What is a data product?

Article • 04/19/2022 • 4 minutes to read

Every application creates and stores data either temporarily or permanently. Many applications also create and save data for operational management purposes, such as error logging and health monitoring. Centralized data teams use ETL processes to consume and process the data these applications produce. Application operation teams often have additional data processing flows for things like application health and KPI status monitoring.

The traditional approach of a waterfall of teams and responsibilities in your data integration isn't ideal. It can lead to knowledge gaps, ownership problems, and communication conflicts that affect your data's quality, timeliness, and value for end users. Application teams are responsible for application performance and success. In their work, they need to make changes to downstream processes owned by other teams, but these changes often don't go according to plan. For example, you might find that a so-called minor upstream change drastically alters a KPI's trend. These kinds of data issues can affect your ability to make critical decisions.

The data mesh approach prevents these issues by adopting the concept of **data as a product**. Application owners and application teams treat data as a fully contained product they're responsible for, rather than a byproduct of some process that others manage. Both applications and analytical data serving tasks are within domain responsibility areas.

Data products are created specifically for analytical consumption. They have defined and agreed-upon shapes, consumption interfaces, and maintenance and refresh cycles, all of which are documented.

Data products are processed domain data assets/datasets shared with downstream processes through interfaces in an SLO. Unless otherwise required, your raw data should be processed, shaped, cleansed, aggregated, and normalized to meet agreed-upon quality standards before you make it available for consumption.

The following sections outline common characteristics that good data products have.

Data product characteristics

Well-designed data products are:

Discoverable, understandable, and trustworthy: Domain teams provide discoverability and understandability by sharing and update information about each data product, its data, its meaning, the format of shape of its data, and its refresh cycle. They communicate changes in data or shape to downstream consumers in a timely manner. Interfaces ensure trustworthiness by providing time-bounded backwards compatibility for data product shapes.

Addressable, natively accessible, and secure: Defined processes for locating and gaining access to each data product provide addressability. Necessary security measures for different access requirements are in place. Data domain ownership mentality shifts from gatekeeping data to serving data with well-defined security precautions. Offered access interfaces are well-documented and can vary in different technologies. Commonly used interfaces for natively accessible data products include APIs, database users, tables or views, and files with necessary access rights.

Interoperable, truthful, and valuable: Data provide interoperability by following defined common standards, such as the same values always having the same name and data type. For example, a column containing customer identification data might be titled *CustomerID* in every data product, and its data might always be an integer, or use snake_case or camelCase in every instance. Data products provide value to customers, and they can also be used as upstream sources for new data products in the same or different domains. However, you can't just carry and copy the same data product in multiple places. Each data product that comes from a previous data product should provide new value and information to downstream consumers. Data products must also always provide truthful, non-erroneous data.

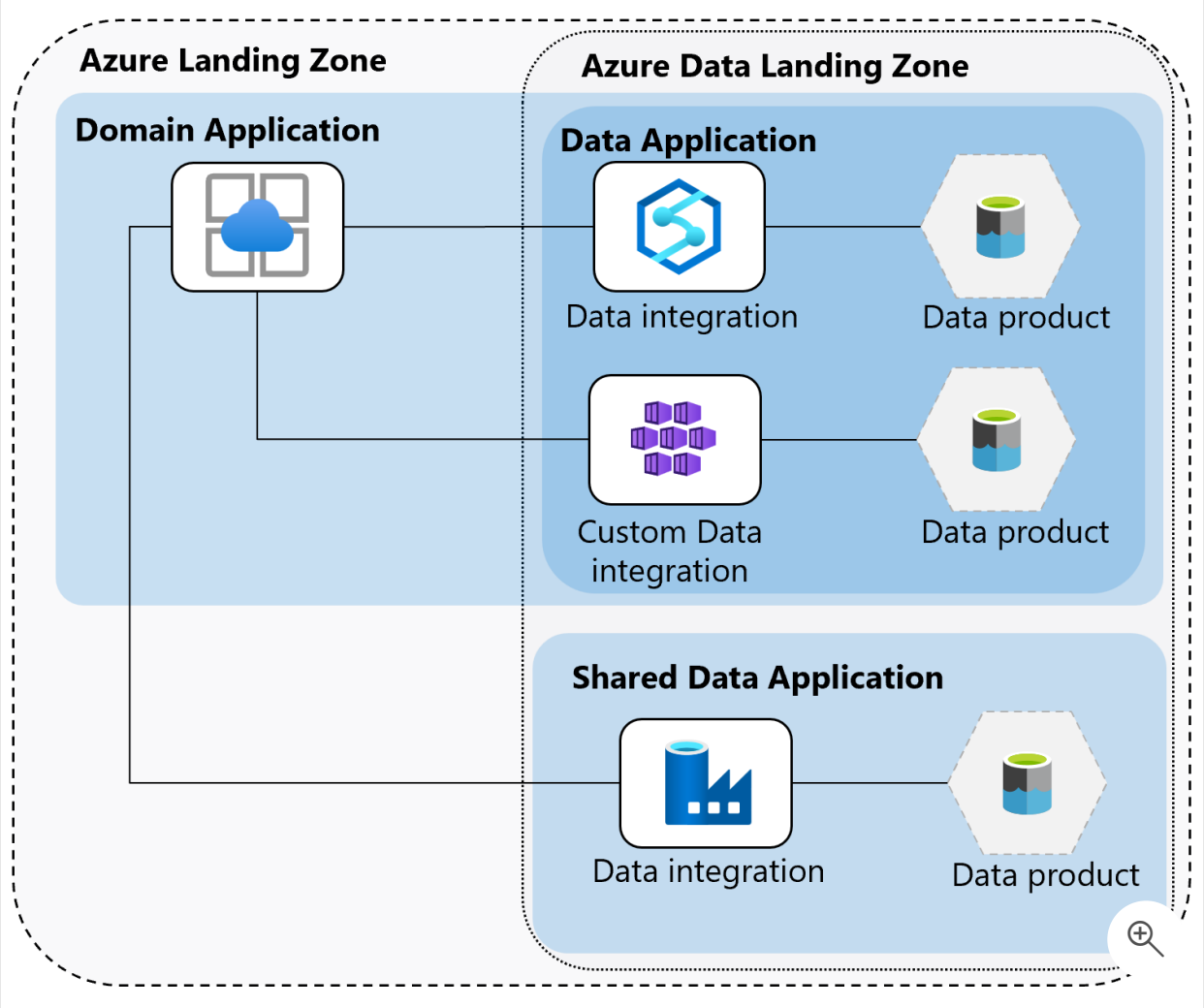
Well-designed, well-maintained data products and their interfaces help organizations avoid duplicating data and can help create a native single source of truth.

