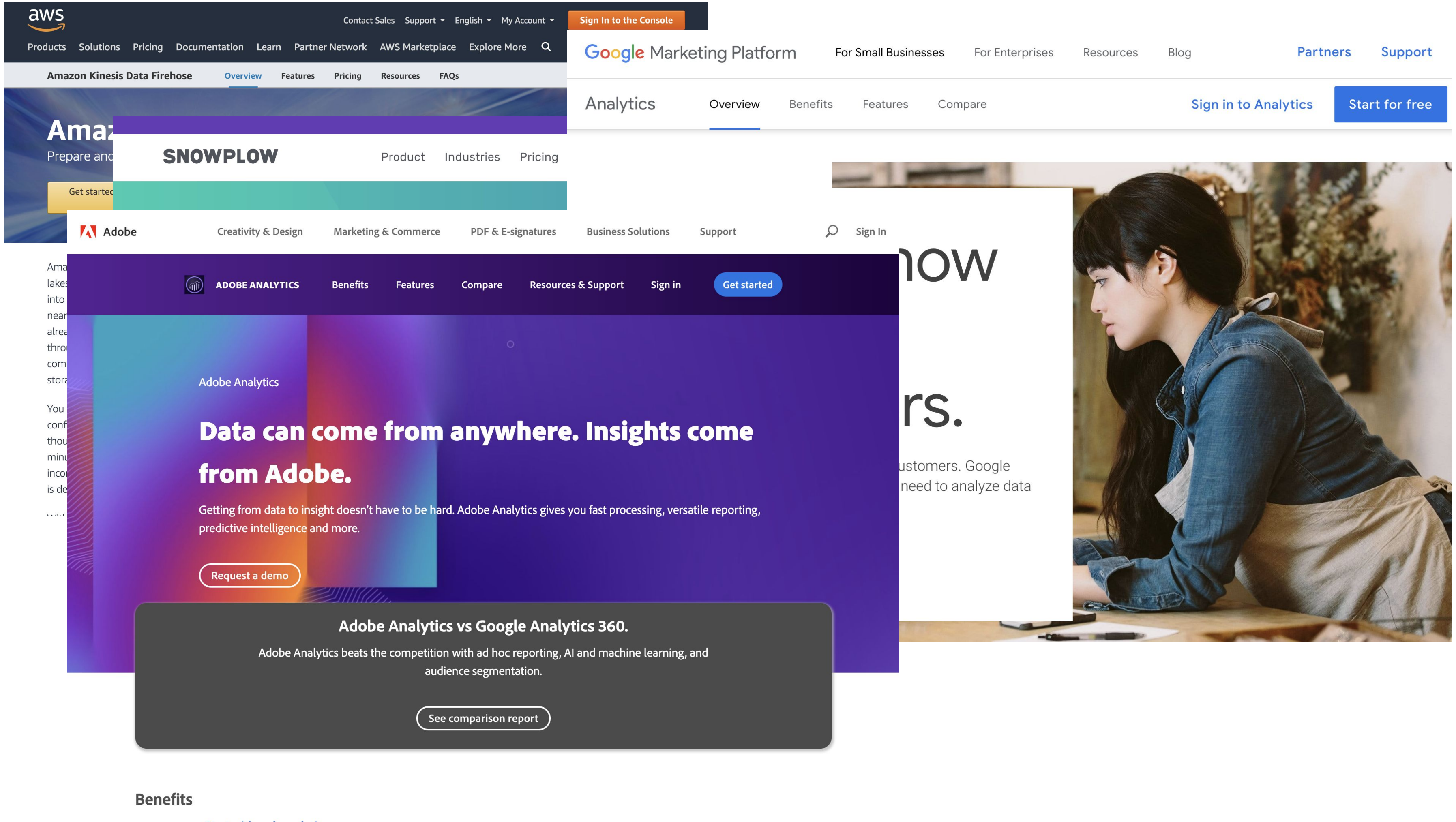
# Data Architecture 101
for Your Business

Bence Faludi - bence@subninja.org

Setting up your **entire data architecture** **should be** straightforward task.

… how **to collect** our frontend data? …



Ama
Prepare and

Ama
lake
into
near
alre
thro
com
stor

You
conf
thou
min
inco
is de

ow

rs.

ustomers. Google
need to analyze data

Adobe Analytics

Data can come from anywhere. Insights come from Adobe.

Getting from data to insight doesn't have to be hard. Adobe Analytics gives you fast processing, versatile reporting, predictive intelligence and more.

Adobe Analytics vs Google Analytics 360.

Adobe Analytics beats the competition with ad hoc reporting, AI and machine learning, and audience segmentation.

Benefits

… **which engine** should we use? …

HIVE

presto

SQL

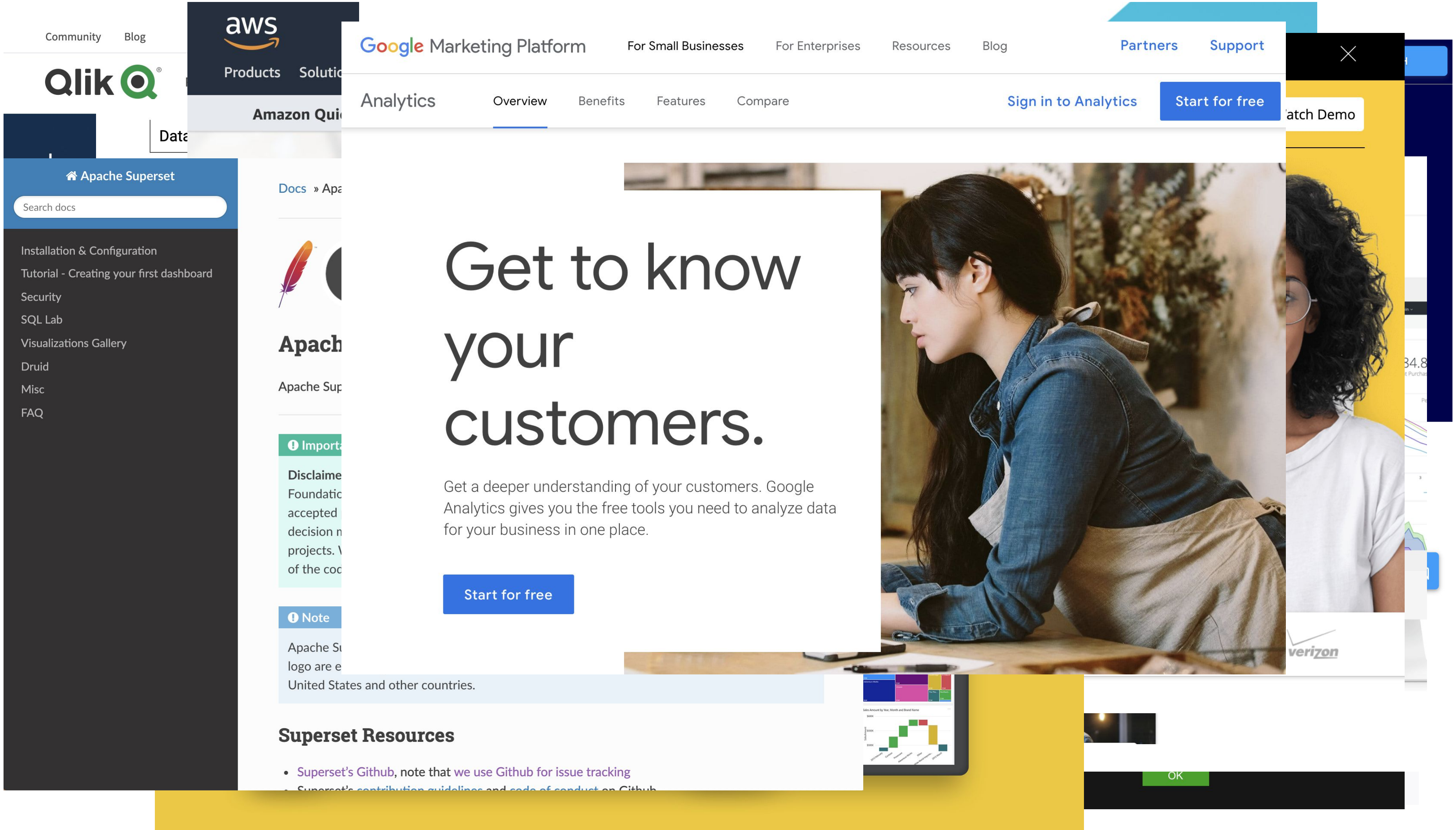
amazon
REDSHIFT

APACHE
Spark™

Azure Cosmos DB

VERTICA

... or just pick a **visualisation** tool ...



