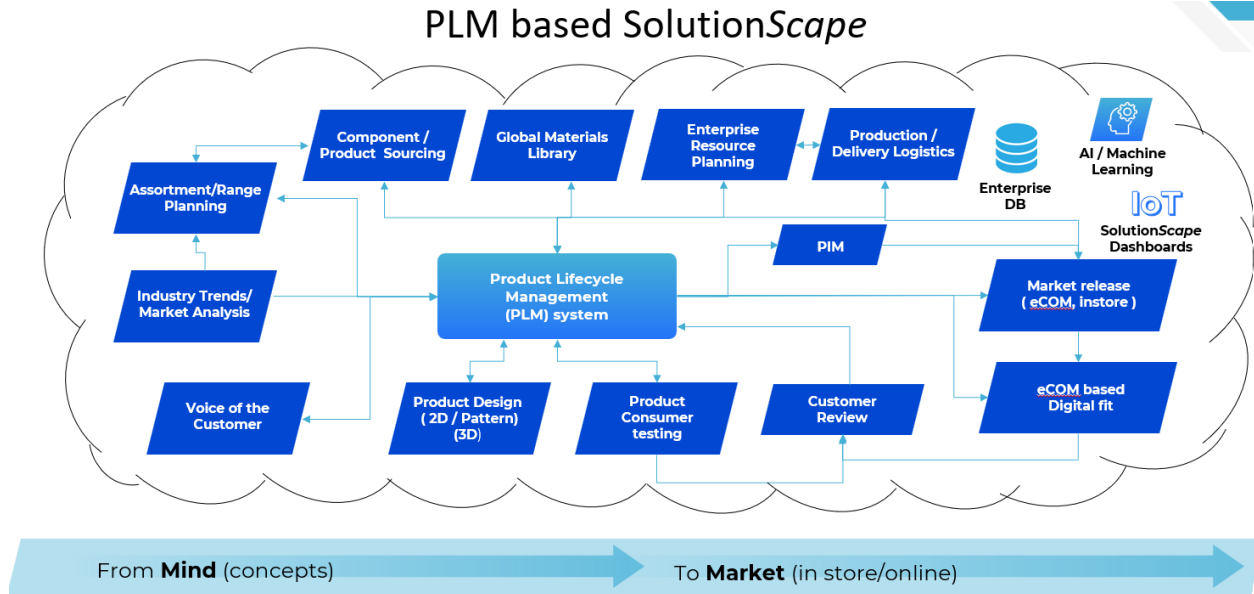
**Author:** Brion Carroll, CEO / CEO / Principal Consultant of Digital Solution Group, LLC

The term **SolutionScape** is defined by **Digital Solution Group (“DSG”)** as the array of systems, applications, and services that are engaged to bring a company’s product from mind to market. As one can assemble, this includes (for most companies) some form of

1. product line or assortment planning,
2. 2D and/or 3D product design,
3. (potentially) voice of the customer,
4. product lifecycle management,
5. product development of a “tech pack”,
6. some level of materials management,
7. some form of color (or finish) management,
8. supply chain management/interaction support,
9. material / product testing,
10. (potentially) enterprise resource planning system,
11. (potentially) product information management (PIM),
12. customer relationship management,
13. logistics management, and
14. any number of micro-systems that serve a unique purpose when blended with or supportive of all the systems, applications, or services previously listed.



The example image below depicts the Apparel / Footwear market of brands and vertical retailers, where there exists a plethora of supportive systems/services that are visible to IoT, Machine Learning/Artificial Intelligence services, and are overlaid with the necessary SolutionScape Dashboarding provides that information aggregation and quick links to specific systems to further investigate relevant information.



At this point one might take a breath to understand the magnitude of **“components”** of a company’s SolutionScape, but one must also step back to realize how many humans are engaged to design, develop, and support all these systems, applications, and services. Some are employed by the vendor of any of these components, while others are consultants brought on board when needed to enhance their operational fitness, and still others are employed by the company to either deliver or support these components in their day-to-day operations.

Now let’s look deeper into each of these components and agree that each has **data** that is the lifeblood of its execution. Data that makes each component function either as a means of obtaining input from its **“users”** or data that is generated as a course of its operations, or data that is provided back to its users.

This data could have many different **“formats”** that could range from generic text, to a numeric (integer or real), to currency (country specific or USD or both), to a date, or a reference to another **“object”**, or even text field that is assembled from or composed of the values of other text fields (often referred to as derived) or a value of a text field whose value is dependent on the value of another field (often referred to as a driven value).



In the case of a text field, it could have a limited set of characters (12 vs 1040 vs unlimited), while a numeric field could have a range set (0-12 vs 1.00 to 3.50 vs unlimited). In the field format referred to as text, it could also be a select set of values, often referred to as a single (or pick) list, or a multi-select list of values that (as noted earlier) could control the value of another list (driven list), such as Category might drive the potential values of a Sub-category field (or “**attribute**”).

There is a wide array of data that is created by, used by, or relied on by any one of the many components of a *SolutionScape*. This reinforces the reason for DSG establishing the term *SolutionScape*. It is component set agnostic. It is data profile agnostic.



