

---

# *Scale and Optimize Data Engineering Pipelines with Software Engineering Best Practices: Modularity and Automated Testing*

Qiang MENG  
Sr. Data Engineer, Levi Strauss & Co.

**DATA+AI**  
SUMMIT EUROPE  
ORGANIZED BY  databricks

# Qiang MENG

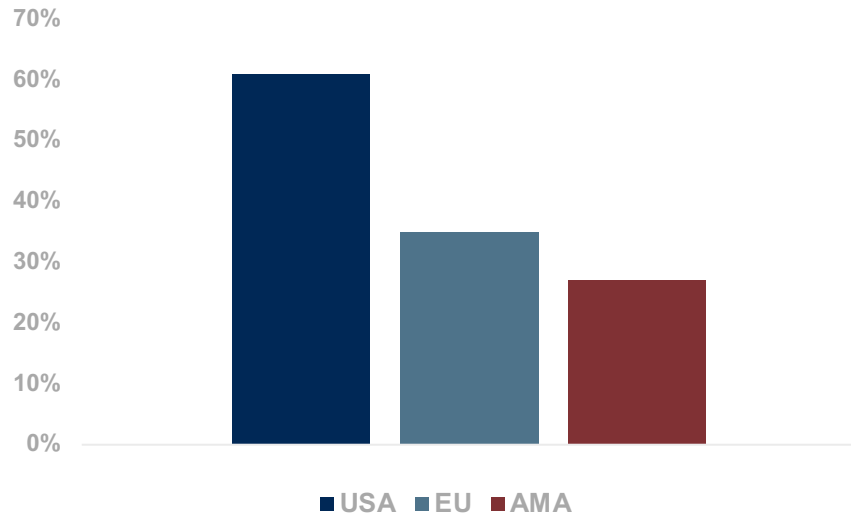
- **Levi Strauss & Co., Belgium**
  - Databricks – AWS – Airflow – Spark - Dataiku
- **H&M Group, Sweden**
  - Databricks – Azure – Airflow – Spark - Hive
- **Softbank Robotics, France**
  - AWS – Luigi – Spark





