

L'analytique embarqué avec des technologies data modernes

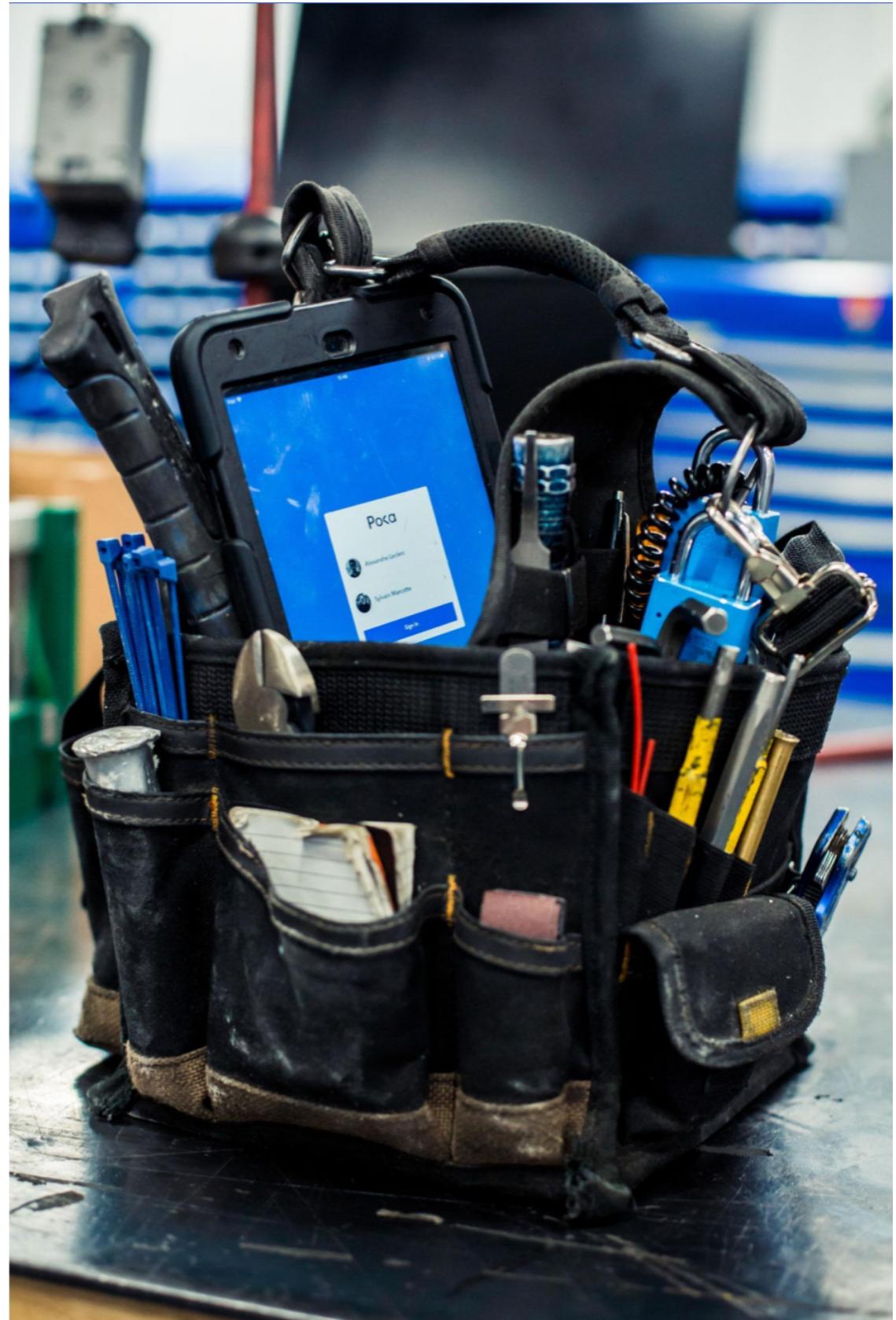
François Vienneau Binette

POKA



QUI

- François, *analytics engineer*
- Poka lancée en analytique en 2020
- 150 personnes, 60 devs, 3 devs data (!)

