Y42

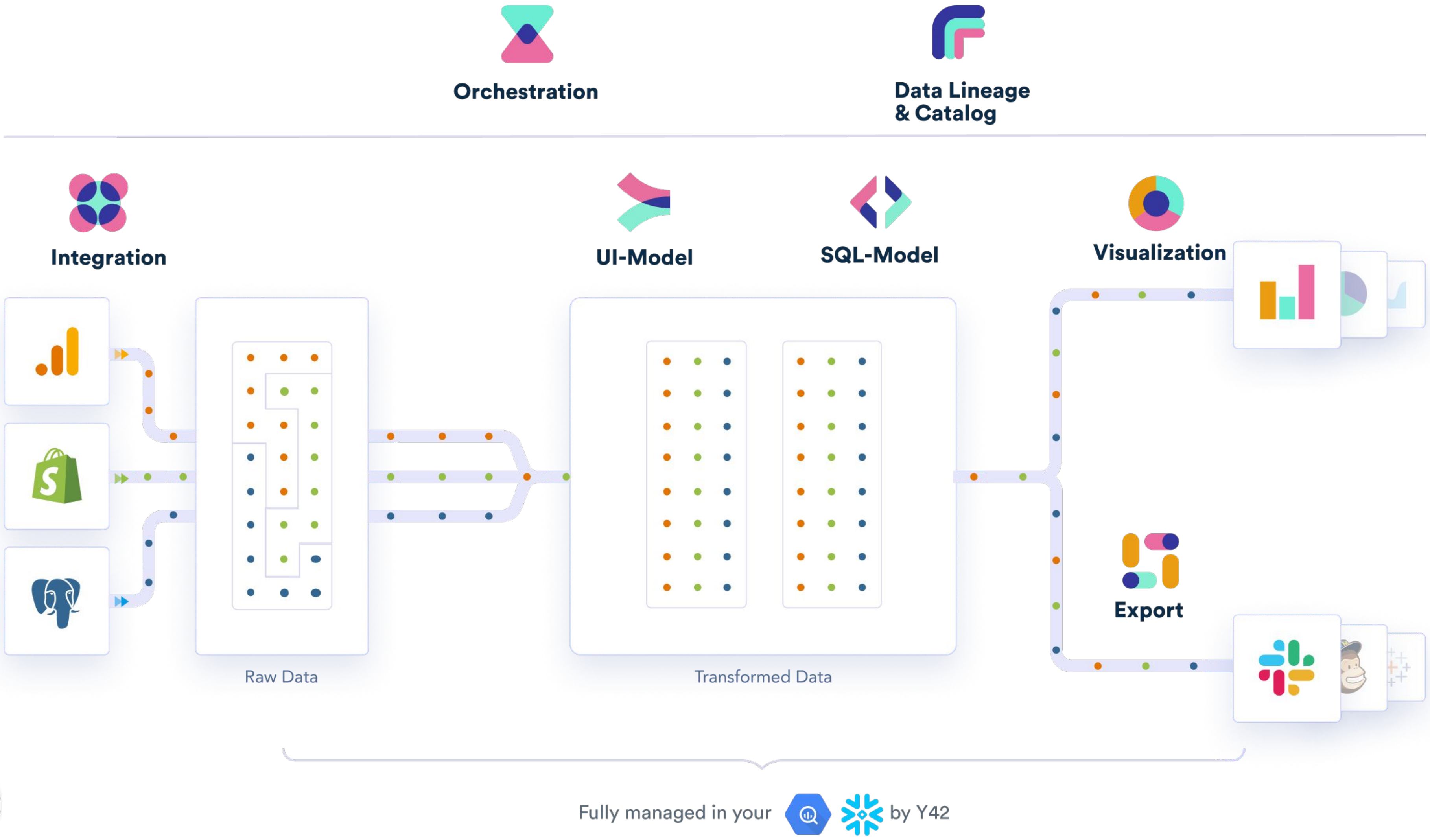
Using Git as a NoSQL database to unleash a new Data Revolution

