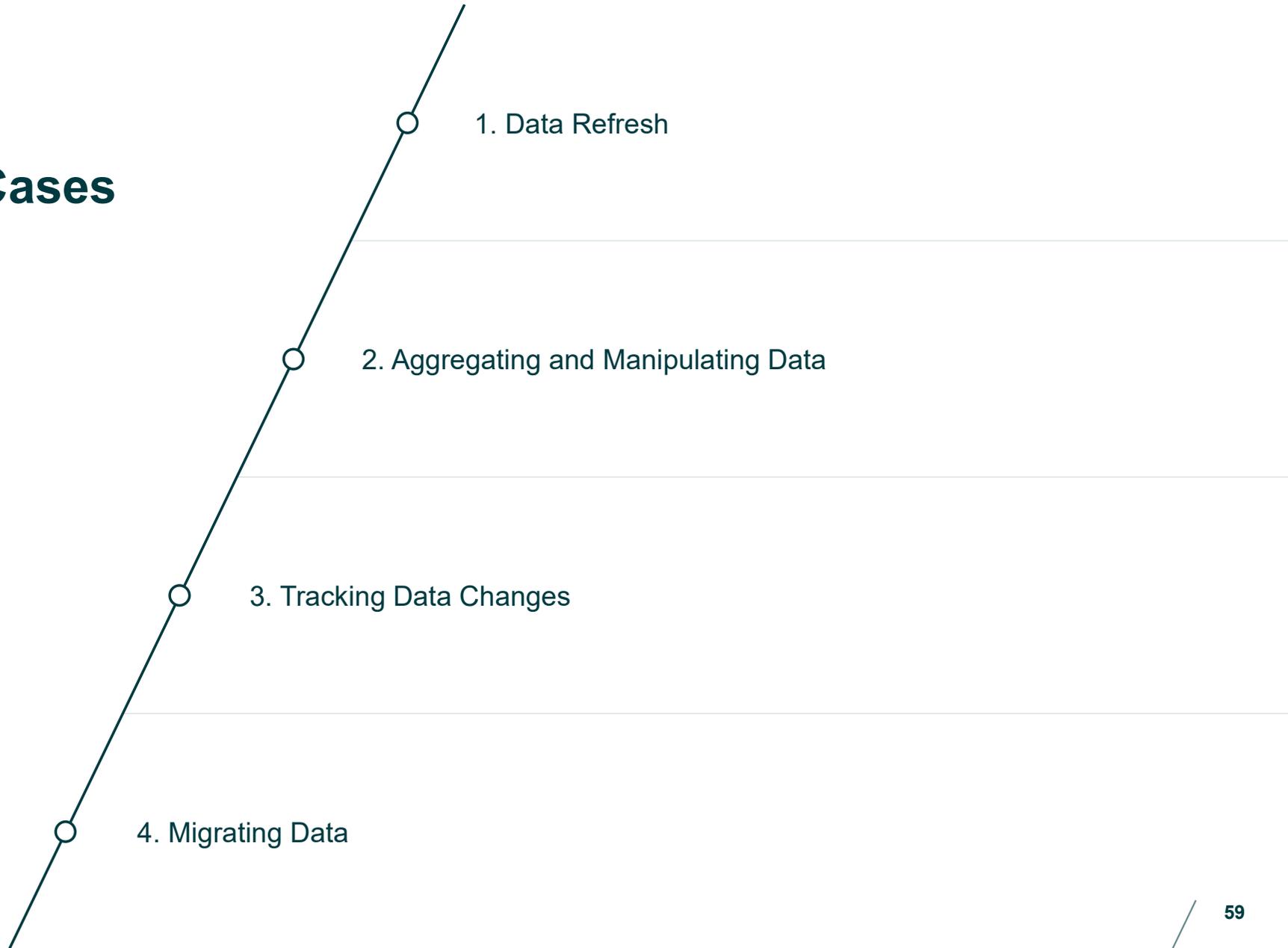


Blender Use Cases

Data Refresh, Aggregating and Manipulating Data, Tracking Changes and Migrating Data