Data product design recommendations

To fulfill data product serving requirements, your domain teams must acquire a new set of skills and use new tools and platforms.

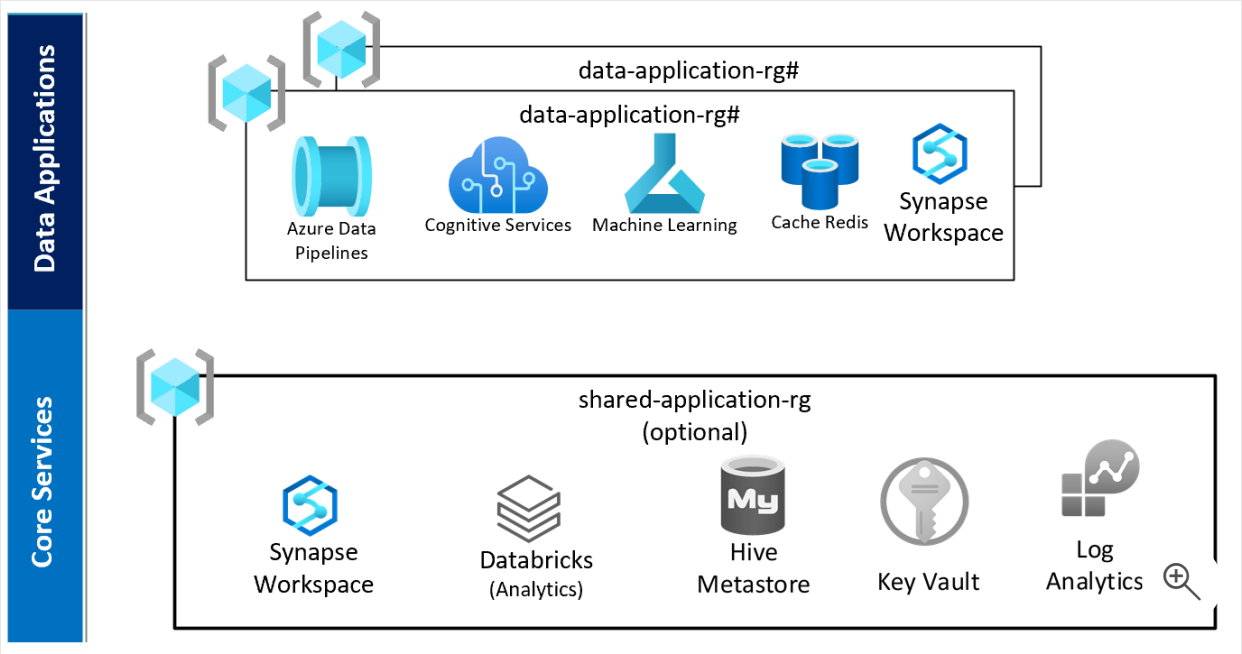
Fully equip your domain application teams to build the data applications and produce or serve data products. Your teams can build data products using a familiar technology stack. They might also prefer to have their own Spark instance or pipeline engine if feasible. For example, a large domain that serves many data products might decide to process and serve data products from their own Azure Synapse Analytics. Smaller organizations and smaller domains of large enterprises might decide to develop and run their data applications on a shared platform, such as a centrally located Azure Data Factory, Azure Synapse Analytics, or Azure Databricks.

Ensure that your data products have the common characteristics described in this article, your lineage repository reflects your data application lineage, and your implementation and access are governed.



Data Product and Data Application Guidance for Azure

You can position all possible approaches for your data application environment within Azure data landing zones if your domain application teams use a shared platform and set of services.



You can find three different data application pattern templates for Azure data landing zones in Cloud-scale analytics data products in Azure - Sample data applications.

Next steps

- Design considerations for self-serve data platforms