Data Council – 24.03.2022

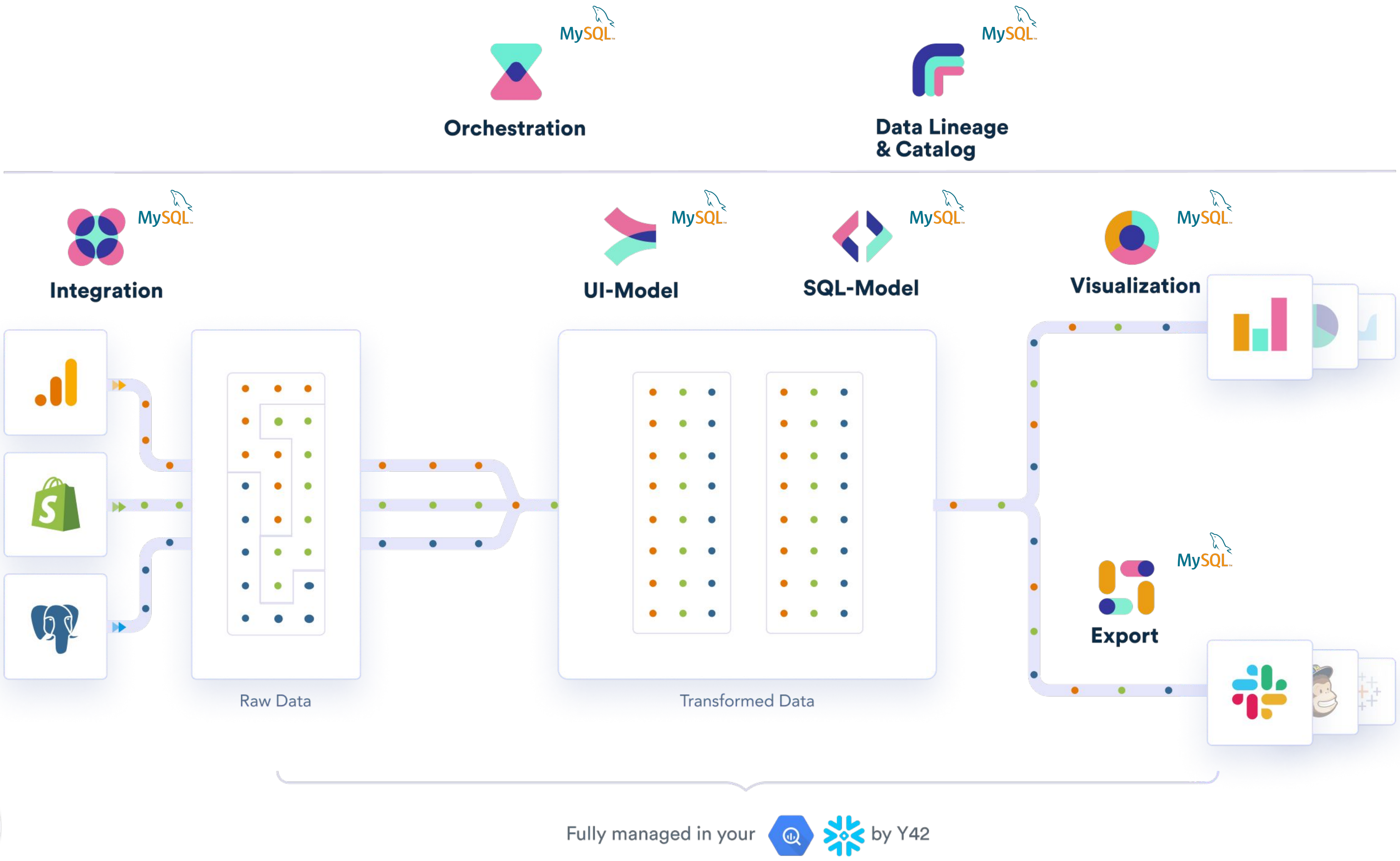
VIDEO 1

Old Y42 set-up

Y42 Initial Set-Up