Community Blog

Qlik

Data

aws

Products Solutio

Amazon Qui

Apache Superset

Search docs

Installation & Configuration

Tutorial - Creating your first dashboard

Security

SQL Lab

Visualizations Gallery

Druid

Misc

FAQ

Docs » Apa

Apach

Apache Sup

Importa

Disclaime
Foundatic
accepted
decision n
projects. V
of the coo

Note

Apache Su
logo are e
United States and other countries.

Superset Resources

- Superset's Github, note that we use Github for issue tracking

Google Marketing Platform For Small Businesses For Enterprises Resources Blog Partners Support

Analytics Overview Benefits Features Compare Sign in to Analytics Start for free

Get to know your customers.

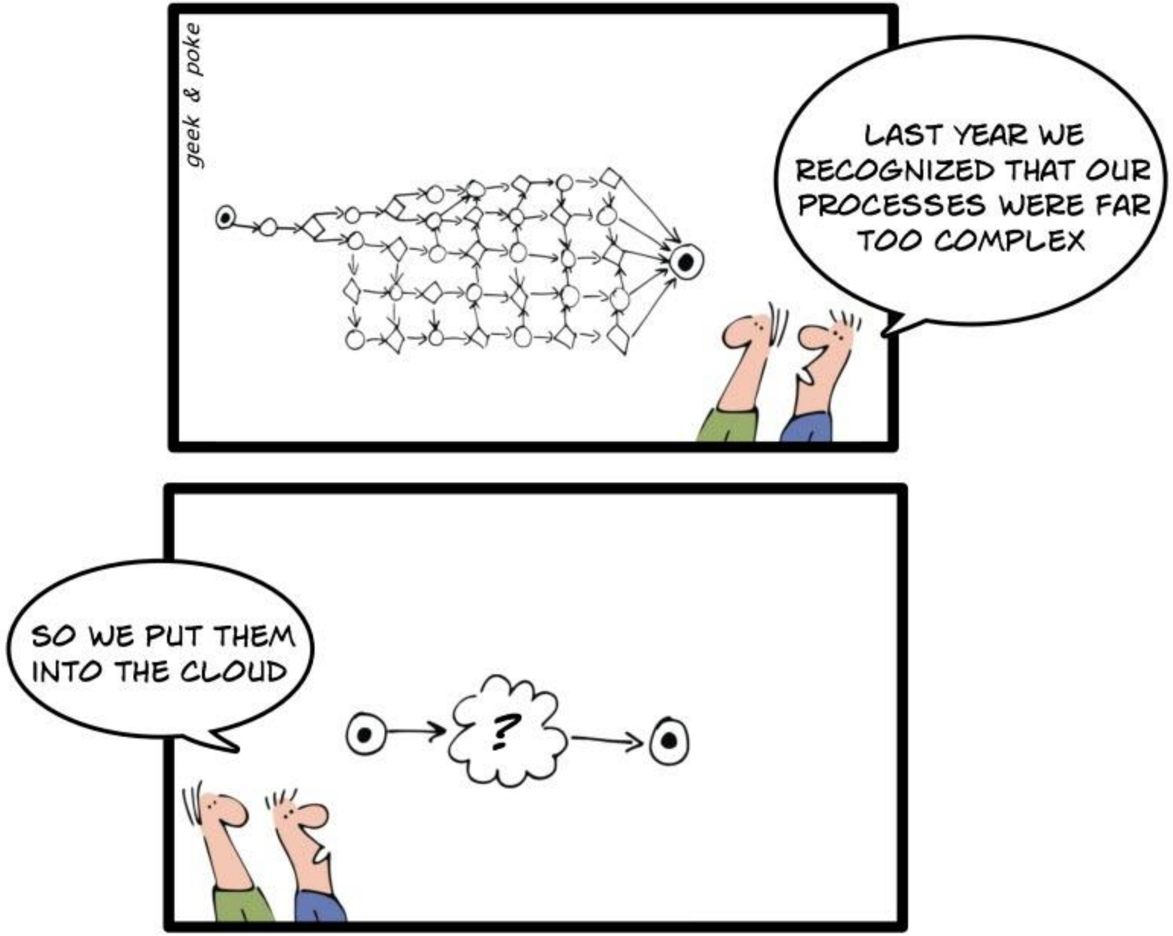
Get a deeper understanding of your customers. Google Analytics gives you the free tools you need to analyze data for your business in one place.

Start for free

Watch Demo

verizon

OK


geek & poke
LAST YEAR WE RECOGNIZED THAT OUR PROCESSES WERE FAR TOO COMPLEX
SO WE PUT THEM INTO THE CLOUD
?

LET THE CLOUDS MAKE YOUR LIFE EASIER

Let’s just be **realistic** and **bullshit-free**!

Big ~~Data~~ Mess

1. **Too many products** are available. Most of them claim they solve all data problems your company encounter; and deliver immediate insights and business value. But NONE is true.
2. Organisational data is mostly **unstructured and not clean**. It is not ready for consumption.
3. Companies are using **various data sources parallelly** but rarely investing in centralisation - and want this behaviour from 3rd party tools.
4. Easy to **stuck with a bad, inefficient and costly architecture**. It's hard, slow and expensive to get rid of them afterwards and clean up the hacks.