There is another phrase we use when discussing how a *SolutionScape* shares data between the various components, and that term is **Digital Fluidity**; which refers to the flow of data between the components in the *SolutionScape*.

If one thinks of all the operational requirements of each component, there is an inherent appreciation that each component (no matter how well “coded”) has some level of requirements about how each piece of data / attribute of each entity / object is interpreted. If the data field is not in alignment with its operational requirements, then it fails to execute properly. If it fails to execute properly, then each component has its own way of reporting this to either a user or a systems administrator, if it does so at all.

In a *SolutionScape* there is never a situation where some portion of the data of one component is not shared (or “**integrated**”) with one or more components. This might be true of a mathematical calculator that is used to multiply two numbers, but it is impossible to have a component never share any data with another component or rely on some amount of data being shared with it.

Therefore, to accomplish *Digital Fluidity* throughout a *SolutionScape* is to establish not just the integration of data between components, but the “**intimation**” (or *intelligent integration*) of data between components. To intimate data between two components is to take all the peculiarities of definition and use of data into account when sharing the data between those two components.

Let’s take an example:

The Assortment Planning (**AP**) has a text field that is derived from the values of three other data fields. For simplicity purposes it is the **Name** of a Product that is derived from the **Model #** (a four-digit numeric data field) + the **Category** (which is a text

field of 20 selectable choices) + **Description** (which is a text field limited to 10 characters). These data value constraints are all enforced by the system of origin – that being the AP system.

This **Name** field (*in this example*) of a given Product is *6195 Generic Product Offering*.

The **Name** field is defined in the AP system and is required to be passed to a Product Lifecycle Management (**PLM**) system (along with other data). Since there are no constraints in the PLM system to have a **Name** field that is (in this example) 34 characters, the **intimation** (intelligent integration) is successful.

Since the **system of origin** of the **Name** field is the AP system, the PLM system is configured to not allow any edits of that data. **If there was** an operational requirement of the business to support the editing of the **Name** field in the PLM system, then the PLM system would have to support the Model, Category, Description data field constraints and use those fields to construct the **Name** field. Otherwise, it would not be possible for the PLM system to (w/o failure) pass the **Name** field back to the AP system, therefore ensuring data integrity of the **Name** field throughout this portion of the *SolutionScape*.

The **Name** field (data) is also passed from PLM to the Enterprise Resource Planning (**ERP**) system and finally to an **ecommerce** web site. This requires that the same level of intimation (*again – intelligent integration*) be established. In this example, the ERP system has a 20-character limit for the **Name** field; so that data field must be truncated during the transmission from the PLM to the ERP system. *However, the ecommerce site has no such limitation.*

Since the ERP system will end up having several “Products” with the same Name value (truncating to 20 characters eliminates the potential inclusion of the Description field), it is required that a unique identification number (**ID**) be established to ensure uniqueness of the Product within the ERP system. Therefore, the AP system must now generate this unique ID and pass it to the PLM system (which requires its support), to then have this ID be included with the Name of the Product, thereby ensuring uniqueness in the ERP system.

Since the ERP system is the system that passes Product data to the **Labeling** System and it is unlikely that a truncated Name or a Product ID will have any meaning to a consumer, it is determined that a **Label Name** data field be created. This Label Name field must be added to the AP, PLM, ERP and Labeling system to fulfill this requirement of the *SolutionScape*.

This is **ONLY** one data field, that has generated two new data fields, along with all of the “can it be edited” decisioning; noting that any component’s editing of any data field has to enforce any of the **original rules** of the system of origin component.

Most companies have dozens to hundreds of data fields that are necessary for any one of the components to operate properly, and upwards of 60% of those data fields must be shared with one or more components.

There is no rest when it comes to supporting the *Digital Fluidity* of a company’s *SolutionScape*.

It is the view of DSG that **all companies** must intimate each component with all upstream and/or downstream component(s) to enable *Digital Fluidity* throughout the *SolutionScape* from Mind to Market. This is what DSG provides as a focused service to Brands, Retailers, Consumer Products, and Consumer Goods companies.

DSG provides a *SolutionScape* approach, leveraging a range of tools that help to give visibility to how intimated components share data with other components **AND** how the various components are technically integration.

**Technical integration** can be achieved via point-to-point (leveraging Application Programming Interfaces (API, or JSON message structures, or staging tables)), or Comma Separated Value (csv) files. Each form of integration must have a means of reporting success or failure. Each integration must have a method of recovery or retry to ensure that each component of the *SolutionScape* has a clear “understanding” of its data integrity **NOT** just within the component, but also respecting the need for data integrity (equalization) throughout the *SolutionScape*.

**Digital Fluidity from Mind to Market is never done by accident.**

To have a *SolutionScape* Assessment done (free of charge) today, either fill out the form below or contact [brion@digitalsolutiongroup.net](mailto:brion@digitalsolutiongroup.net) or [mike@digitalsolutiongroup.net](mailto:mike@digitalsolutiongroup.net).