Y42 Initial Set-Up



Orchestration

Data Lineage & Catalog

Integration

UI-Model

SQL-Model

Visualization

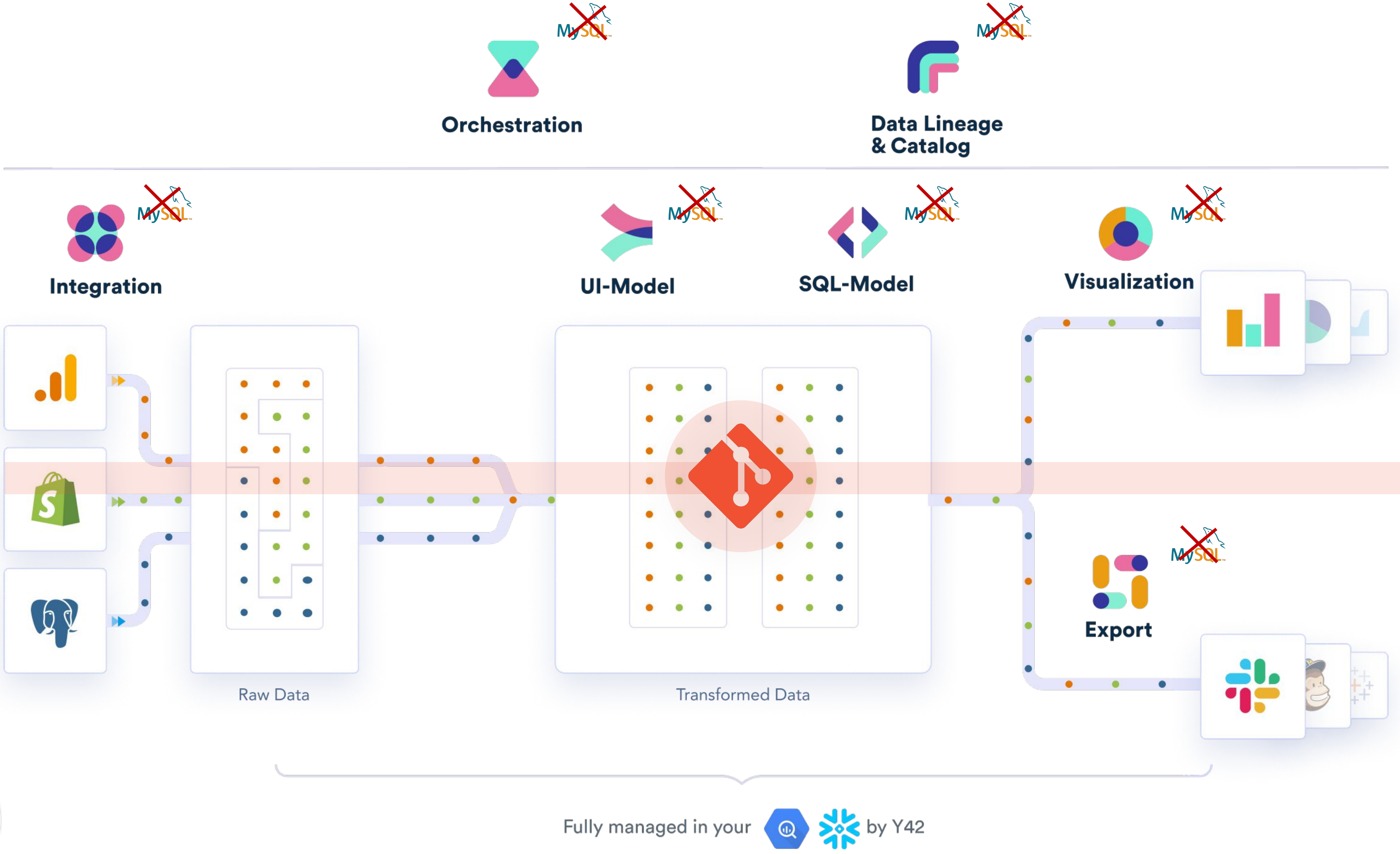