Bence Faludi

Independent contractor

bence@subninja.org

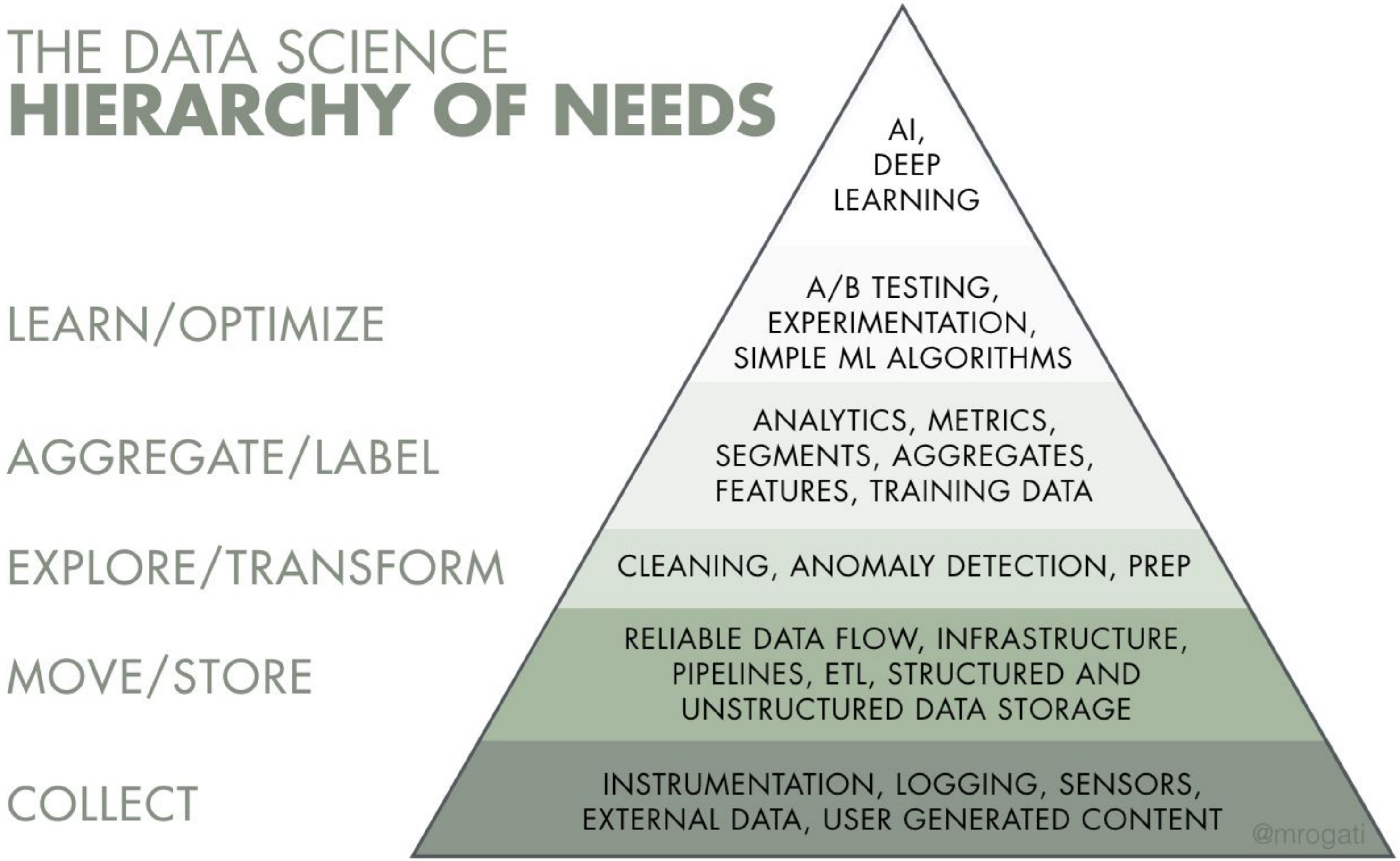
My background

- **Contractor** for various companies.
- Data Engineer at **Facebook**
- Data Scientist at **Microsoft**
- Data Engineer at **Wunderlist**
- etc...

Worked with data sizes from few 1000s to billions of active users.

Contributor of **night-shift** and **metl** open source ETLs.

THE DATA SCIENCE **HIERARCHY OF NEEDS**



AI, DEEP LEARNING
LEARN/OPTIMIZE
A/B TESTING, EXPERIMENTATION, SIMPLE ML ALGORITHMS
AGGREGATE/LABEL
ANALYTICS, METRICS, SEGMENTS, AGGREGATES, FEATURES, TRAINING DATA
EXPLORE/TRANSFORM
CLEANING, ANOMALY DETECTION, PREP
MOVE/STORE
RELIABLE DATA FLOW, INFRASTRUCTURE, PIPELINES, ETL, STRUCTURED AND UNSTRUCTURED DATA STORAGE
COLLECT
INSTRUMENTATION, LOGGING, SENSORS, EXTERNAL DATA, USER GENERATED CONTENT
@mrogati

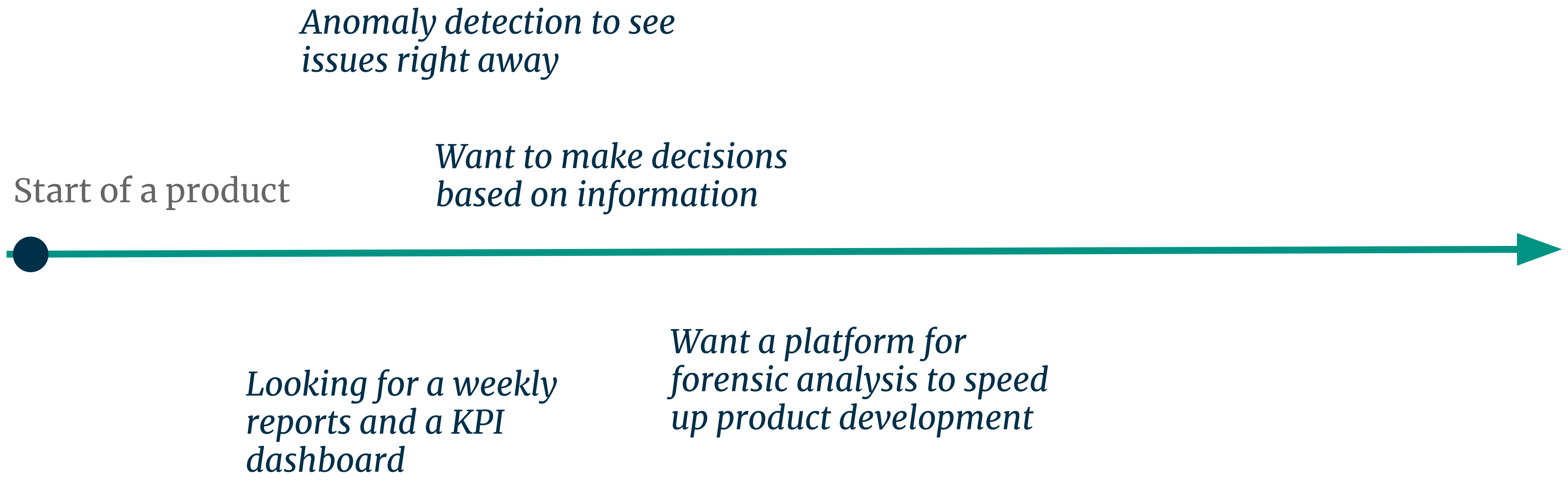
Many tools aim to **merge these layers** and **make it elective**.

Questions to always ask

1. Do you handle **unclean data**?
2. **How quick** will all those transformations and queries be?
3. Where is the **cache stored**?
4. Does it **affect the performance** of our database by running parallel, or inefficient queries?
5. What do we need to prepare to make the product effective?
6. How big exactly is the **data loss of the tracker**?
7. Can we **export the data model** we made within the system?

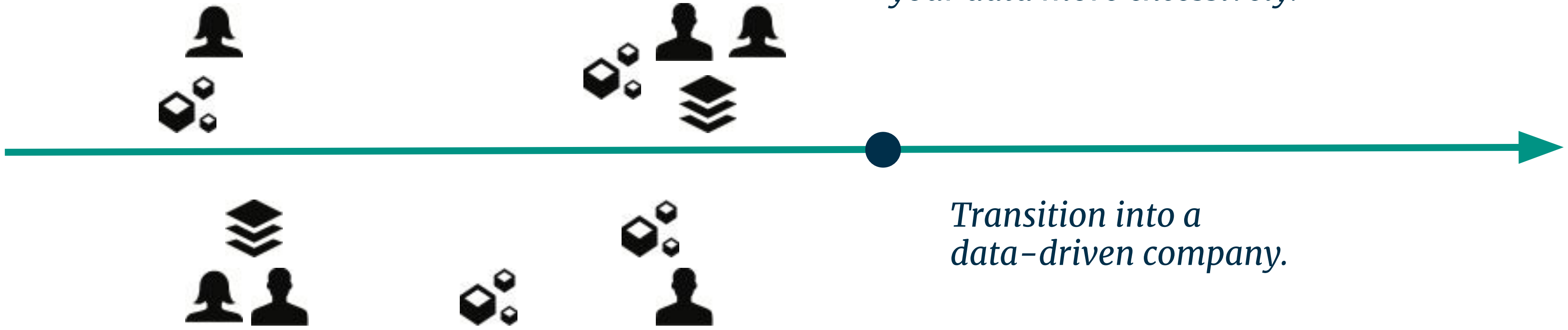
Never believe in **hype**, or **shiny web pages** when selecting products. They want to lock you into their ecosystem.

