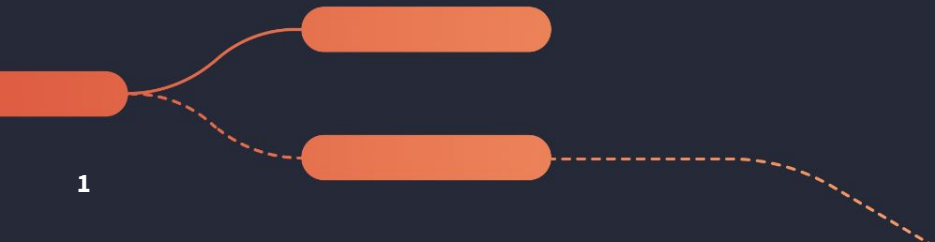
dbt Mesh Working Session

Part 1: Overview of features



Analytics8

Why consider a dbt Mesh of multiple projects?

- Too many models in your project
- Separate workflows developed by different teams
- Communication challenges, and lack of reliability for your data products
- Security and governance requirements are increasing and would benefit from increased isolation.

From the dbt Labs Best Practice Guide: How we build our dbt Mesh projects

https://docs.getdbt.com/best-practices/how-we-mesh/mesh-1-intro

Model Access & Groups

Access Level	Can be referenced by
Protected (default)	Models in the same project
Private	Models in the same group
Public	Models from any project/group

- Use groups to define an “owner” for a dbt model
- Groups are a great bridge to a full multi-project mesh

```
groups:
  - name: customer_success
    owner:
      name: Customer Success Team
      email: cx@jaffle.shop

models:
  - name: dim_customers
    group: customer_success
    access: public
```

https://docs.getdbt.com/docs/collaborate/govern/model-access#access-modifiers

Model Contracts

- Enforce model contracts for reliable, consistent models
- The column names, order, and datatypes specified in the YAML are used to materialize the model (vs the models' SQL)
- Enables downstream dbt projects to trust the public mart models they are referencing

```
models:
  - name: dim_customers
    config:
      contract:
        enforced: true
    columns:
      - name: customer_id
        data_type: int
        constraints:
          - type: not_null
      - name: customer_name
        data_type: string
      ...
```

https://docs.getdbt.com/docs/collaborate/govern/model-contracts

Model Versioning

```
models:
 - name: dim_customers
   latest_version: 1
   config:
     materialized: table
     contract: {enforced: true}
   columns:
     - name: customer_id
       description: This is the primary key
       data_type: int
     - name: country_name
       description: Where this customer lives
       data_type: varchar

   versions:
     - v: 1
     - v: 2
       columns:
         - include: all
           exclude: [country_name]
```

https://docs.getdbt.com/docs/collaborate/govern/model-versions

Cross-project references

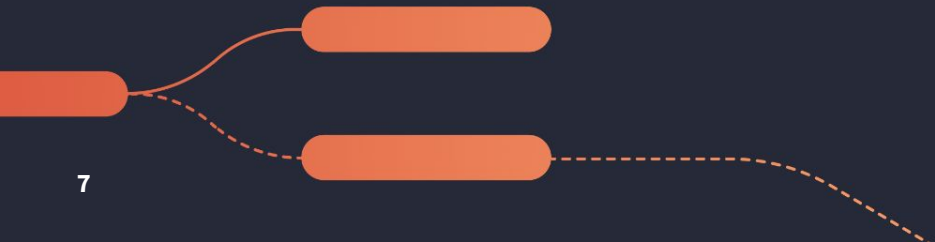
```
with monthly_revenue as (

   select * from {{ ref('jaffle_finance',
'monthly_revenue') }}

),
```

- Reference public models from another dbt project
- Specify the project's name in the ref

https://docs.getdbt.com/docs/collaborate/govern/project-dependencies#how-to-write-cross-project-ref

dbt Mesh Working Session

Part 2: role-playing exercises



Analytics8

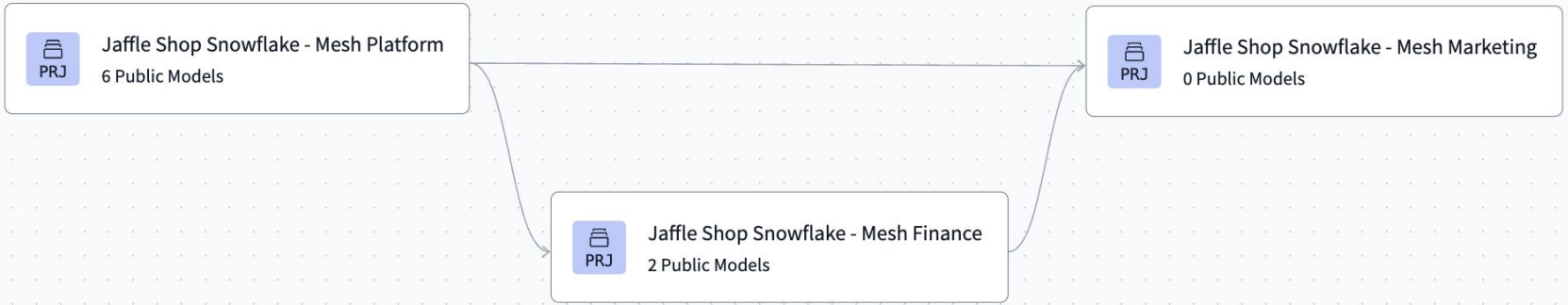
You are now part of The Jaffle Shop Mesh

Your mission:

Collaborate in groups of 3 to come up with processes that will solve problem scenarios.

Choose your role:

- The Marketing analytics engineer
- The Platform engineer
- The Finance analyst
- The data scientist (scenario 3)



Dealing with breaking changes & conflicting needs

An analytics engineer from the marketing team needs an update to a model produced by the data platform team - a case statement needs to be altered.

The **marketing analytics engineer** submits a PR with the change to the platform repo, a **data platform engineer** approves the PR, and soon the updated model is in production.

Unfortunately, this ends up breaking a dashboard that the finance team made, which depends on a finance model that references the same data platform model. A **finance analyst** asks for the change to be reverted.

Negotiating access & untangling lineage

The marketing project currently uses the 2 public models from the finance project. However, the **marketing analytics engineer** identifies another protected intermediate model in the finance project that contains data they would like to use for a new dashboard (and this data is not in either of the public finance models).

The **finance analyst** is hesitant to make an intermediate model public, and points out that much of the data in this model comes from a data platform public model (though some columns are unique to the finance project). The **platform engineer** would like for shared resources to always be part of the platform project.

The twist: When the finance analyst investigates the marketing use case, they identify a marketing model that they would like to reference (e.g. ad spend).

The mesh is growing… (maybe?)

The data science team has been working in the Platform project up till now. However, the section of the DAG that is entirely data science models has been growing, and increasing the complexity of the Platform project. The data science team wants to make sure that other models in the Platform project are not referencing models too early in their chain of transformations, and they also are starting to develop different processes around PRs.

A **data scientist** from the Platform team proposes that a data science project splits off from the Platform project. The **platform engineer** is concerned that dependencies with other platform models are still too tightly intertwined. A **marketing analytics engineer** thinks some of their models would belong in a data science project.

Thank you

This presentation deck will be sent out tomorrow

Analytics8