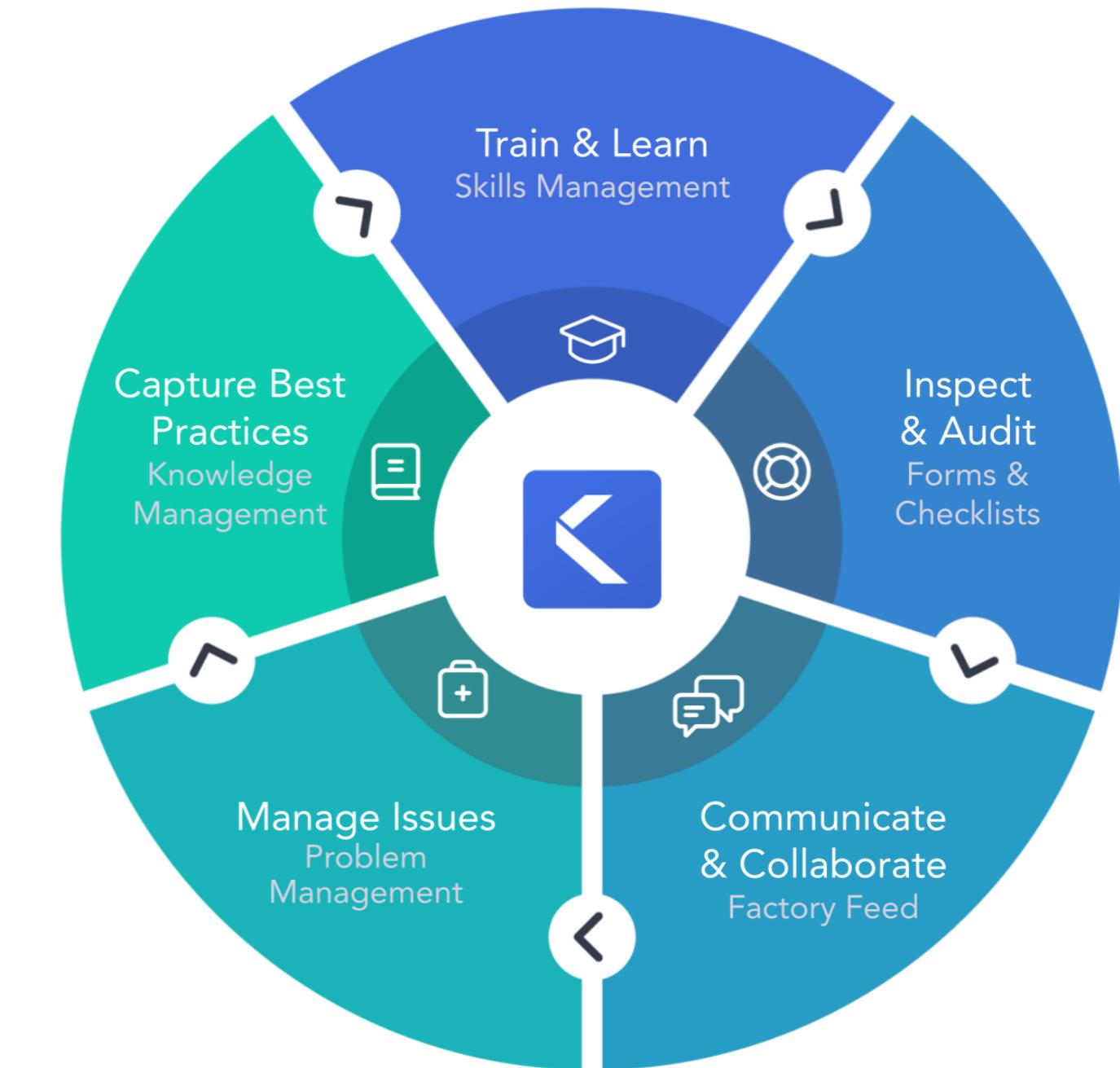


POKA.IO

- Fondée en 2014
- SaaS B2B pour manufacturiers

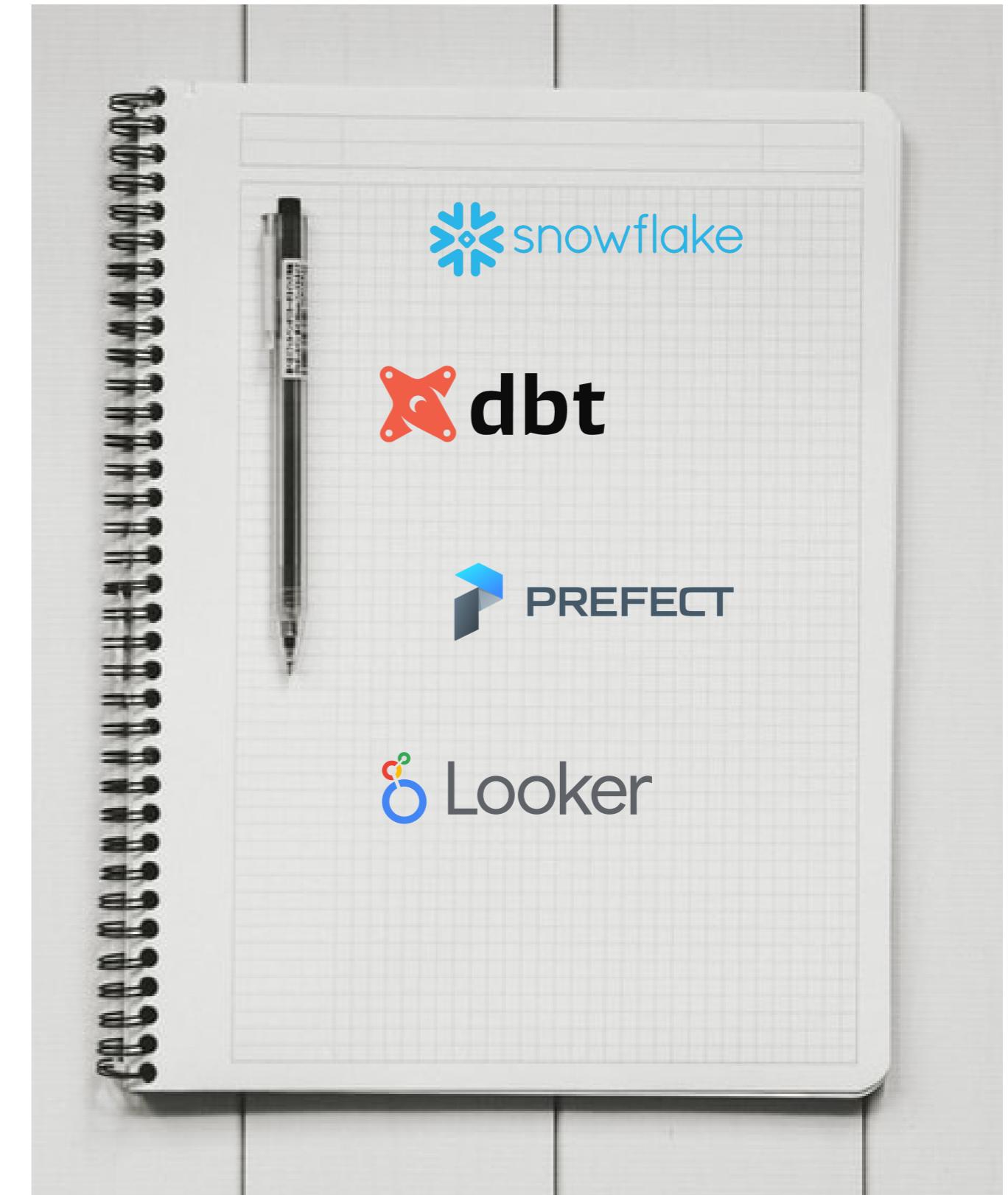
Donner aux travailleurs les connaissances et les outils nécessaires à l'amélioration de la productivité de l'usine: formation, rapports, base de connaissance, communication, etc.

- Web app (Django/Angular), iOS, Android, Windows Store, Blackberry (à venir)
- Travailler avec nous: poka.io/fr/carrieres



AU MENU

1. Pourquoi on fait de l'analytique embarqué
(embedded analytics, embeds)
2. Nos choix technologique & architecture de données
3. Survol de l'intégration dans notre application
4. Démo



AVIS

- Vision simple car beaucoup de détails
- Trop de technologies pour démos
- Travail d'équipe, sur plusieurs mois
- Améliorations possibles
- Fonctionne très bien
- Merci de ne pas me poursuivre



POURQUOI

Pour Poka

- Fonctionnalité (*feature*) intéressante
- Différentiateur vs concurrence
- Échelons de prix (*pricing tiers*)

Pour nos clients

- Favoriser l'adoption
- Forme de contrôle/comparaison
- Intelligence d'affaires accessible

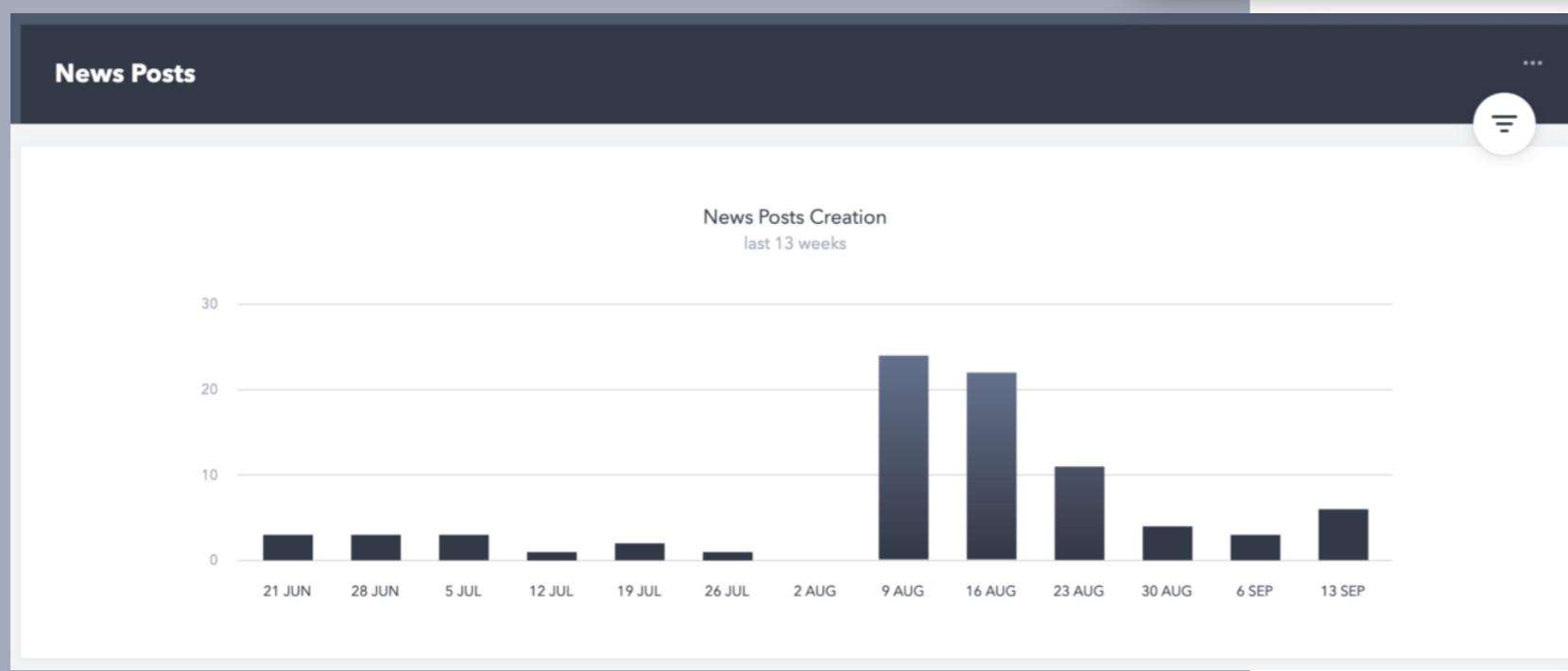
Les *embedded analytics*, c'est à la mode!