What can we do?


Anomaly detection to see issues right away
Start of a product
Want to make decisions based on information
Looking for a weekly reports and a KPI dashboard
Want a platform for forensic analysis to speed up product development

Just about to start your business

1. **Evaluate the best stack to use**, and select products that work together.
2. **Focus on centralisation** from the beginning. Raw data access is a key.
3. Make sure you **design all events** from the backend and the frontend that are needed for your KPIs. You don’t want to measure everything!
4. **Step by step**, don’t shoot for ML when you don’t even have proper logging or aggregations.
5. You **only need Data Engineers**. Data Scientist can join later when the leg work will be done.



Recognise you want to use your data more excessively.

Transition into a data-driven company.

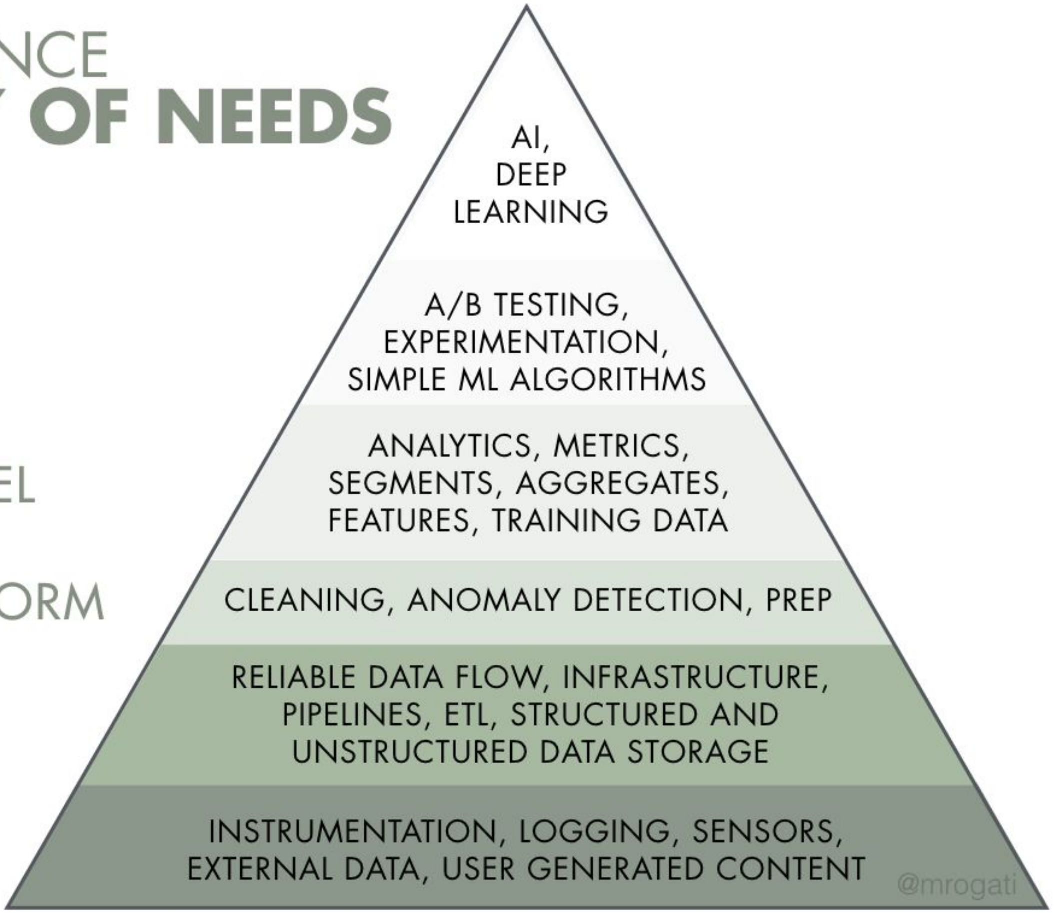
You are road-blocked by your current setup and looking for new opportunities / improvements.

Changing your ongoing business

1. Do not be afraid moving away from your current way of doing things but **select solutions that will not lock you in**. Buying a new product will not solve your issues, it will just make larger mess.
2. Prepare for a long journey. **Ship wins step by step** by migrating over already existing services and enabling new (previously not possible) functionality.
3. **Centralise data sources** into a single Data Warehouse instead of using visualisation tools that can “connect to everything”.
4. Train your Data Scientists and Data Analytists to **use SQL**.

What does a good data architecture provide?

THE DATA SCIENCE **HIERARCHY OF NEEDS**


LEARN/OPTIMIZE
AGGREGATE/LABEL
EXPLORE/TRANSFORM
MOVE/STORE
COLLECT
AI, DEEP LEARNING
A/B TESTING, EXPERIMENTATION, SIMPLE ML ALGORITHMS
ANALYTICS, METRICS, SEGMENTS, AGGREGATES, FEATURES, TRAINING DATA
CLEANING, ANOMALY DETECTION, PREP
RELIABLE DATA FLOW, INFRASTRUCTURE, PIPELINES, ETL, STRUCTURED AND UNSTRUCTURED DATA STORAGE
INSTRUMENTATION, LOGGING, SENSORS, EXTERNAL DATA, USER GENERATED CONTENT
@mrogati

Data collection

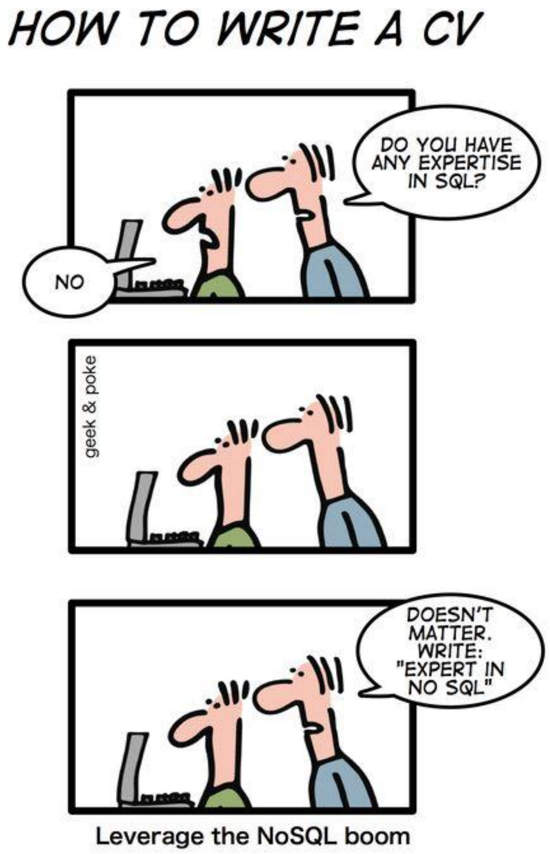
- **Ownership and access of data**: we own the data and the data access is not bounded to active subscriptions.
- **Near-real time raw data**: have access to unfiltered raw data within minutes.
- **No data sampling**: all incoming data is queryable and not just a subset.
- **Ad blockers**: Ad blockers responsible for many lost events. Keep in mind.
- **Personal Identification Information**: possible to turn off PII scraping
- **Data model**: custom events can be sent in not-flat format (e.g.: JSON)
- **SDKs with persistent layer**: collected logs are stored on the offline device.