Blender Use Cases



Large data refresh across multiple tenants on a daily basis

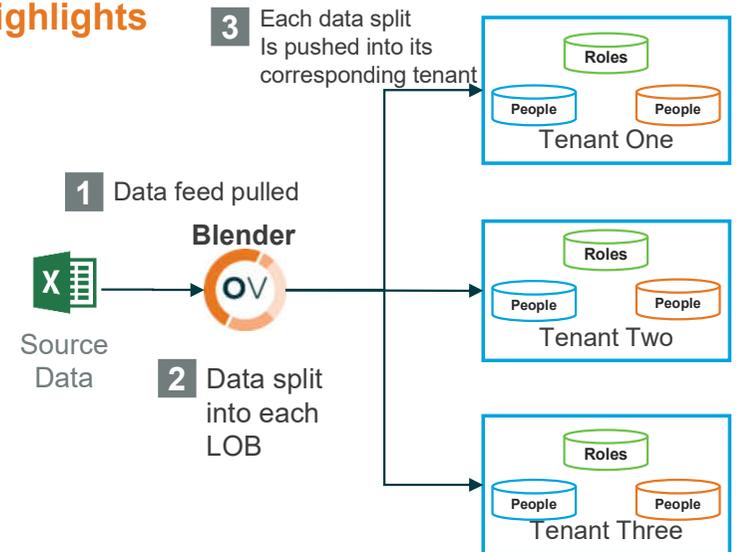
Challenges

- Large volume of data that needs to be refreshed on a daily basis
- Multiple datasets within each tenant that need to be refreshed
- Complex security rules
- If the process was done manually the refresh would take ~40 hours
- If refreshed manually there are opportunities for human error leading to security issues



Highlights

- Error reporting and logging mitigates manual refresh risk
- Entire refresh of data takes between 1-2 hours
- Security is enforced through data being split between tenants, lowering the risk of security breaches



Results



More auditable through error logging and reporting



Significant reduction in time taken to refresh data



Lower risk of data security issues

Aggregating approved, modelled data between tenants to create a merged master dataset

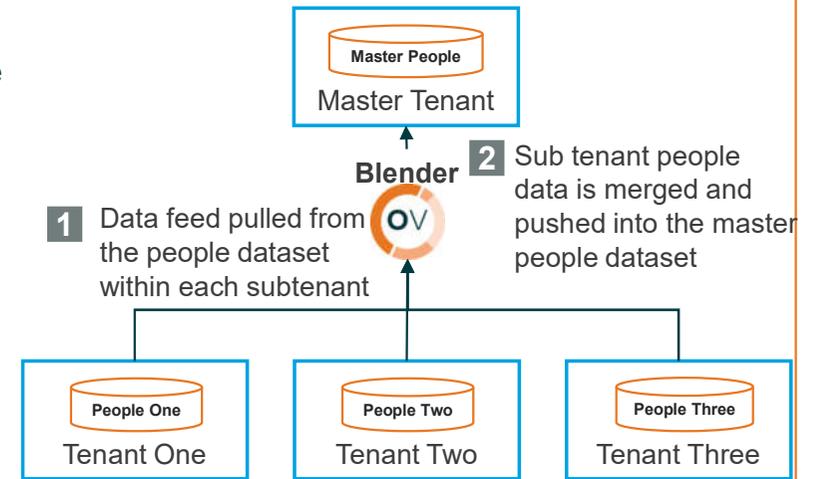
Challenges

- Multiple tenants that contain modelled datasets
- Modelled data has been approved by business owners, this may be the case within one tenant or all tenants
- Master tenant will contain a large volume of data
- Security rules ensure that users only have access to specific tenants and datasets
- Once modelled the impact of the changes should be assessed for the entire organisation



Highlights

- Datasets that are approved are tagged to indicate that they should be merged into the master.
- Approval process is handled by the client
- Data is merged into the large master dataset without human intervention
- Security is enforced through data being split between tenants, lowering the risk of security breaches