# Levi's Data, Analytics & AI

Q3 Digital Sales Increase



# Agenda

## I. Hypothesis

- A Data Engineering project in Apache Airflow

## II. Optimization

- Software Engineering best practices

## III. Auto-testing

- Unit test, Functional test, End to End test, and Smoky test

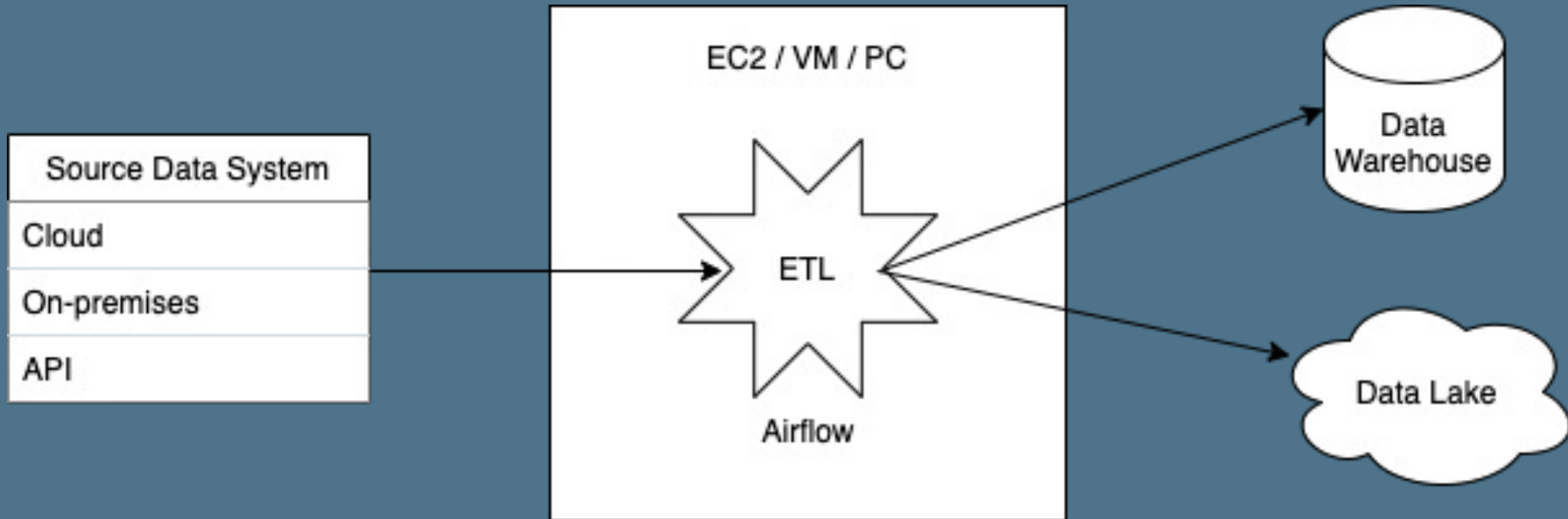


# I. Hypothesis



# Hypothesis

Data Engineering – Cloud - Airflow - TB-level - Batch





# Hypothesis

## Dev Environments

- Local – Dev – Preprod – Prod

## Sample Data

- ETL in local / Dev

## Code Versioning – GIT

- feature - dev - release - hotfix – master

## Conventions

- DB, Schema, Table, Column, Pipeline

## Container-orchestration

- Docker, K8S, ...

## Spark clusters

- Databricks, EMR, ...

## Data Versioning

- Delta Lake, ...

## CICD



# Airflow

- **airflow**
  - dags
  - operators
  - ...
- **utils**
- **modules**
- **docs**
- **tests**
- **others**