Storage and flow

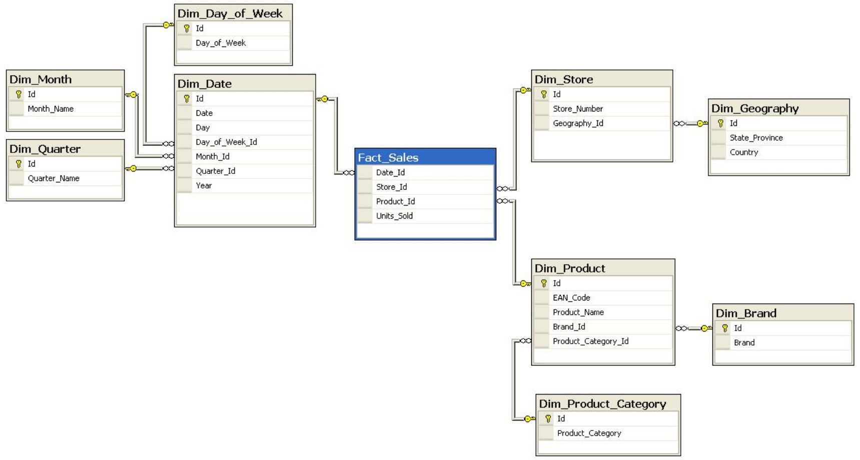
- **Schedulable pipelines with dependencies:** pipelines run at specified times after all dependencies met. Provides notifications, alerts, logging, SLAs, and easily extendable with other sources.
- **Collected data transformation** as minimal as it can be, and the rest can be done within the database engine.
- **Raw level data** stored on the storage, but accessible via the query engine.
- **Data Partitions** are possible to reduce the size of queried data, and keep previous versions of cumulative or event based tables.
- **Data Reprocess**: recrunch numbers any time when business goals change or to fix errors.

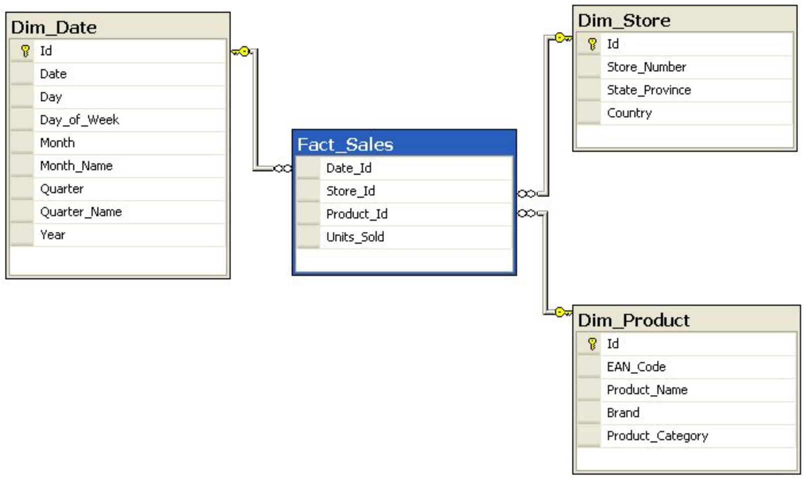
Database/Query engine

- Read **Benchmarks** about performance and pricing. Do the math…
- Look for **distributed query engines**.
- **SQL compatibility**, you don't want to learn something new.
- Support of **complex data-types**, and **CUBE** grouping make your life easier. **Partitions** are key.
- Encryption and compression matter.


HOW TO WRITE A CV
DO YOU HAVE ANY EXPERTISE IN SQL?
NO
geek & poke
DOESN'T MATTER. WRITE: "EXPERT IN NO SQL"
Leverage the NoSQL boom

Snowflake vs Star Schema



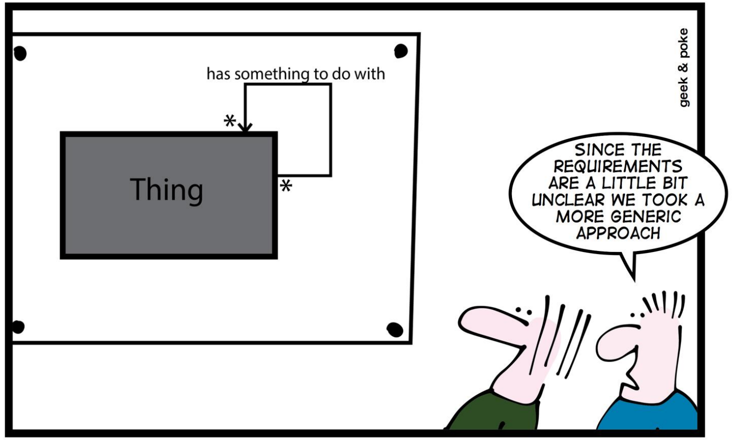


* pictures from wikipedia

Data model

Snowflake schema and **Star schema** use dimension tables to describe data aggregated in a fact table, but dimension tables are denormalised in snowflake schema.

- **Star schema is better for analytics**: reduce query complexity, and speed up queries
- **Flat truth-tables**: enable quick overview for your business units by making flat tables containing your criterias. Partition it wisely.
- Store your **aggregations as Cubes**.


has something to do with
*
*
Thing
SINCE THE REQUIREMENTS ARE A LITTLE BIT UNCLEAR WE TOOK A MORE GENERIC APPROACH
geek & poke

HOW TO CREATE A STABLE DATA MODEL

Prepare your aggregations and metrics

You can even **do this within SQL**, no need for fancy visualisation tools. **Materialise your metrics** for all required dimensions to load your dashboards as fast as possible.

```
-- Grouping sets
SELECT Country, Region, SUM(Sales) AS TotalSales
FROM Sales
GROUP BY GROUPING SETS ( ROLLUP (Country, Region), CUBE (Country, Region) );

-- Cubes
SELECT Country, Region, SUM(Sales) AS TotalSales
FROM Sales
GROUP BY CUBE (Country, Region);
```

* Azure SQL Data Warehouse SQL examples

Visualisation

- **Self-hosted** vs **Hosted**
- Support **native SQL execution** for advanced users - they make the complex reports, and they don’t want to struggle with crappy and limited interfaces.
- Provide **interactive query builder** for beginners - you can’t expect everyone to be a SQL magician, and you want people to drill-down into specific reports and findings.
- **No middle-layer modeling language**: it’s a pain to debug, and learn. SQL is still the most efficient and most widely known option.

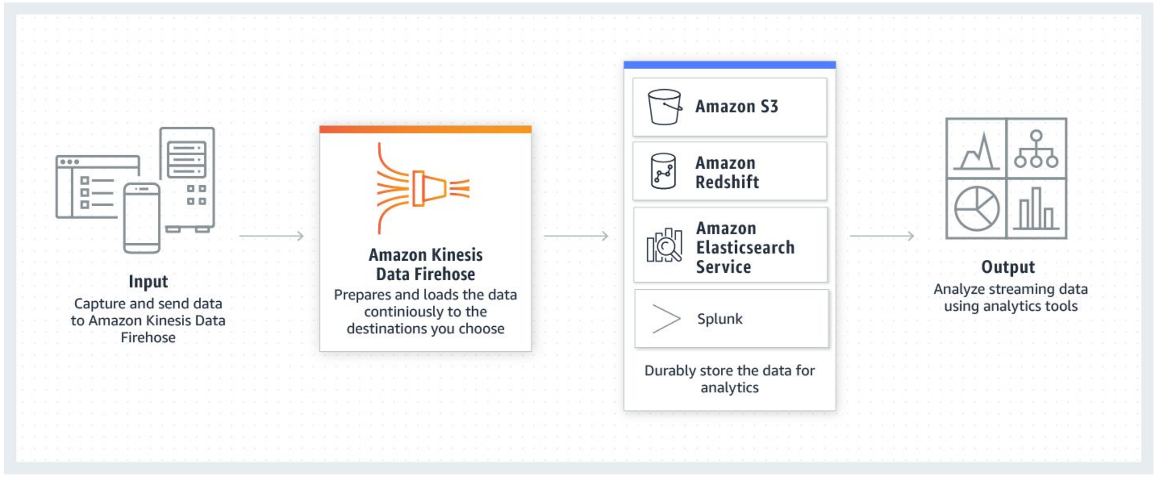
+ anything your business needs (email reports, warnings, public access, etc.).