Export

Raw Data

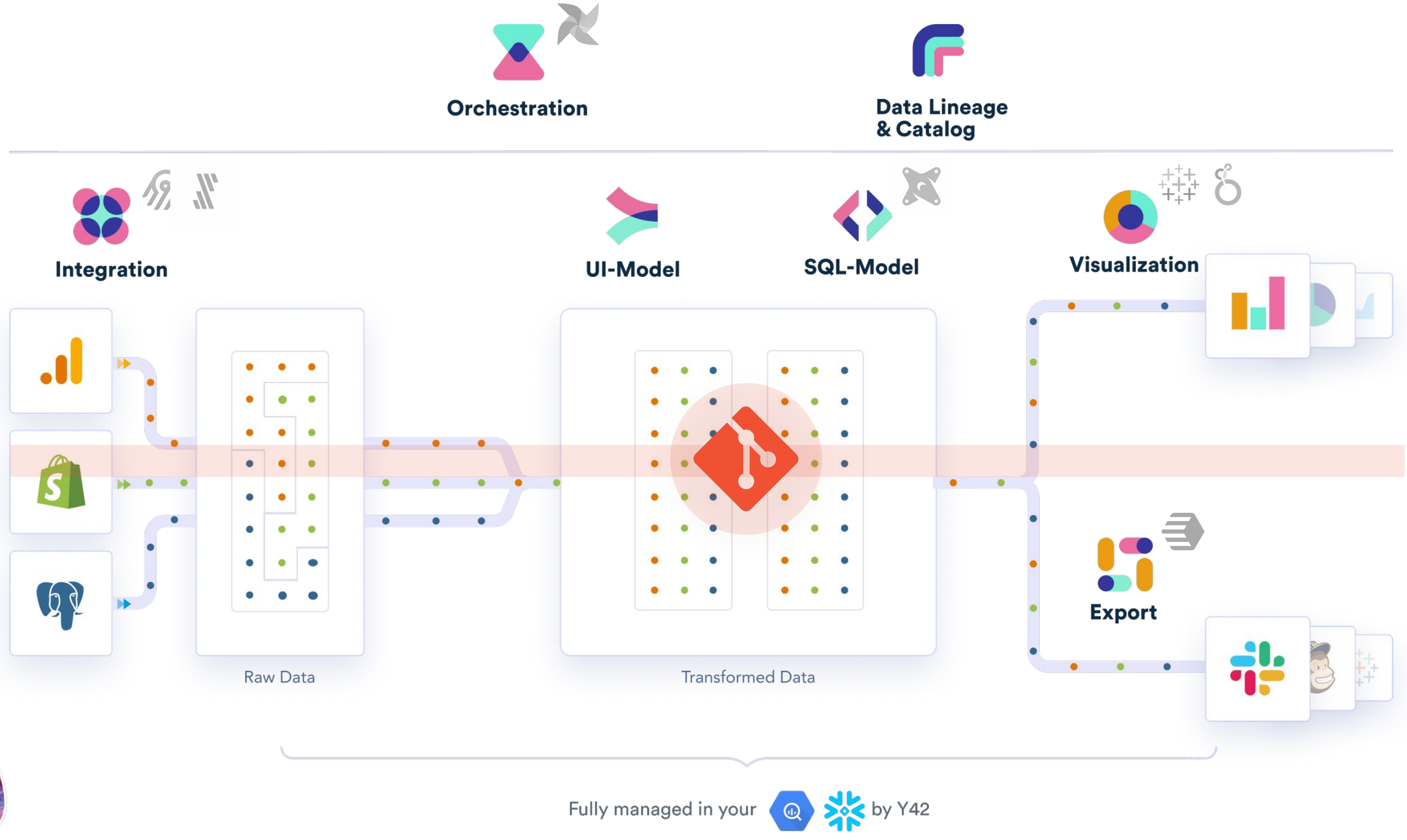
Transformed Data

Fully managed in your by Y42