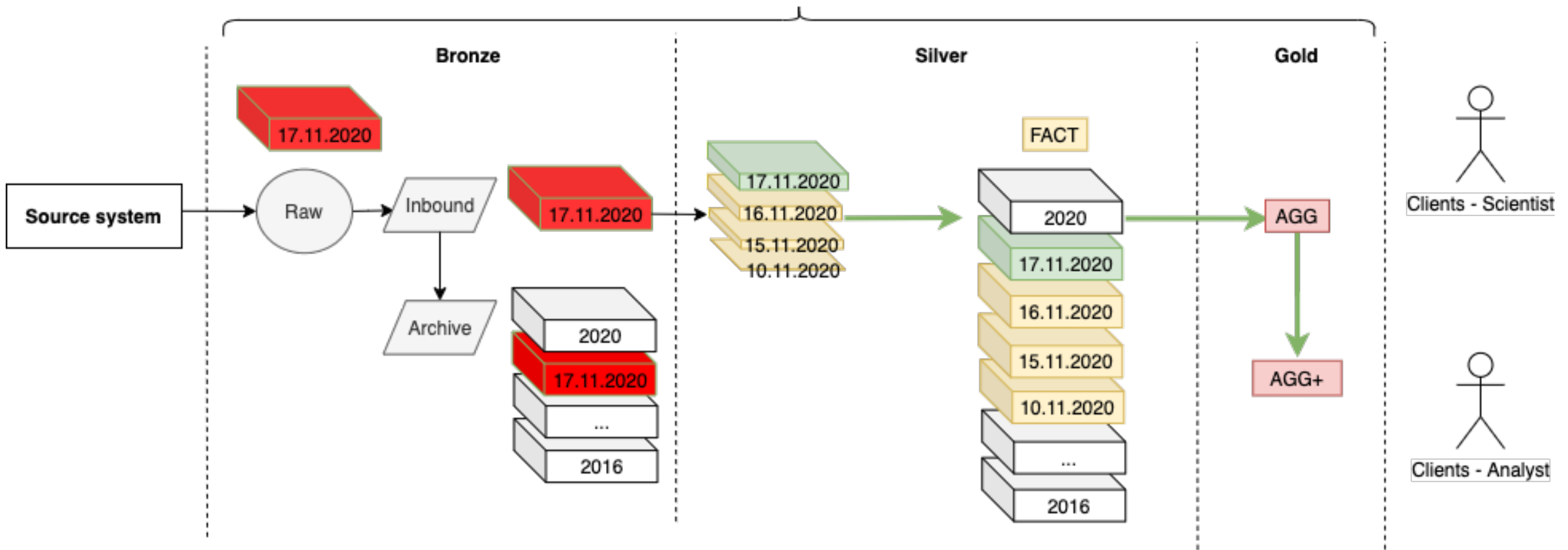


- **airflow**
  - dags
  - perators
  - ...
- **others**

A woman with purple hair, wearing a denim jacket, is smiling and using a handheld payment device at a retail counter. A cashier is holding a smartphone displaying a product page. The background shows a clothing store with racks of jeans and other items.