Example of compatible data architecture stack on AWS

Collect	Amazon Kinesis Data Firehose
Data storage	Amazon S3
Data flow / ETL	Apache Airflow
Database / Query engine	Amazon EMR - Presto (Amazon Athena for large jobs)
Visualisation tool	Apache Superset

Amazon Kinesis Data Firehose to S3

It captures and loads data in near real-time. It **loads data into Amazon S3 within a minute** after data sent to the server. It provides SDKs for Android, iOS, Web (via Amplify JS), and you can integrate backend services as well.

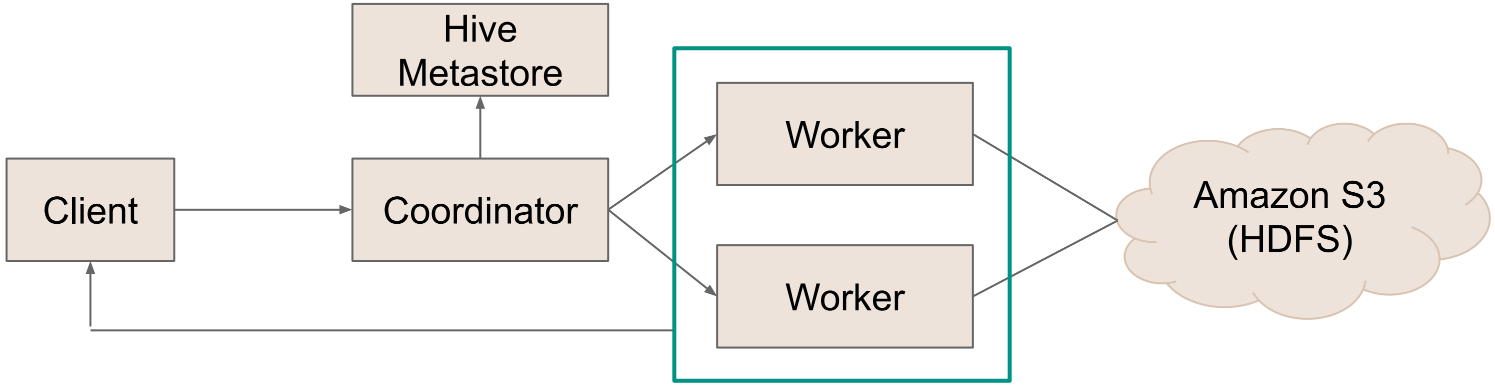

Input
Capture and send data to Amazon Kinesis Data Firehose
Amazon Kinesis Data Firehose
Prepares and loads the data continiously to the destinations you choose
Amazon S3
Amazon Redshift
Amazon Elasticsearch Service
Splunk
Durably store the data for analytics
Output
Analyze streaming data using analytics tools

* picture from AWS website

Amazon EMR – Presto

“Presto is a **distributed SQL query engine** designed to query large data sets distributed over one or more heterogeneous data sources.”

Presto is a distributed system that runs on Hadoop. We will use **Amazon S3** to store, and query data directly. All incoming data will be accessible by the querying engine immediately after the data was written onto S3. It’s quicker, and cheaper than Amazon Redshift.



Amazon EMR – Presto

Presto supports **lambda functions**, **cubes**, **grouping sets**, various **analytical functions**, and **complex data-types** such as maps, and arrays.

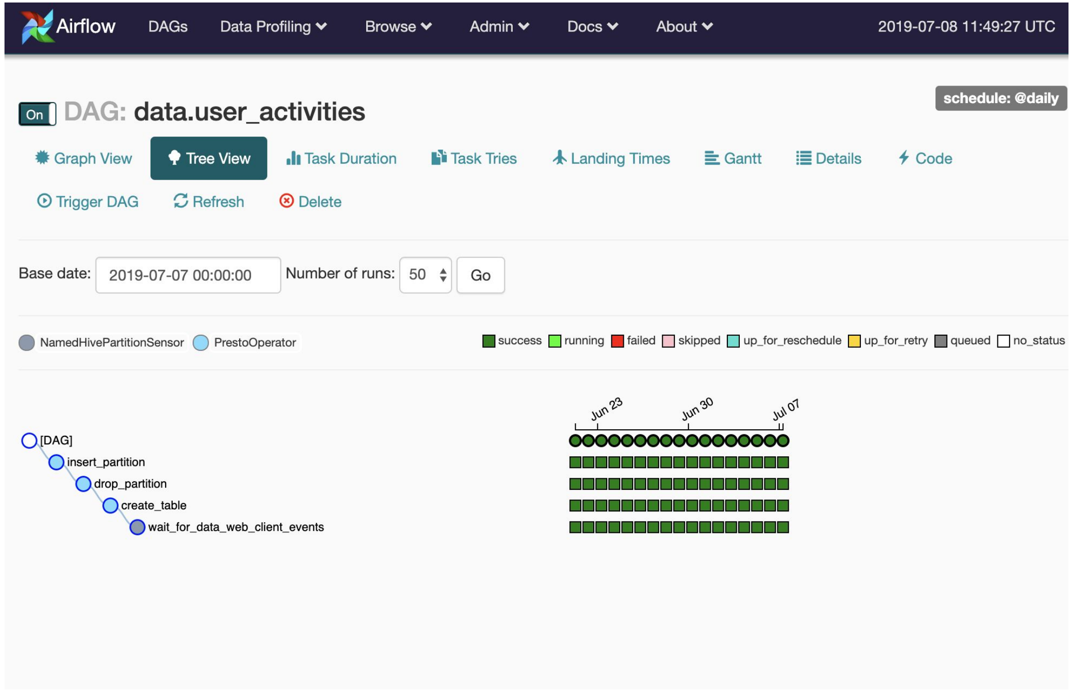
```
SELECT
  GROUPING(u.platform) AS "grouping_id",
  MAP_FROM_ENTRIES(ARRAY[('platform', u.platform)]) AS "dimensions",
  CAST(COUNT(DISTINCT u.device_id) AS DOUBLE) AS "devices",
  CAST(SUM(COUNT(DISTINCT u.device_id)) OVER (
    PARTITION BY GROUPING(u.platform)
  ) AS DOUBLE) AS "total_devices",
FROM data.user_activities AS u
WHERE u.ds = '{{ ds }}'
GROUP BY GROUPING SETS ((), (u.platform))
```

Apache Airflow

Airflow is a Python-based **workflow management framework** to automate scripts in order to perform tasks.

It’s extendable, and provides a good monitoring interface - but quite complex.