Source: <https://www.jeremywinslife.com/cartoon#8>

AVANT

- Géré côté *frontend*
- Développement nécessaire (p.ex., D3.js)
- Difficile de gérer requêtes complexes
- Charge directe sur BD de production
- Peu d'extensibilité, niveau PostgreSQL et niveau humain



User connections

WEEK OF FEB 21, 2022

	MON 21	TUE 22	WED 23	THU 24	FRI 25	SAT 26	SUN 27
DAILY ACTIVE USERS	18	14	0	0	0	0	0

NUMÉRO D'URGENCE - A - 6106 Accountant

Anna Abdou Assistant to the Regional manager

Alex Abdu Design Artist

Kalel Acevedo Packaging Lead

Levi Ackerman Cleaning Team

Alda Ad Packaging Specialist

Miliano Adam Director of Technology

Owen ADAMS

User activities

Factory Feed Training Content

CUMULATIVE UNTIL TODAY

	NEWS POSTS CREATED	CALLS FOR HELP OPENED	VIEWS	COMMENTS	LIKES	ASSIGNMENTS CREATED
NUMÉRO D'URGENCE - A - 6106 Accountant	-	-	-	-	-	-
Anna Abdou Packaging Specialist	-	-	-	1	-	-
Alex Abdu Design Artist	-	-	-	-	-	-
Kalel Acevedo Packaging Lead	-	-	9	-	1	-
Levi Ackerman Cleaning Team	3	1	113	5	-	2
Alda Ad Packaging Specialist	-	-	1	-	-	-
Miliano Adam Director of Technology	-	-	3	-	-	-
Owen ADAMS	7	-	-	2	-	-

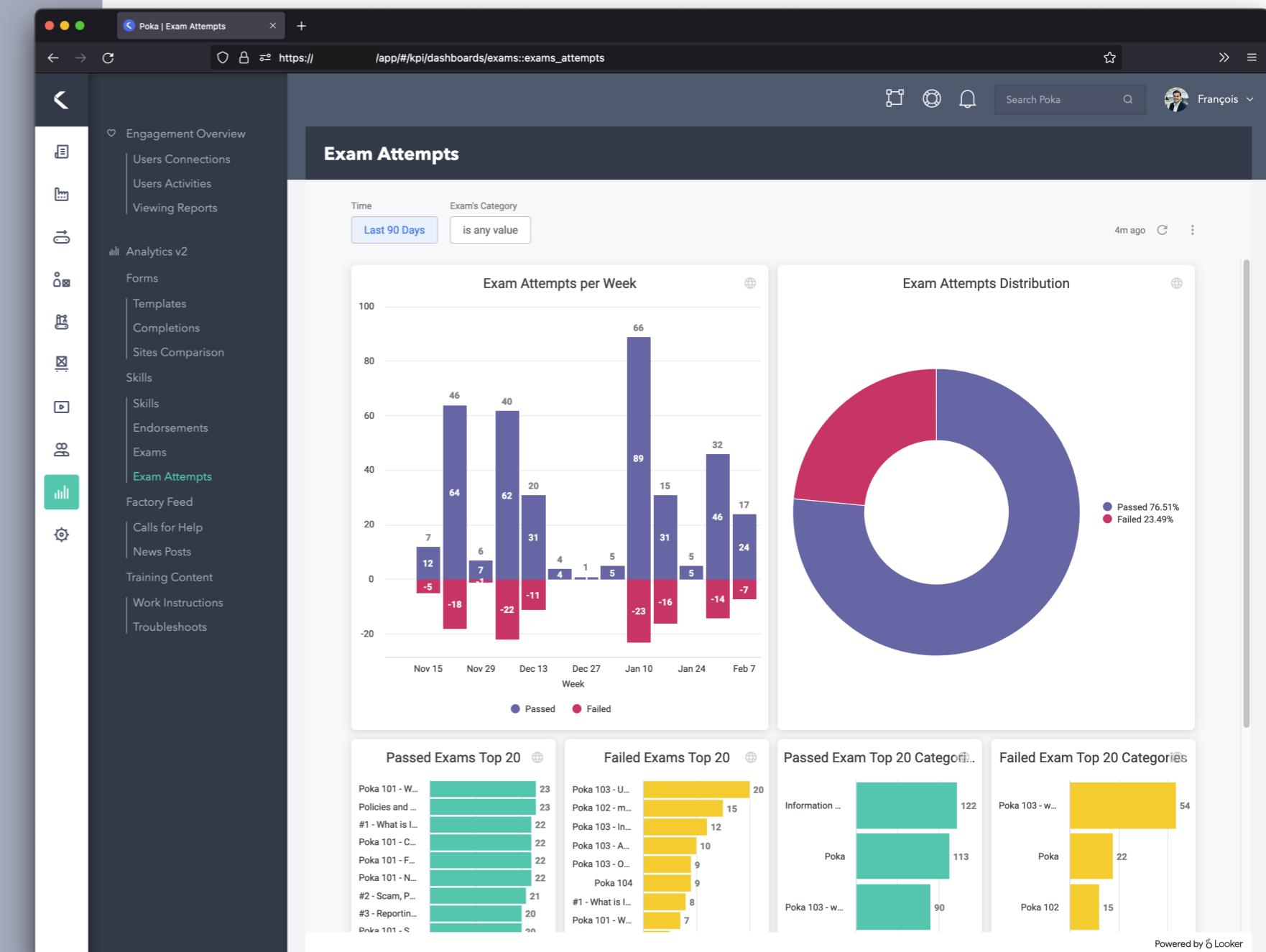
CONTRAINTES

- Autonomie *front/backend*
- Rafraîchissement auto.
- Intégrer meilleures pratiques de dev. (tests, intégration continue, documentation, etc.)
- Extensibilité (*données, dashboards*)
- Assez simple
- Pleinement infonuagique
- Esthétique
- Ne roule pas sur un laptop



APRÈS

- Géré côté équipe analytique
- Développement en SQL & LookML
- Requêtes complexes facilités
- Système analytique indépendant (performance)
- Gestion du code
- Rapidité de traitement
- Assez simple (!)