Y42 Replacing mySQL with Git



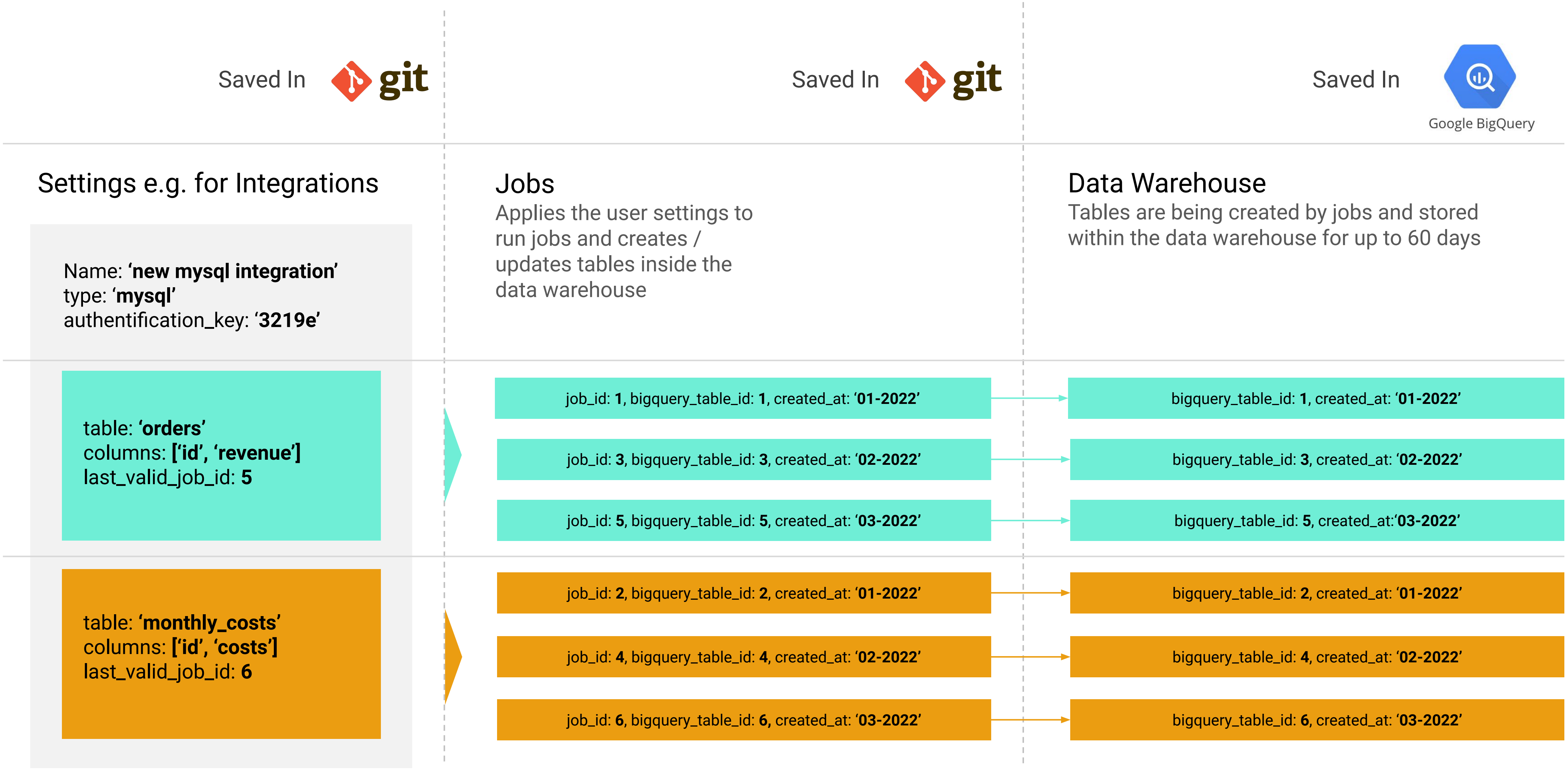
Y42 Connecting with the Modern-Data-Stack