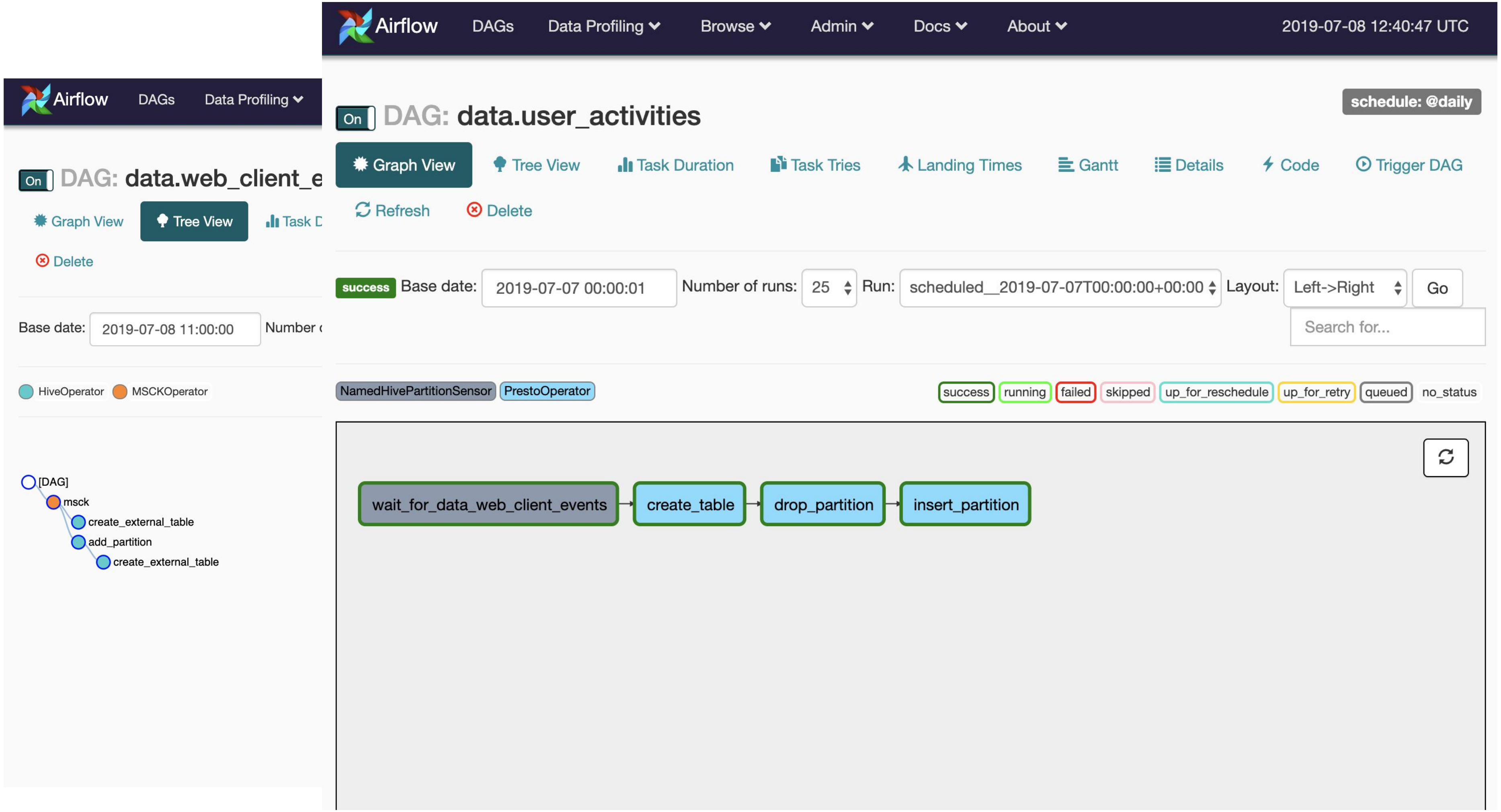
Use **night-shift** instead for maximum simplicity.



DAG: data.user_activities

Base date: 2019-07-07 00:00:00 Number of runs: 50

[DAG]
insert_partition
drop_partition
create_table
wait_for_data_web_client_events



2019-07-08 12:40:47 UTC

DAG: data.user_activities

schedule: @daily

success Base date: 2019-07-07 00:00:01 Number of runs: 25 Run: scheduled__2019-07-07T00:00:00+00:00 Layout: Left->Right

NamedHivePartitionSensor PrestoOperator

wait_for_data_web_client_events → create_table → drop_partition → insert_partition

DAG: data.web_client_e

Base date: 2019-07-08 11:00:00 Number

HiveOperator MSCKOperator

```
[DAG]
  msck
    create_external_table
    add_partition
      create_external_table
```

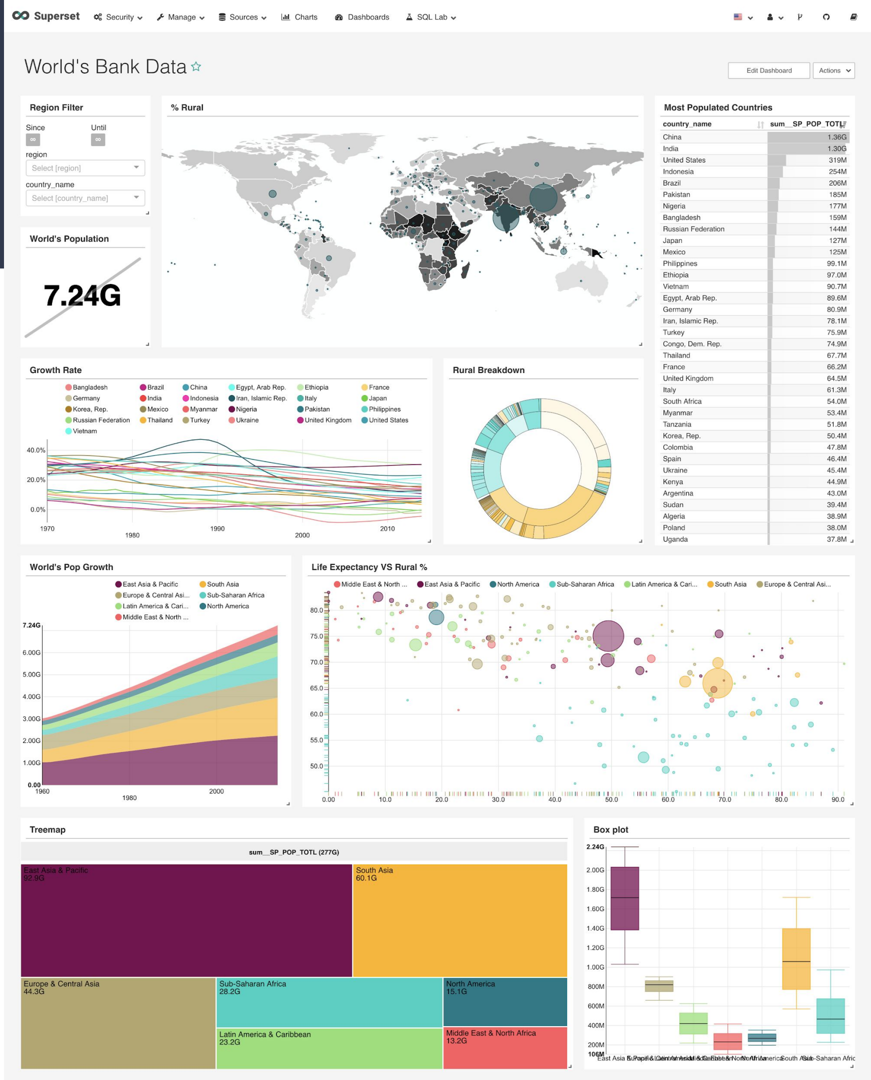
Apache Superset

It’s a free, **self-hosted visualisation tool**, with an easy-to-use interface for exploring data.