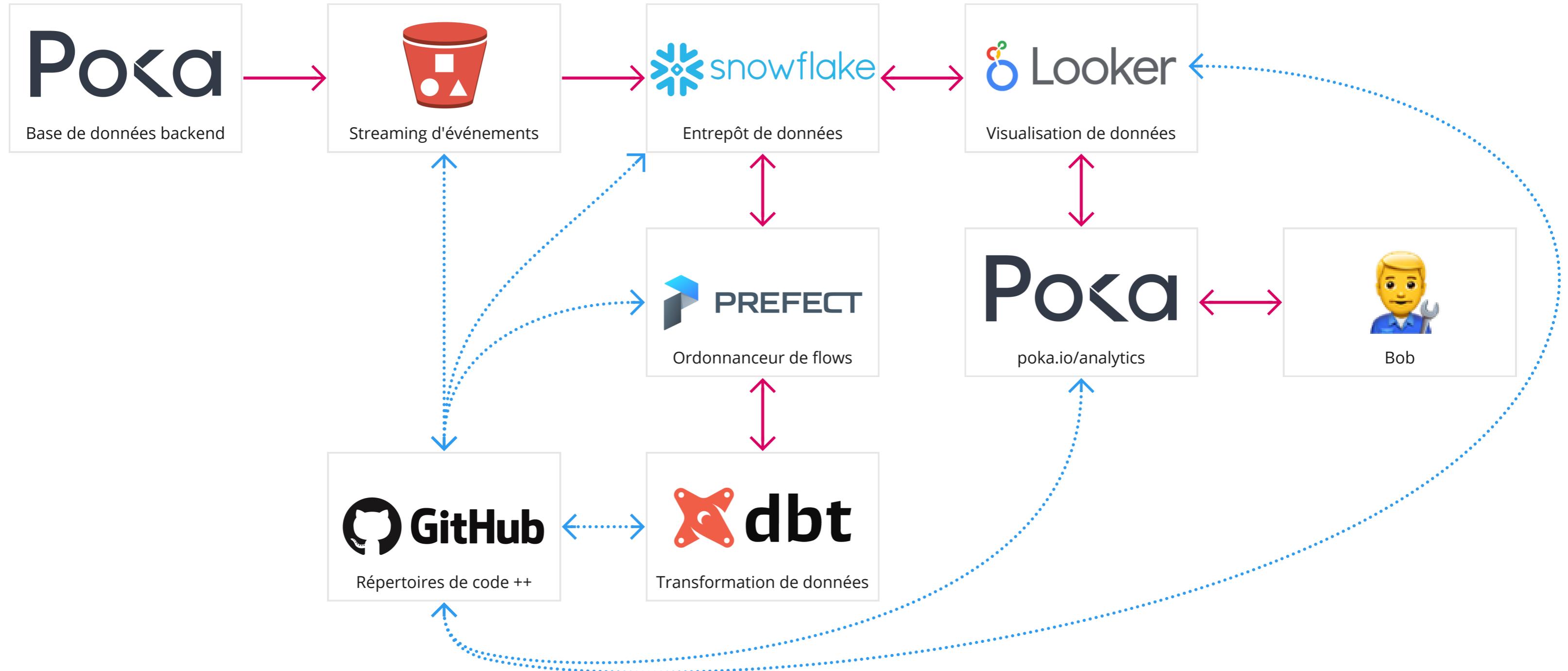
Sources de données

Extraction & chargement

Stockage & transformations

Accès & analyses

Utilisateurs & autres



SNOWFLAKE

- Entrepôt de données infonuagique
- Stocke ensemble données analytiques

Avantages

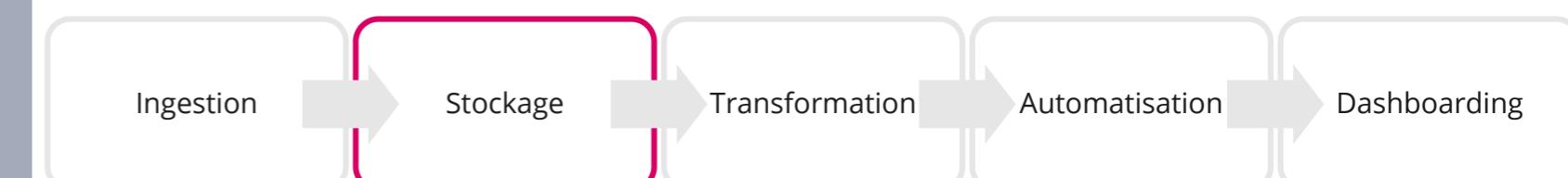
- Pas de maintenance, garantie intégrité
- Cheval de guerre pour les traitements
- Extensible et facile d'utilisation
- Tous nos clients dans même table!

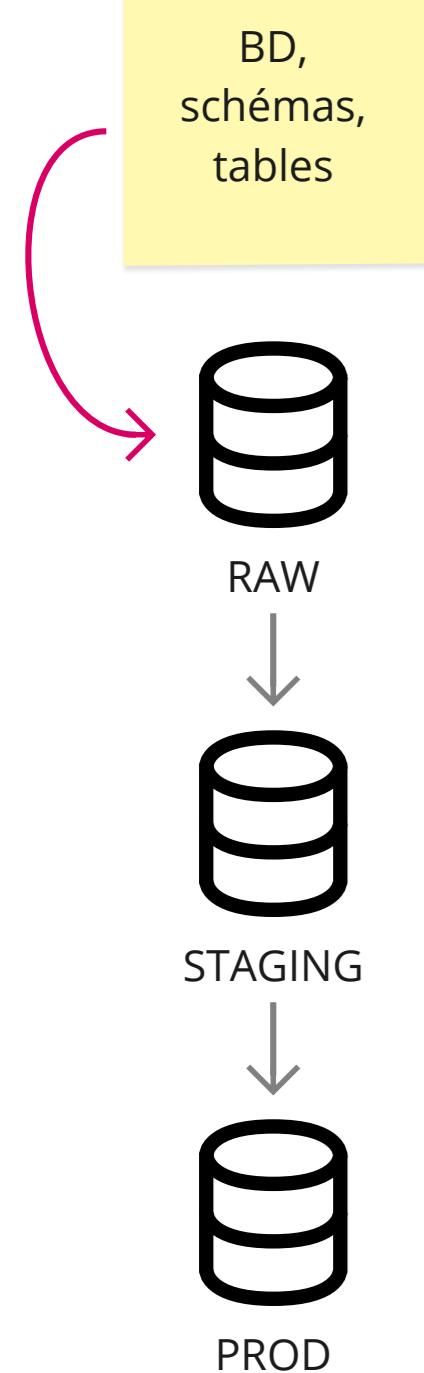
Inconvénients

- Ça coûte un peu cher (25\$/TB + Compute)

Autre options

- Google BigQuery, Amazon Redshift





https://snowflakecomputing.com/console#/internal/worksheet

François Vienneau-Binette
SYSADMIN

COMPUTE_WH (XS)

```

1 select * from "DBT_POKA_DB_PROD"."MART_POKAIO"."DIM_POKAIO_EXAM" limit 100;
  