VIDEO 2

New Y42 set-up

Y42 Settings > Jobs > Data Warehouse Tables



Saved In git

Saved In git

Saved In Google BigQuery

Settings e.g. for Integrations

Name: **‘new mysql integration’**
type: **‘mysql’**
authentification_key: **‘3219e’**

table: **‘orders’**
columns: **[‘id’, ‘revenue’]**
last_valid_job_id: **5**

table: **‘monthly_costs’**
columns: **[‘id’, ‘costs’]**
last_valid_job_id: **6**

Jobs

Applies the user settings to run jobs and creates / updates tables inside the data warehouse

job_id: **1**, bigquery_table_id: **1**, created_at: **‘01-2022’**

job_id: **3**, bigquery_table_id: **3**, created_at: **‘02-2022’**

job_id: **5**, bigquery_table_id: **5**, created_at: **‘03-2022’**

job_id: **2**, bigquery_table_id: **2**, created_at: **‘01-2022’**

job_id: **4**, bigquery_table_id: **4**, created_at: **‘02-2022’**

job_id: **6**, bigquery_table_id: **6,** created_at: **‘03-2022’**

Data Warehouse

Tables are being created by jobs and stored within the data warehouse for up to 60 days

bigquery_table_id: **1**, created_at: **‘01-2022’**

bigquery_table_id: **3**, created_at: **‘02-2022’**

bigquery_table_id: **5**, created_at:**‘03-2022’**

bigquery_table_id: **2**, created_at: **‘01-2022’**

bigquery_table_id: **4**, created_at: **‘02-2022’**

bigquery_table_id: **6**, created_at: **‘03-2022’**

Git as-a-database Challenges

- Performance
- Access Control
- App UX to seamlessly integrate Git & code

Y42 Performance Concerns

Alex 10:06 AM
hey so regarding the worries about a git repository's size,

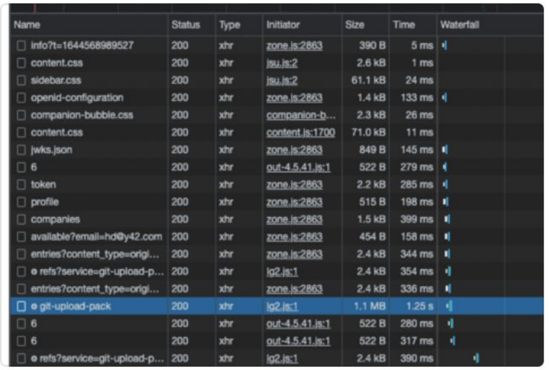
i've created a branch `performance-test-1` and just stupidly copied all of the models many times, there are now 608 integrations, 216 models, 176 orchestrations and 48 visualizations, which takes 255 megabytes space on disk.

i actually can't believe myself when saying this but the `git clone` takes only 1.1MB in download lol as you can see in the screenshot.

