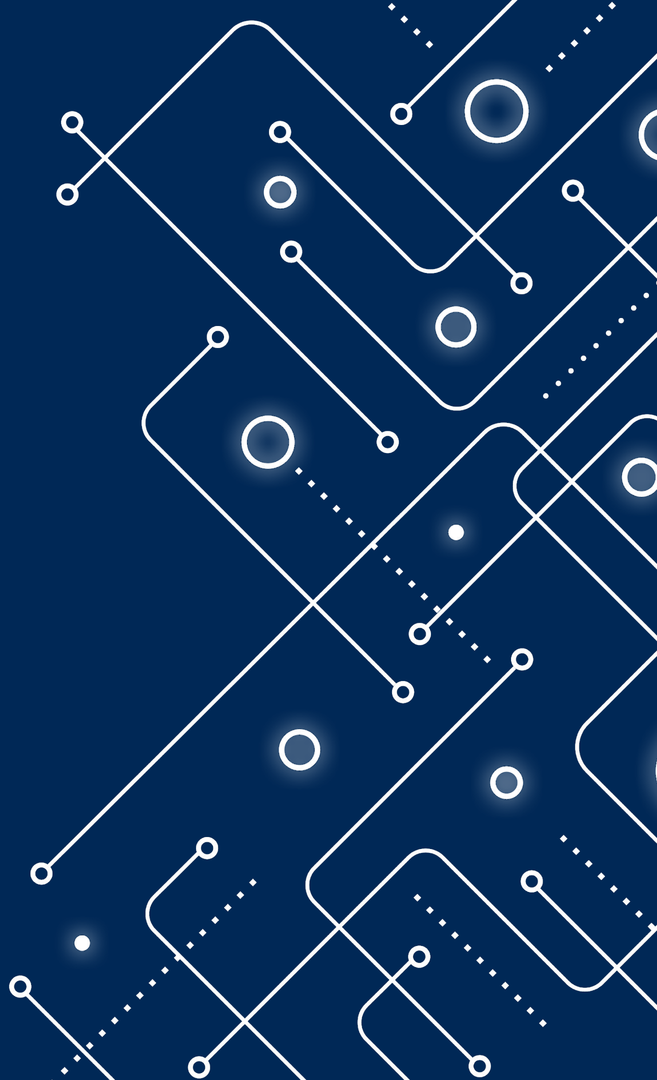
Collaboration for Structured Data



Outline

- Motivation
  - Why do we need structured Data?
  - The Problems of Data Management for Teams
- Challenges
  - Why can't I just use git?
  - Why can't I just use my Database?
- Solution
  - Distributed Data Collaboration using Revision Control

Motivation: Why Structured Data?

Data is Core

Structured data gives us the ability to surface the data we need to software (which can then give it to people)

- Data analytics / data science is required for good decisions, but machines need to be able to eat the data. Currently 80% of time taken in curation.
- Curation and editing to obtain high quality data is therefore critical and it is facilitated by well structured (schema controlled) data.
- Publishing of data requires easy access (query) and sufficient information to surface the data appropriately (graphs, charts, forms, web-pages etc.)

Challenges: Data is Still in the Dark Ages

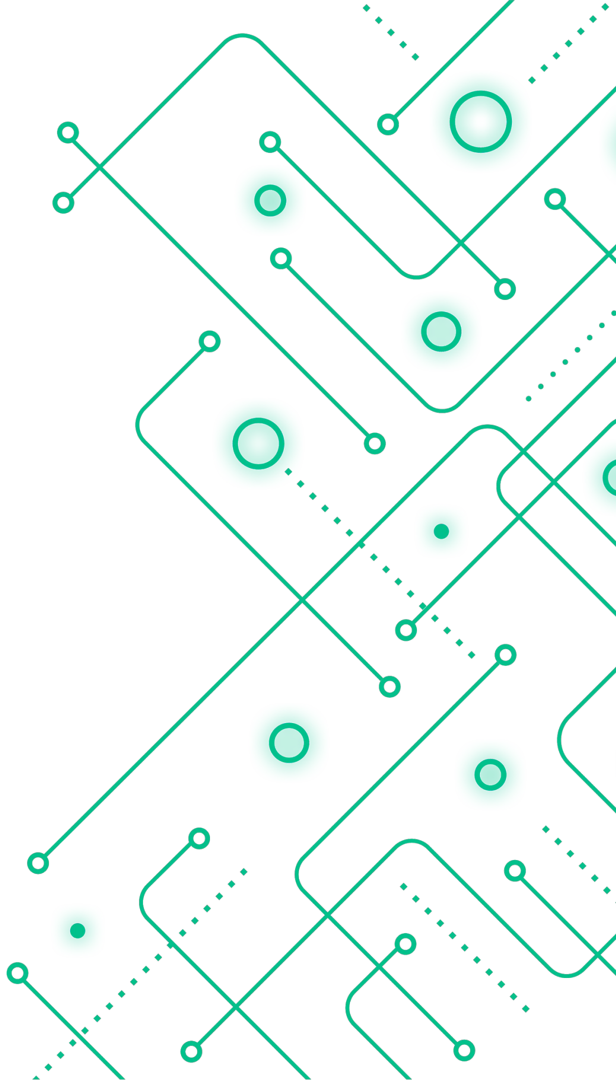
Structured or Unstructured, we're doing it wrong

- Core data in excels and CSVs
  - Distribution via e-mail and slack
  - Repeatedly recleaning, reparsing, recasting
  - Nobody is sure which version it is "foo1.3_final2.xls"
- Git isn't really the right tool for data
  - Git can be used but awkward and stores changes as lines of text.
  - It doesn't scale brilliantly on data
  - We need *structure* and *discoverability*
- Databases are awkward (that's why the csvs are still there)
  - You can add revision features (by table, by row, by etc.)
  - But doesn't resolve changes to structure (schema)
  - You have to hand-roll your distribution
  - Centralised and not "bottom-up", increasing the costs of experimentation

Managing Data means Collaborating

Software *solved* "team" with **git**

- Provenance (revisions and authorship)
- Safety (rollback, branching)
- Quality (CI/CD pipelines)
- Distribution (push/pull/clone/merge)



Solution:

We need databases *designed* for Distributed Collaboration

But What Should a Tool **Built** for Collaboration on Structured Data Look Like?

Discoverability & Schema

Structured data requires a real database - not just git

- Queries: Easy retrieval and update programmatically (not just updating a monster CSV file)
- Structured and typed entities - data cleaning and casting should not be a constantly re-solved problem
- Having a schema means we pay attention has to be payed to schema migration

Revision Control

- Scalable to typical dataset sizes
- Provenance - authorship, commit time etc.
- Pipelining - Modern CI/CD architectures work because of branching
- Safety - branch and rollback allow test and experimentation (before pushing to production)

Collaboration

Collaboration is really the main reason your data-engineers are likely still using CSVs and Excels and not your centralised database.

- Modification has to be ***safe***
- Effective collaboration ***requires*** revision control
- Distribution between collaborators: **push** / **pull** / **clone** - remove the perceived need to slack / e-mail data.
  - I need the data where I have the problem - on my computer or on my ML processing server
  - Distribution requires ***deltas***, compact representations of only what was changed
- Change management: **merge** - gives you the ability to do truly *distributed* data management
- End-to-End encryption should be possible

There are multiple solutions to this problem from custom toolchains through to a number of open-source databases such as DVC, Dolt and TerminusDB.

If you're interested in an open-source solution to collaborative revision control for graphs or complex datasets, you should Give TerminusDB a try!

Terminusdb.com