Results



More manageable through data slicing



Significant reduction in time merge large datasets



Lower risk of data security issues through tenant separation

Comparing periodical data to track changes over time

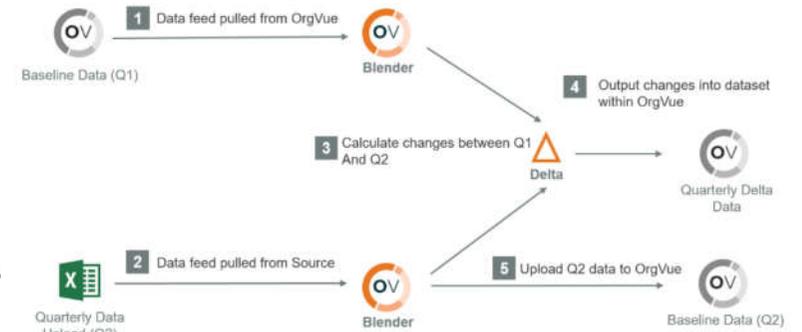
Challenges

- A new dataset is created each quarter
- The quarter datasets reside within OrgVue historically
- Need the ability to compare different periodical data and output a report showing leavers, movers and joiners between dates



Highlights

- Data refresh is also handled in an automated way through Blender
- Once the new quarter dataset is created it is compared against the previous quarter data
- A delta dataset is created showing the changes, additions and deletions between the two datasets within OrgVue



Results



More auditable through automated refresh and delta reporting



Significant reduction in time taken to produce change reports between datasets



Flexible comparison of datasets

Copying data and dataset configuration between tenants or tabs

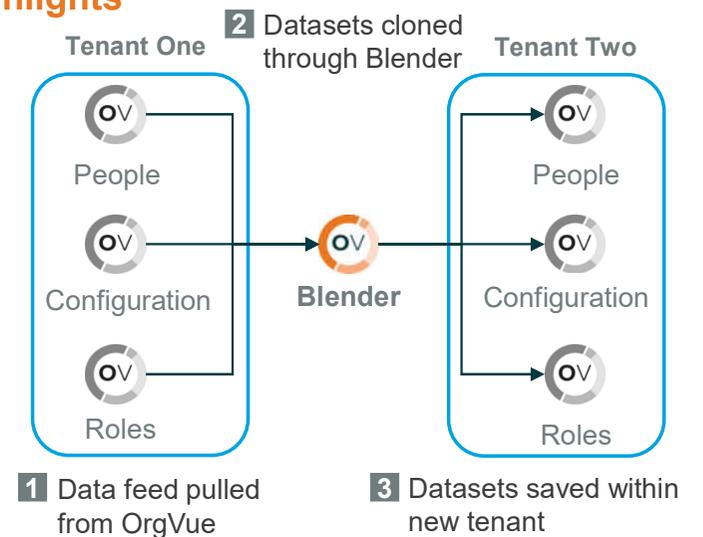
Challenges

- A group of standard datasets need to be copied from one tenant to another for reuse
- Each dataset has a number of pinned reports and branches that also need to be copied over
- For some of the datasets only configuration needs to be copied, for other datasets data will also need to be included



Highlights

- Datasets can be copied with out without data
- Standard configuration can be copied to multiple tenants or multiple tabs within tenants
- All pinned reports and branches can be copied over with each dataset
- Datasets can be copied in bulk while still allowing individual configuration options



Results



More auditable through logging and reporting



Significant reduction in time taken to duplicate standard datasets



Lower risk of data security issues as data is encrypted during copying

Benefits and Cautions

Good and bad use cases of Blender

Blender should only be used to solve specific problems

Good use cases include:

- Data refresh across a standard set of datasets on a daily, weekly, monthly basis.
- Aggregating large data within or between tenants
- Creating subsets of data from OrgVue datasets
- Splitting a source file into multiple OrgVue datasets
- Copying configuration from one dataset or tenant to another
- Hierarchical transformations and checks before loading data
- Usage reporting

Bad use cases include:

- Repairing source data on each data refresh
- Complex transformations that can be completed elsewhere