```

Results Data Preview

Row	TENANT_ID	EXAM_ID	VERSION_NUM	EXAM_CATEGOR	PLANT_ID	IS_GLOBAL	CREATED_DATE	CREATED_BY	MODIFIED_DATE	LAST_MODIFIED	DELETED_DATE	DELETED_BY	EXAM_NAME_LC	PASS_MARK_TYP
1	e169e09a-7...	e169e09a-7...	1.0.0	e169e09a-7...	NULL	TRUE	2021-06-23 ...	e169e09a-7...	2021-06-23 ...	e169e09a-7...	NULL	NULL	{"es": "Exam...}	percentage
2	e169e09a-7...	e169e09a-7...	3.0.0	e169e09a-7...	NULL	TRUE	2021-06-23 ...	e169e09a-7...	2021-06-25 ...	e169e09a-7...	NULL	NULL	{"es": "Exam...}	percentage
3	e169e09a-7...	e169e09a-7...	2.0.0	e169e09a-7...	e169e09a-7...	FALSE	2021-06-24 ...	e169e09a-7...	2021-06-28 ...	e169e09a-7...	NULL	NULL	{"pt": "Serig...}	percentage
4	e169e09a-7...	e169e09a-7...	1.0.0	e169e09a-7...	NULL	TRUE	2021-06-25 ...	e169e09a-7...	2021-06-25 ...	e169e09a-7...	NULL	NULL	{"es": "Exam...}	percentage
5	e169e09a-7...	e169e09a-7...	1.0.0	e169e09a-7...	NULL	TRUE	2021-06-25 ...	e169e09a-7...	2021-06-25 ...	e169e09a-7...	NULL	NULL	{"es": "Exam...}	percentage
6	e169e09a-7...	e169e09a-7...	1.0.0	e169e09a-7...	NULL	TRUE	2021-06-26 ...	e169e09a-7...	2021-06-26 ...	e169e09a-7...	NULL	NULL	{"es": "Exam...}	percentage
7	e169e09a-7...	e169e09a-7...	1.0.0	e169e09a-7...	NULL	TRUE	2021-06-27 ...	e169e09a-7...	2021-06-27 ...	e169e09a-7...	NULL	NULL	{"es": "Exam...}	percentage
8	e169e09a-7...	e169e09a-7...	1.0.0	e169e09a-7...	NULL	TRUE	2021-06-27 ...	e169e09a-7...	2021-06-27 ...	e169e09a-7...	NULL	NULL	{"es": "Exam...}	percentage
9	e169e09a-7...	e169e09a-7...	1.0.0	e169e09a-7...	NULL	TRUE	2021-06-27 ...	e169e09a-7...	2021-06-27 ...	e169e09a-7...	NULL	NULL	{"es": "Exam...}	percentage
10	e169e09a-7...	e169e09a-7...	1.0.0	e169e09a-7...	NULL	TRUE	2021-06-27 ...	e169e09a-7...	2021-06-27 ...	e169e09a-7...	NULL	NULL	{"es": "Exam...}	percentage
11	e169e09a-7...	e169e09a-7...	1.0.0	e169e09a-7...	NULL	TRUE	2021-06-27 ...	e169e09a-7...	2021-06-27 ...	e169e09a-7...	NULL	NULL	{"es": "Exam...}	percentage
12	e169e09a-7...	e169e09a-7...	1.0.0	e169e09a-7...	NULL	TRUE	2021-06-27 ...	e169e09a-7...	2021-06-27 ...	e169e09a-7...	NULL	NULL	{"es": "Exam...}	percentage
13	e1969cf3-5...	e1969cf3-5...	1	e1969cf3-5...	e1969cf3-5...	FALSE	2018-10-16 ...	e1969cf3-5...	2019-06-20 ...	e1969cf3-5...	2018-10-17 ...	e1969cf3-5...	{"fr": "Quest...}	percentage
14	e1969cf3-5...	e1969cf3-5...	1.0.0	e1969cf3-5...	NULL	TRUE	2021-02-10 ...	e1969cf3-5...	2021-02-10 ...	e1969cf3-5...	NULL	NULL	{"en": "Exa...}	percentage
15	e1969cf3-5...	e1969cf3-5...	1	e1969cf3-5...	e1969cf3-5...	FALSE	2018-10-16 ...	e1969cf3-5...	2019-06-20 ...	e1969cf3-5...	2018-10-17 ...	e1969cf3-5...	{"fr": "Qts G...}	percentage
16	e1969cf3-5...	e1969cf3-5...	1.0.0	e1969cf3-5...	e1969cf3-5...	FALSE	2021-11-30 ...	e1969cf3-5...	2021-11-30 ...	e1969cf3-5...	2022-01-06 ...	e1969cf3-5...	{"fr": "Exam...}	percentage

Warehouses

Worksheets

+
Requêtes
SQL pour
TOUT!

DBT

- Transformation dans *ELT*
- Documentation et test sur données

Avantages

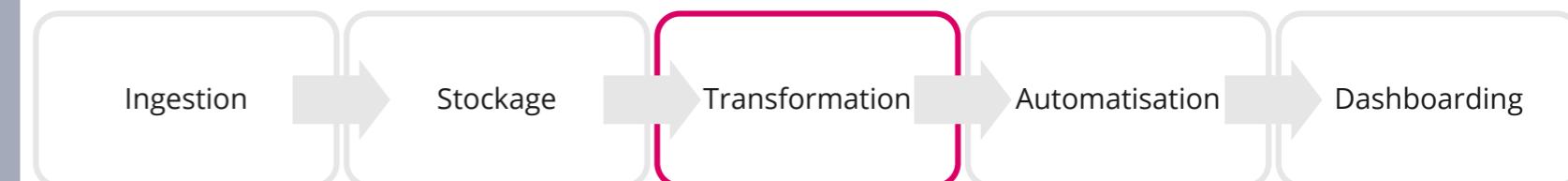
- Accessible: SQL, Jinja (macros), YML (doc)
- Open source (version CLI)
- Option cloud (dbt Cloud)
- GitHub

Inconvénients

- En développement

Autre options

- Talent Open Studio, Airflow, NiFi, Spark
- Outils legacy: SAS, SSIS, IBM DataStage
- LookML (!)



Macros
(SQL)

Liste de
modèles

Fichier de
doc & tests

Modèle en
SQL

```
dim_pokaio_exam.sql — data-dbt

dbt > poka > models > marts > pokaio > exams > dim_pokaio_exam.sql

with source as (
    select * from {{ ref('stg_pokaio_exam') }}
),
renamed as (
    select
        -- Keys
        tenant_id as tenant_id,
        {{ compound_key('EXAM', 'event_data:id') }} as exam_id,
        event_data:version.number::varchar as version_number,
        {{ compound_key('EXAMCATEGORY', 'event_data:category') }} as exam_category_id,
        {{ compound_key('PLANT', 'event_data:plant') }} as plant_id,
        (plant_id is null) as is_global,
        -- Dates
        event_data:created::timestamp as created_date,
        {{ compound_key('USER', 'event_data:added_by') }} as created_by,
        event_data:modified::timestamp as modified_date,
        {{ compound_key('USER', 'event_data:last_modified_by') }} as last_modified_by,
        event_data:deleted::timestamp as deleted_date,
        {{ compound_key('USER', 'event_data:deleted_by') }} as deleted_by,
        -- Data
        event_data:name::variant as exam_name_locale,
        event_data:pass_mark.type::varchar as pass_mark_type,
        event_data:pass_mark.value::number as pass_mark_value,
        event_data:provider_name::varchar as provider_name,
        case
            when deleted_date is not null then 'deleted'
            when timediff(seconds, created_date, modified_date) < 5 then 'created'
            else 'modified'
        end as exam_status
    from source
)
select * from renamed
order by exam_id
```

The screenshot shows a code editor window with a dark theme. The file being edited is `dim_pokaio_exam.sql`. The code is a DBT model definition for a dimension table named `dim_pokaio_exam`. It uses a `with source as` block to select from a reference to a stage table named `stg_pokaio_exam`. The `source` block contains several `select` statements for different columns, including keys, dates, and data. The `data` section includes a `case` statement to determine the status of the record based on the `deleted_date`. The final part of the code is a `select * from renamed` and `order by exam_id`.