Perfect for **collaboration between teams** - and to save money before you commit yourself to use a more advanced enterprise-ready tool.

Support native SQL execution but provide interface for interactive data exploration.

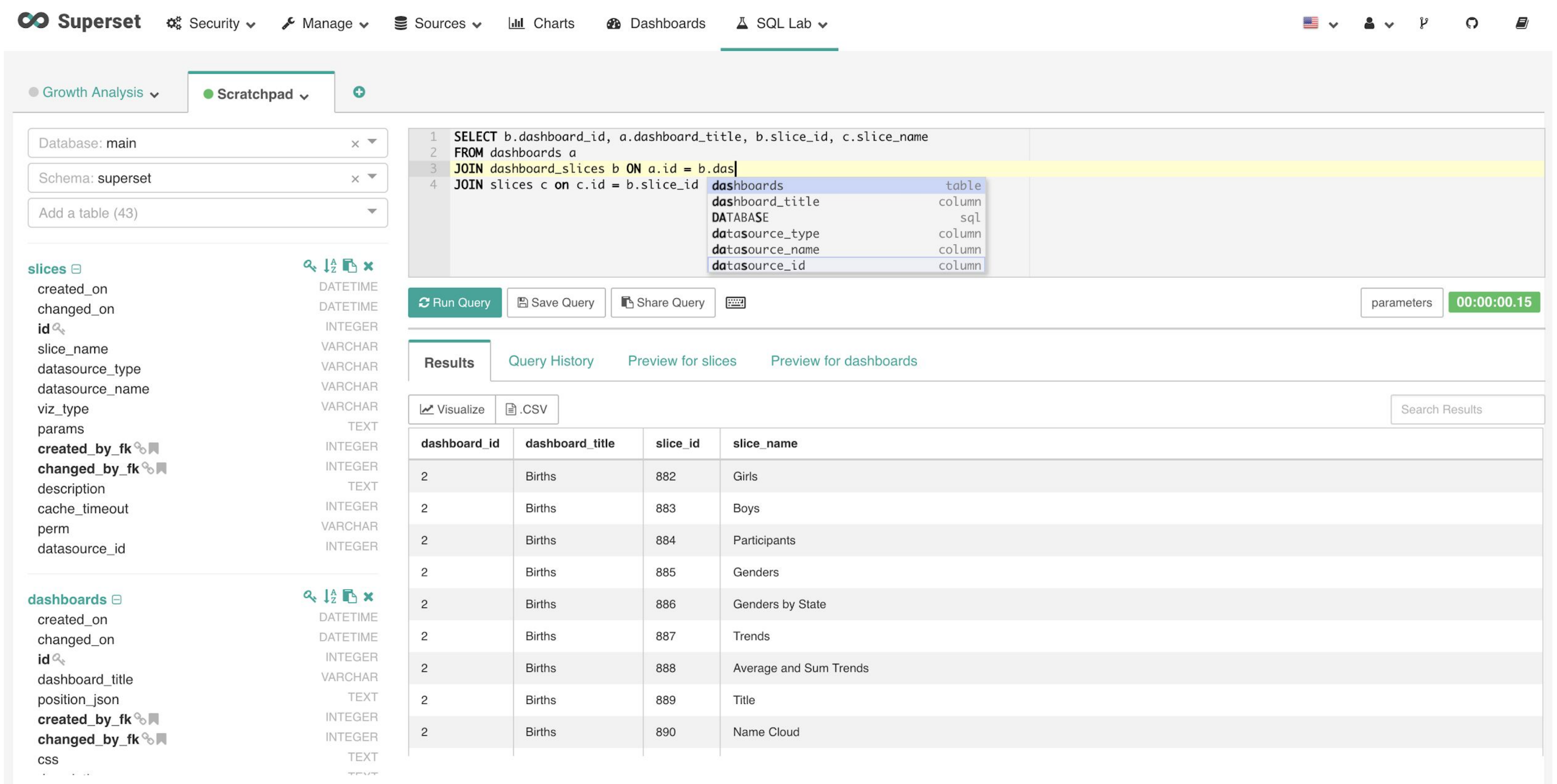
Personal recommendation is to use **Chartio** afterwards.



Most Populated Countries

country_name	sum__SP_POP_TOTL
China	1.36G
India	1.30G
United States	319M
Indonesia	254M
Brazil	206M
Pakistan	185M
Nigeria	177M
Bangladesh	159M
Russian Federation	144M
Japan	127M
Mexico	125M
Philippines	99.1M
Ethiopia	97.0M
Vietnam	90.7M
Egypt, Arab Rep.	89.6M
Germany	80.9M
Iran, Islamic Rep.	78.1M
Turkey	75.9M
Congo, Dem. Rep.	74.9M
Thailand	67.7M
France	66.2M
United Kingdom	64.5M
Italy	61.3M
South Africa	54.0M
Myanmar	53.4M
Tanzania	51.8M
Korea, Rep.	50.4M
Colombia	47.8M
Spain	46.4M
Ukraine	45.4M
Kenya	44.9M
Argentina	43.0M
Sudan	39.4M
Algeria	38.9M
Poland	38.0M
Uganda	37.8M

* picture from https://superset.incubator.apache.org/



```
SELECT b.dashboard_id, a.dashboard_title, b.slice_id, c.slice_name
FROM dashboards a
JOIN dashboard_slices b ON a.id = b.das
JOIN slices c on c.id = b.slice_id
```

```
dashboards              table
dashboard_title        column
DATABASE                  sql
datasource_type        column
datasource_name        column
datasource_id          column
```

slices

created_on	DATETIME
changed_on	DATETIME
id	INTEGER
slice_name	VARCHAR
datasource_type	VARCHAR
datasource_name	VARCHAR
viz_type	VARCHAR
params	TEXT
created_by_fk	INTEGER
changed_by_fk	INTEGER
description	TEXT
cache_timeout	INTEGER
perm	VARCHAR
datasource_id	INTEGER

dashboards

created_on	DATETIME
changed_on	DATETIME
id	INTEGER
dashboard_title	VARCHAR
position_json	TEXT
created_by_fk	INTEGER
changed_by_fk	INTEGER
css	TEXT

dashboard_id	dashboard_title	slice_id	slice_name
2	Births	882	Girls
2	Births	883	Boys
2	Births	884	Participants
2	Births	885	Genders
2	Births	886	Genders by State
2	Births	887	Trends
2	Births	888	Average and Sum Trends
2	Births	889	Title
2	Births	890	Name Cloud

* picture from https://superset.incubator.apache.org/

Collect	Amazon Kinesis Data Firehose
Data storage	Amazon S3
Data flow / ETL	Apache Airflow
Database / Query engine	Amazon EMR - Presto (Amazon Athena for large jobs)
Visualisation tool	Apache Superset or anything else

This is just **one** basic system of the many.

Things to avoid

Don’t

- Use a marketing tool as your main event tracker - sorry but no Adobe Analytics, or Google Analytics.
- Build a data warehouse in a standard relational database. It was not meant for that, and will crash under the data volume.
- Store all raw events within a table without partitions.
- Measure all the events of the universe - because “we (might) need it”.
- Do all your data transformations within a visualisation tool or in Excel.
- Over-engineer & build your own ETL - use something that is done.

Recap

Thank you!
bence@subninja.org

1. Putting together a data infrastructure can be straightforward if we are looking for **products that work together**, and follow the guidance on required product abilities.
2. Don't believe in hype or shiny web pages when selecting products. No product can solve your problems instantly, **you must put together the basics** step by step.
3. **Centralisation** is key. Clean and organise your data - do the legwork!
4. Make sure you **own the data**.
5. Invest in training your people to **learn SQL**.