- git clone takes about 2.5 seconds
- checking out the `performance-test-1` branch takes about 15 seconds
- loading the integrations list takes some additional time but that is not yet optimized so I'm not worried about that

refreshing the app again with an already cloned repo took about 10 seconds until the integrations started loading

image.png ▾



Name	Status	Type	Initiator	Size	Time	Waterfall
□ info?t=1644568989527	200	xhr	zone.js:2863	390 B	5 ms	
□ content.css	200	xhr	jsu.js:2	2.6 kB	1 ms	
□ sidebar.css	200	xhr	jsu.js:2	61.1 kB	24 ms	
□ openid-configuration	200	xhr	zone.js:2863	1.4 kB	133 ms	
□ companion-bubble.css	200	xhr	companion-b...	2.3 kB	26 ms	
□ content.css	200	xhr	content.js:1700	71.0 kB	11 ms	
□ jwks.json	200	xhr	zone.js:2863	849 B	145 ms	
□ 6	200	xhr	out-4.5.41.js:1	522 B	279 ms	
□ token	200	xhr	zone.js:2863	2.2 kB	285 ms	
□ profile	200	xhr	zone.js:2863	515 B	198 ms	
□ companies	200	xhr	zone.js:2863	1.5 kB	399 ms	
□ available?email=hd@y42.com	200	xhr	zone.js:2863	454 B	158 ms	
□ entries?content_type=origi...	200	xhr	zone.js:2863	2.4 kB	344 ms	
□ ● refs?service=git-upload-p...	200	xhr	lg2.js:1	2.4 kB	354 ms	
□ entries?content_type=origi...	200	xhr	zone.js:2863	2.4 kB	336 ms	
□ ● git-upload-pack	200	xhr	lg2.js:1	1.1 MB	1.25 s	
□ 6	200	xhr	out-4.5.41.js:1	522 B	280 ms	
□ 6	200	xhr	out-4.5.41.js:1	522 B	317 ms	
□ ● refs?service=git-upload-p...	200	xhr	lg2.js:1	2.4 kB	390 ms	

❤️ 1

Chris 3:29 PM
fyi: just tested it a bit locally, with a local git repo:
had 11k files (somwhat similar to the run files), with 4 commits each

- the repo ended up with 12GB data in the `.git` folder 😮
- a shallow clone (depth=1) was ~50mb (roughly the size of the files)
- git gc takes a while (1-2min), but brings down the overall repo size to 94Mb, while maintaining the full history
- `git gc --aggressive` is a bit slower, but brings down the size to ~74Mb (still with full history)

Hung Dang 3:40 PM
Is that good or bad? 😂

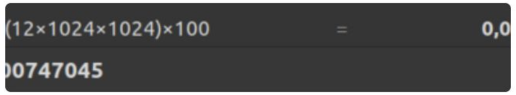
Chris 3:42 PM
good 🙂 initially it's quite big, but we can bring it back down to almost only the file sizes inside of the repo, so it's not forever growing

Hung Dang 3:46 PM
So if we have our own git server life is good again and the Frontend just clones from our git server that optimizes everything regularly? (edited)

Chris 3:49 PM
I assume github,... also have the garbage collect enabled, so we probably could always clone everything - but with our own git server we could enforce it - or clone only a part of the history, which is small.

and to explain the "good": history is big, but we can shrink it back down to 0.0007% of it's size with a single command

image.png ▾



```
(12×1024×1024)×100        =        0,0
00747045
```

🙏 1

Y42

Git as-a-database Powers an all-in-one Data Workspace

All-in-one Data Workspace

- No more stitching together / switching between 5+ tools
- Integrates with the Modern-Data-Stack
- Full transparency, orchestration and observability across every data pipeline, from Integration to Visualization
- Omniscient data lineage and catalog with global search

Openly Programmable

- Build and run everything-as-code paired with the Y42 UI, API or CLI.
- Y42's open programmability makes Templates easy to use and easy to create.
- Spin up entire pipelines and pre-built environments in one click.

Collaboration at Scale

- Analysts and engineers collaborate seamlessly using code and/or no-code.
- Version control, auditing, backup, and replication powered by Git as a database.
- Access control, canvas editor, tagging, commenting and more help teams work together efficiently.

Y42

Be part of the Revolution

Sign up for our beta release in 6 weeks

https://y42.com