Annotations with red arrows point to specific parts of the code:

- An arrow points from the "Macros (SQL)" box to the `dim_pokaio_exam.sql` file in the Explorer sidebar.
- An arrow points from the "Liste de modèles" box to the `dim_pokaio_exam.sql` file in the Explorer sidebar.
- An arrow points from the "Fichier de doc & tests" box to the `! _pokaio_schema.yml` file in the Explorer sidebar.
- An arrow points from the "Modèle en SQL" box to the main code area of the editor.

PREFECT

- Ordonnanceur infonuagique
- Envoi traitements aux bon endroits
- "Roule" nos modèles dbt

Avantages

- Pas de maintenance
- Possibilités illimitées
- Python
- GitHub

Inconvénients

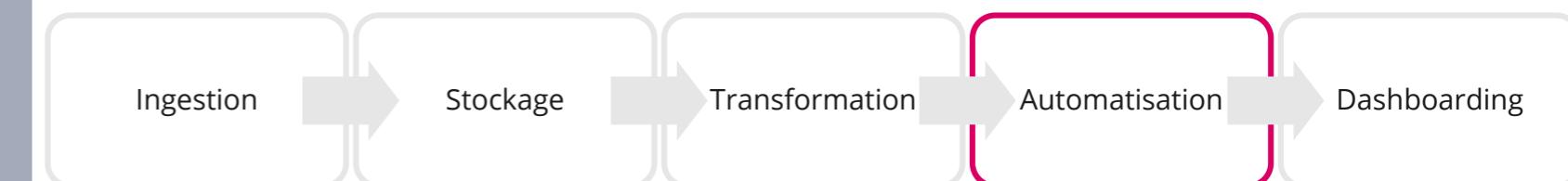
- Limitations version gratuite

Autre options

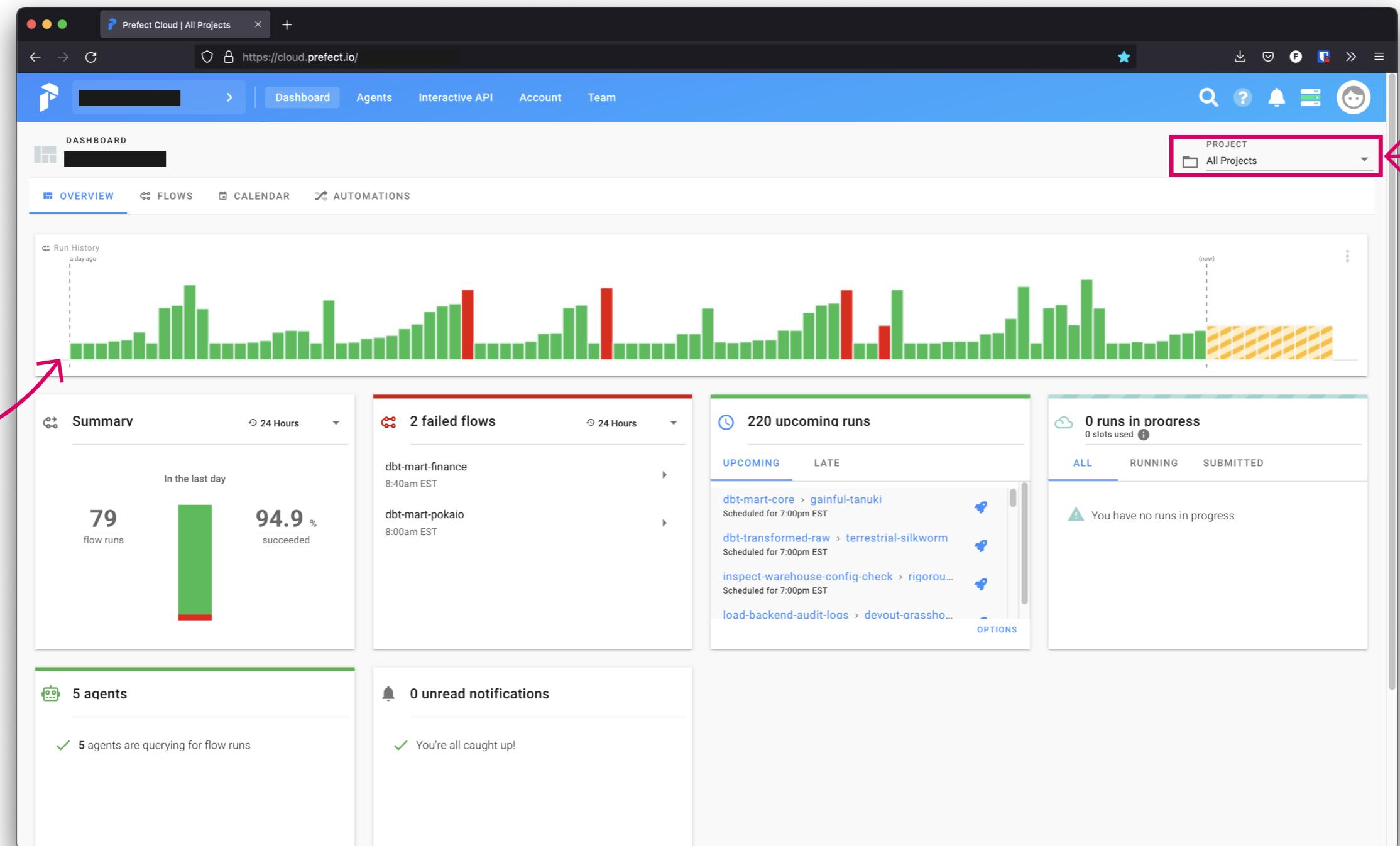
- Apache Airflow, NiFi,



PREFECT



Chronologie
des tâches
(flows)



Projets
PROD, TEST,
DEV

LOOKER

- Visualisation de données & analytique

Avantages

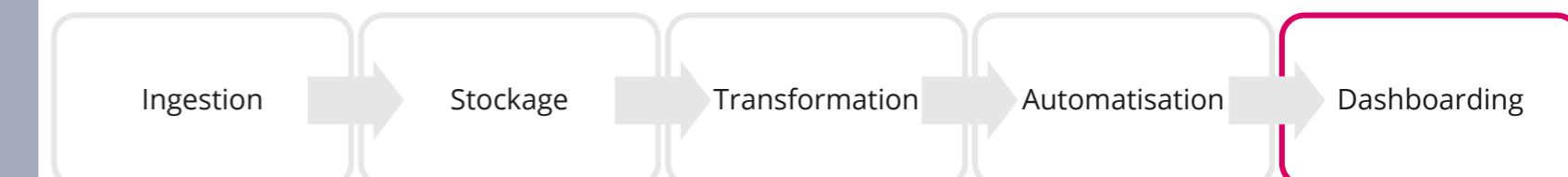
- Pas de maintenance, garantie intégrité
- Intégration hors Looker (*embedding*)
- Traitements faits côté Snowflake
- LookML/*dashboard-as-code*
- GitHub

