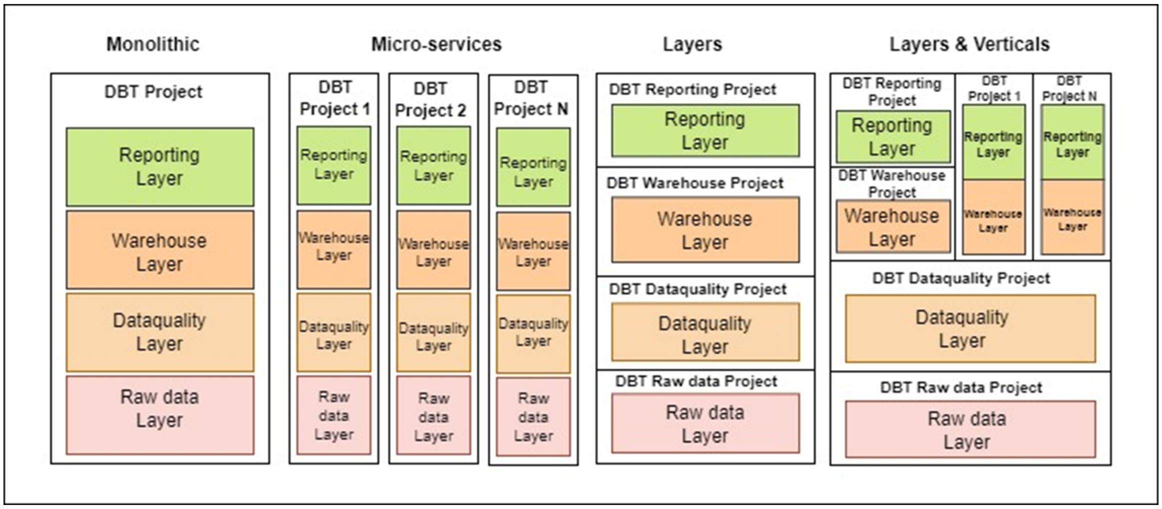
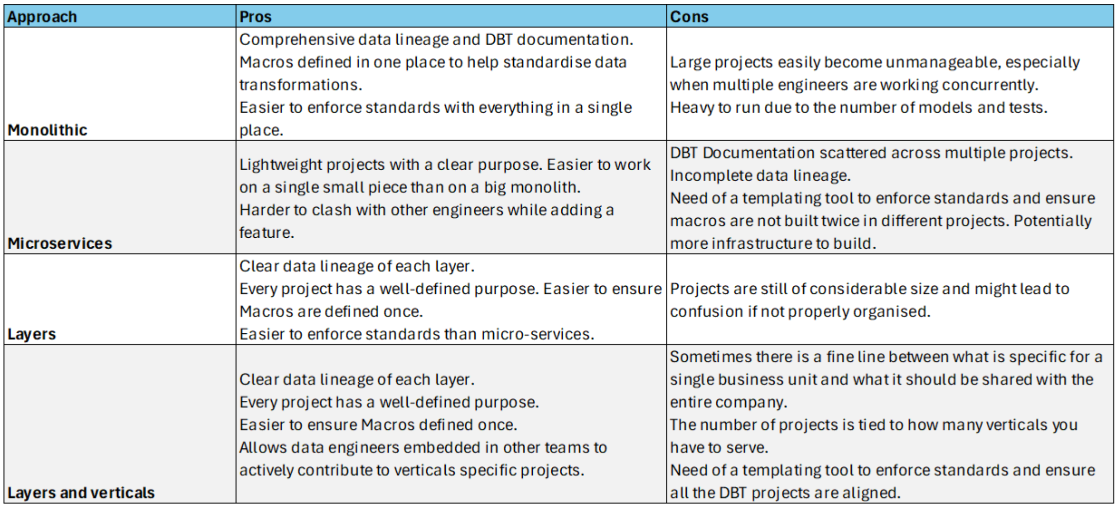
All about DBT projects



Types of DBT projects Architectures

- **Monolithic:**A single dbt project for your entire platform. It will clean your data, build fact and dimension and materialize any table that might be needed to support your reporting and analytics.
- **Micro-services**: Multiple small projects, each with a single purpose. Projects based on Subject areas within Business verticals catering specific needs of the business.
- **Layers:** Your data platform will be normally structured in logical layers. Like Raw-data/Landing layer, Data-quality/Staging layer, Warehouse /SDDS layer, Reporting/Publish layer.
- **Layers and verticals:** This is a Hybrid approach. It's like Layers approach but after Data-quality one DBT project is built with all facts and dimensions relevant for the entire organisation and some vertical-specific projects are created to serve specific areas of the organization.

DBT projects architecture Merits & Demerits



Approach	Pros	Cons
Monolithic	Comprehensive data lineage and DBT documentation. Macros defined in one place to help standardise data transformations. Easier to enforce standards with everything in a single place.	Large projects easily become unmanageable, especially when multiple engineers are working concurrently. Heavy to run due to the number of models and tests.
Microservices	Lightweight projects with a clear purpose. Easier to work on a single small piece than on a big monolith. Harder to clash with other engineers while adding a feature.	DBT Documentation scattered across multiple projects. Incomplete data lineage. Need of a templating tool to enforce standards and ensure macros are not built twice in different projects. Potentially more infrastructure to build.
Layers	Clear data lineage of each layer. Every project has a well-defined purpose. Easier to ensure Macros are defined once. Easier to enforce standards than micro-services.	Projects are still of considerable size and might lead to confusion if not properly organised.
Layers and verticals	Clear data lineage of each layer. Every project has a well-defined purpose. Easier to ensure Macros defined once. Allows data engineers embedded in other teams to actively contribute to verticals specific projects.	Sometimes there is a fine line between what is specific for a single business unit and what it should be shared with the entire company. The number of projects is tied to how many verticals you have to serve. Need of a templating tool to enforce standards and ensure all the DBT projects are aligned.