Inconvénients

- Ça coûte très cher
- LookML, bonne courbe d'apprentissage

Autre options

- Tableau, Qlick, Sisence, Mode, Sigma, etc.
- À la main avec des librairies



Views
(tables)

exam.view

```
view: exam {  
    sql_table_name: @DB_SCHEMA.EMBEDS_EXAMS ;;  
  
    dimension: pk1_exam_id {  
        primary_key: yes  
        hidden: yes  
  
        type: string  
        sql: ${exam_id} ;;  
    }  
  
    dimension: exam_id {  
        label: "Exam"  
        type: string  
        sql: ${TABLE}.exam_id ;;  
    }  
  
    dimension: tenant_id {  
        label: "Tenant"  
        type: string  
        sql: ${TABLE}.tenant_id ;;  
    }  
  
    dimension: exam_name {  
        description: "Exam Name"  
        type: string  
        sql: UDF_EXTRACT_TRANSLATED_TEXT(${TABLE}.exam_name_locale, '@USER_LOCALE') ;;  
    }  
  
    dimension: category {  
        label: "Category"  
        type: string  
        sql: ${TABLE}.category ;;  
    }  
  
    dimension_group: created_date {  
        label: "Created Date"  
        type: time  
        sql: ${TABLE}.created_date ;;  
    }  
  
    dimension: created_by {  
        label: "Created By"  
        type: string  
        sql: ${TABLE}.created_by ;;  
    }  
}
```

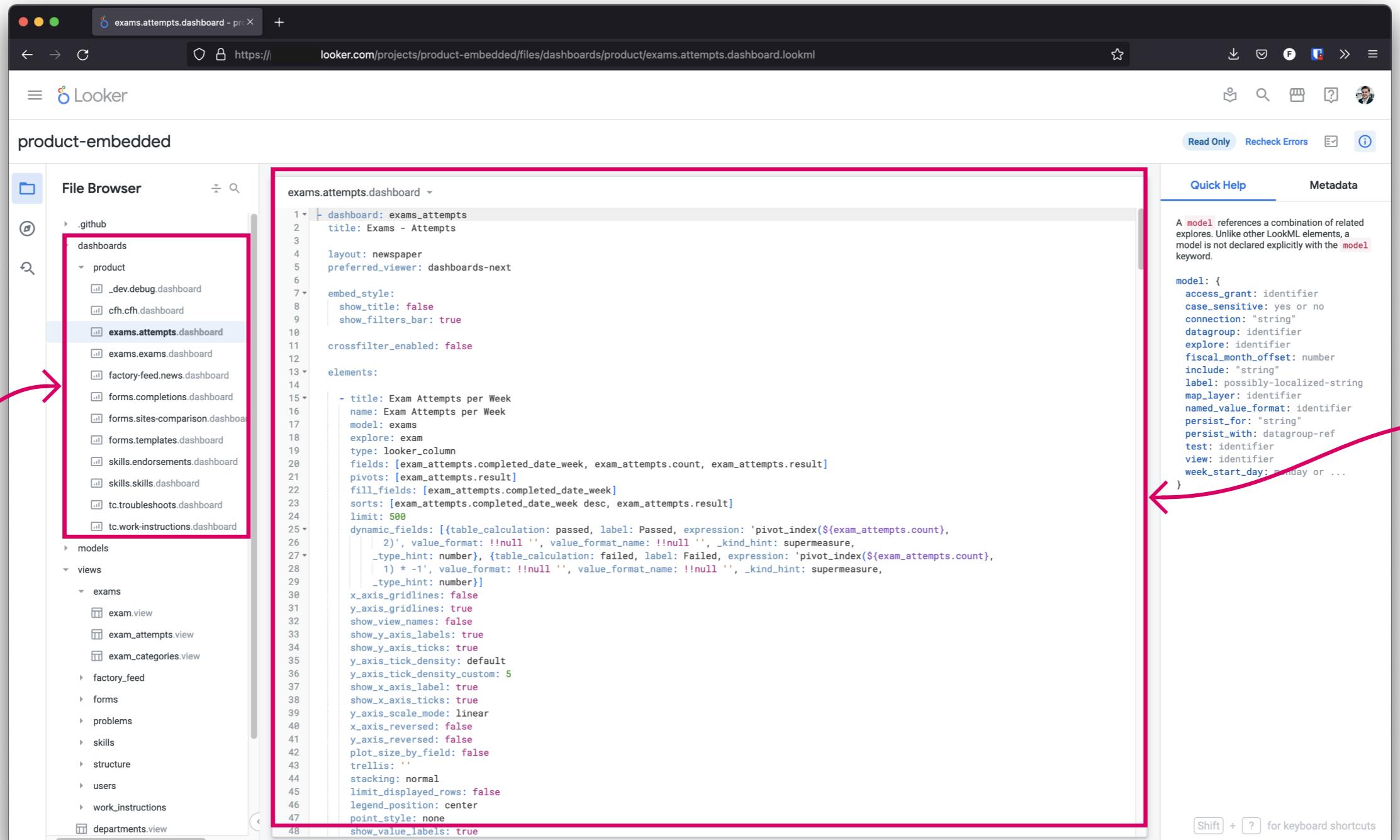
Quick Help Metadata

A **model** references a combination of related explores. Unlike other LookML elements, a model is not declared explicitly with the **model** keyword.

model: {
 access_grant: identifier
 case_sensitive: yes or no
 connection: "string"
 datagroup: identifier
 explore: identifier
 fiscal_month_offset: number
 include: "string"
 label: possibly-localized-string
 map_layer: identifier
 named_value_format: identifier
 persist_for: "string"
 persist_with: datagroup-ref
 test: identifier
 view: identifier
 week_start_day: monday or ...
}

Déclare les
champs en
LookML

Dashboard-as-code



The screenshot shows the Looker Studio interface with the URL <https://looker.com/projects/product-embedded/files/dashboards/product/exams.attempts.dashboard.lookml>. The interface includes a File Browser on the left, a large central code editor, and various navigation and help panels on the right.

File Browser: Shows a tree structure of files and folders. A red box highlights the 'dashboards' folder under '.github/product'. Another red box highlights the 'exams.attempts.dashboard' file in the 'dashboards' folder.

Code Editor: Displays the LookML configuration for the 'exams.attempts.dashboard' file. The code defines a dashboard named 'Exams - Attempts' with a 'newspaper' layout, a 'preferred_viewer' of 'dashboards-next', and an 'embed_style' with 'show_title: false' and 'show_filters_bar: true'. It also specifies 'crossfilter_enabled: false' and a list of 'elements'.

Right Panel: Includes 'Quick Help' and 'Metadata' sections. The 'Quick Help' section provides a detailed explanation of the 'model' keyword. The 'Metadata' section lists various LookML elements and their descriptions.

```
1  dashboard: exams_attempts
2  title: Exams - Attempts
3
4  layout: newspaper
5  preferred_viewer: dashboards-next
6
7  embed_style:
8    show_title: false
9    show_filters_bar: true
10
11  crossfilter_enabled: false
12
13  elements:
14
15    - title: Exam Attempts per Week
16      name: Exam Attempts per Week
17      model: exams
18      explore: exam
19      type: looker_column
20      fields: [exam_attempts.completed_date_week, exam_attempts.count, exam_attempts.result]
21      pivots: [exam_attempts.result]
22      fill_fields: [exam_attempts.completed_date_week]
23      sorts: [exam_attempts.completed_date_week desc, exam_attempts.result]
24      limit: 500
25
26      dynamic_fields: [{table_calculation: passed, label: Passed, expression: 'pivot_index(${exam_attempts.count}, 2)', value_format: '!null ''', value_format_name: '!null ''', _kind_hint: supermeasure, _type_hint: number}, {table_calculation: failed, label: Failed, expression: 'pivot_index(${exam_attempts.count}, 1) * -1', value_format: '!null ''', value_format_name: '!null ''', _kind_hint: supermeasure, _type_hint: number}]
27
28      x_axis_gridlines: false
29      y_axis_gridlines: true
30      show_view_names: false
31      show_y_axis_labels: true
32      show_y_axis_ticks: true
33      y_axis_tick_density: default
34      y_axis_tick_density_custom: 5
35      show_x_axis_label: true
36      show_x_axis_ticks: true
37      y_axis_scale_mode: linear
38      x_axis_reversed: false
39      y_axis_reversed: false
40      plot_size_by_field: false
41      trellis: ''
42      stacking: normal
43      limit_displayed_rows: false
44      legend_position: center
45      point_style: none
46      show_value_labels: true
47
48
```

LookML
dashboard

1

2

3

4

inventory.py

+
Code frontend pour créer la liste

single.py

+
Code frontend pour wrap le iframe

The screenshot shows a web-based dashboard titled "Exam Attempts" from the URL https://poka.io/app/#/kpi/dashboards/exams::exams_attempts. The dashboard features a sidebar on the left with sections like Engagement Overview, Users Connections, Users Activities, Viewing Reports, and Analytics v2. Under Analytics v2, the "Exam Attempts" option is highlighted. The main area displays several charts and tables:

- Chart 1 (Top Right):** A donut chart titled "Exam Attempts Distribution" showing the percentage of passed vs failed attempts. Legend: Passed 76.51% (blue), Failed 23.49% (red).
- Chart 2 (Left):** A bar chart titled "Exam Attempts per Week" showing the number of attempts per week. Data (approximate values): Nov 15: 46 (Passed), -5 (Failed); Nov 29: 40 (Passed), -22 (Failed); Dec 13: 20 (Passed), -11 (Failed); Dec 27: 4 (Passed), 1 (Failed); Jan 10: 66 (Passed), -23 (Failed); Jan 24: 32 (Passed), -14 (Failed); Feb 7: 17 (Passed), -7 (Failed).
- Table 3 (Bottom Left):** "Passed Exams Top 20" showing the top 20 exams with the highest number of passes. Data:

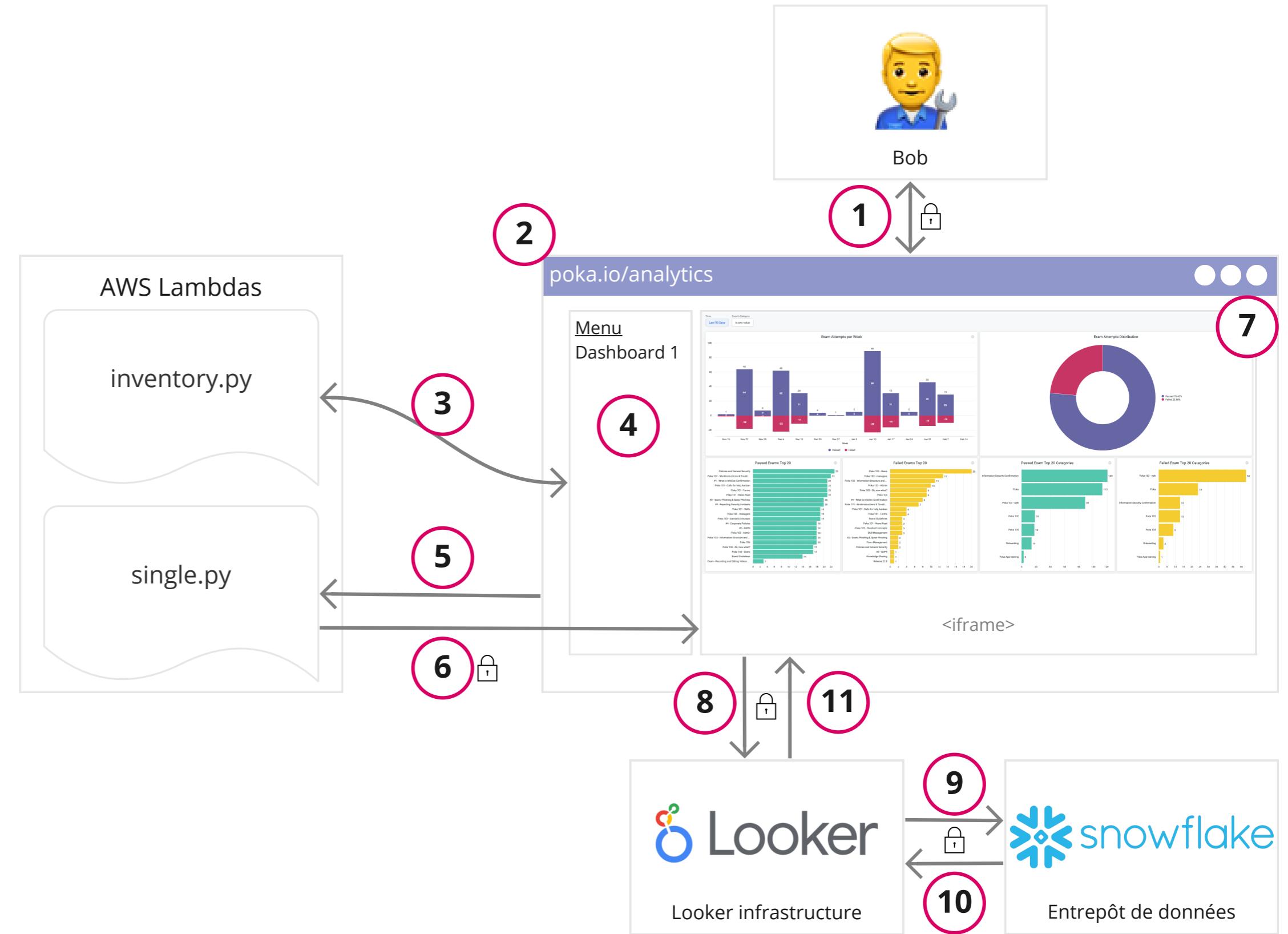
Exam	Passes
Poka 101 - W...	23
Policies and ...	23
#1 - What is I...	22
Poka 101 - C...	22
Poka 101 - F...	22
Poka 101 - N...	22
#2 - Scam, P...	21
#3 - Reportin...	20
Poka 101 - S...	20
- Table 4 (Bottom Middle):** "Failed Exams Top 20" showing the top 20 exams with the highest number of fails. Data:

Exam	Fails
Poka 103 - U...	20
Poka 102 - m...	15
Poka 103 - In...	12
Poka 103 - A...	10
Poka 103 - O...	9
Poka 104	9
#1 - What is I...	8
Poka 101 - W...	7
- Table 5 (Bottom Right):** "Passed Exam Top 20 Categories" showing the top 20 categories with the highest number of passes. Data:

Category	Passes
Information ...	122
Poka	113
Poka 103 - w...	90
- Table 6 (Bottom Far Right):** "Failed Exam Top 20 Categories" showing the top 20 categories with the highest number of fails. Data:

Category	Fails
Poka 103 - w...	54
Poka	22
Poka 102	15

1. Authentification à poka.io
2. Clique sur section Analytics
3. *inventory.py* crée la liste
4. Clique sur un dashboard
5. Backend contacte *single.py*
6. *single.py* retour URL signée
7. Frontend affiche le *iframe*
8. Looker check whitelist DNS
9. Looker requête sur Snowflake
10. Data est retourné à Looker
11. Dashboard rempli de données



DÉMO

EN SOMME

- Indépendance
- Rafraîchissement automatisé
- *Peer-reviewed, CI/CD*
- Extensible
- Dans l'infonuagique
- Ne roule pas sur un *laptop*

Prochaines étapes

- Analyses en libre-service (*Looker Explore*)
- Mise-à-jour



QUESTIONS

François > fvienneaubinette@poka.io

Article sur notre tech blog: <https://medium.com/poka-techblog>