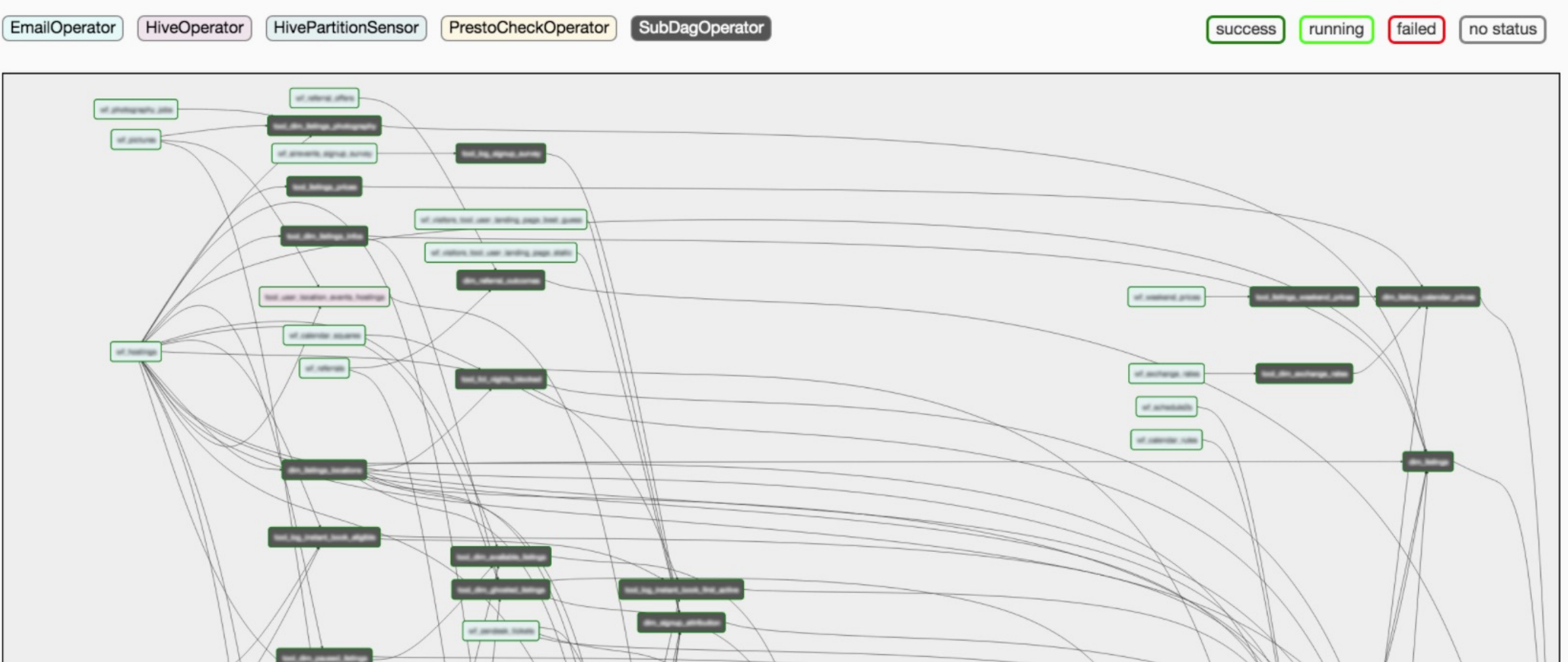
## II. Optimization

# Data Lake



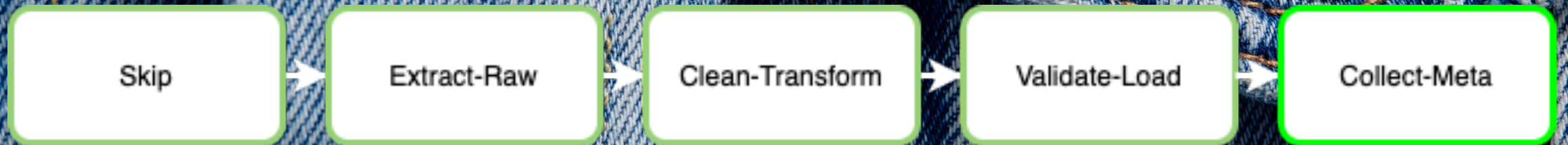
# A Big Data ETL

# Optimization - Airflow DAG



# Optimization - Airflow DAG

A simple template of tasks – Data Feeds Based



# Optimization - Airflow Operator

Do the real work

- Default Operators

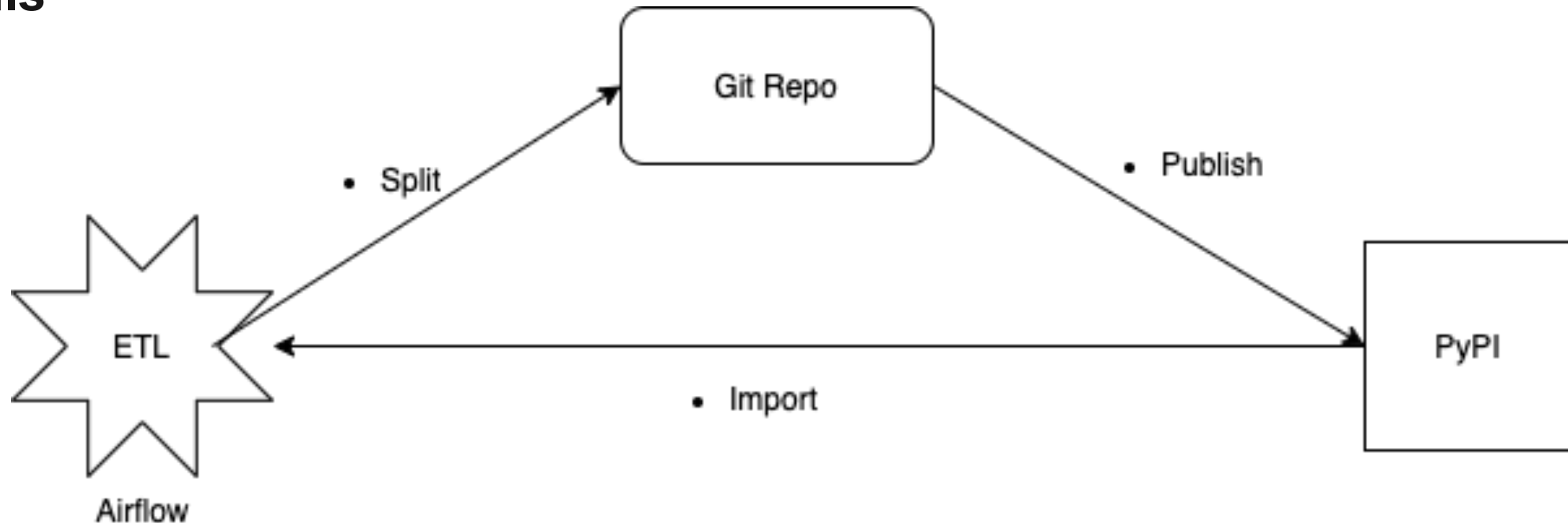
- Keep as they are

- **Customized Operators**

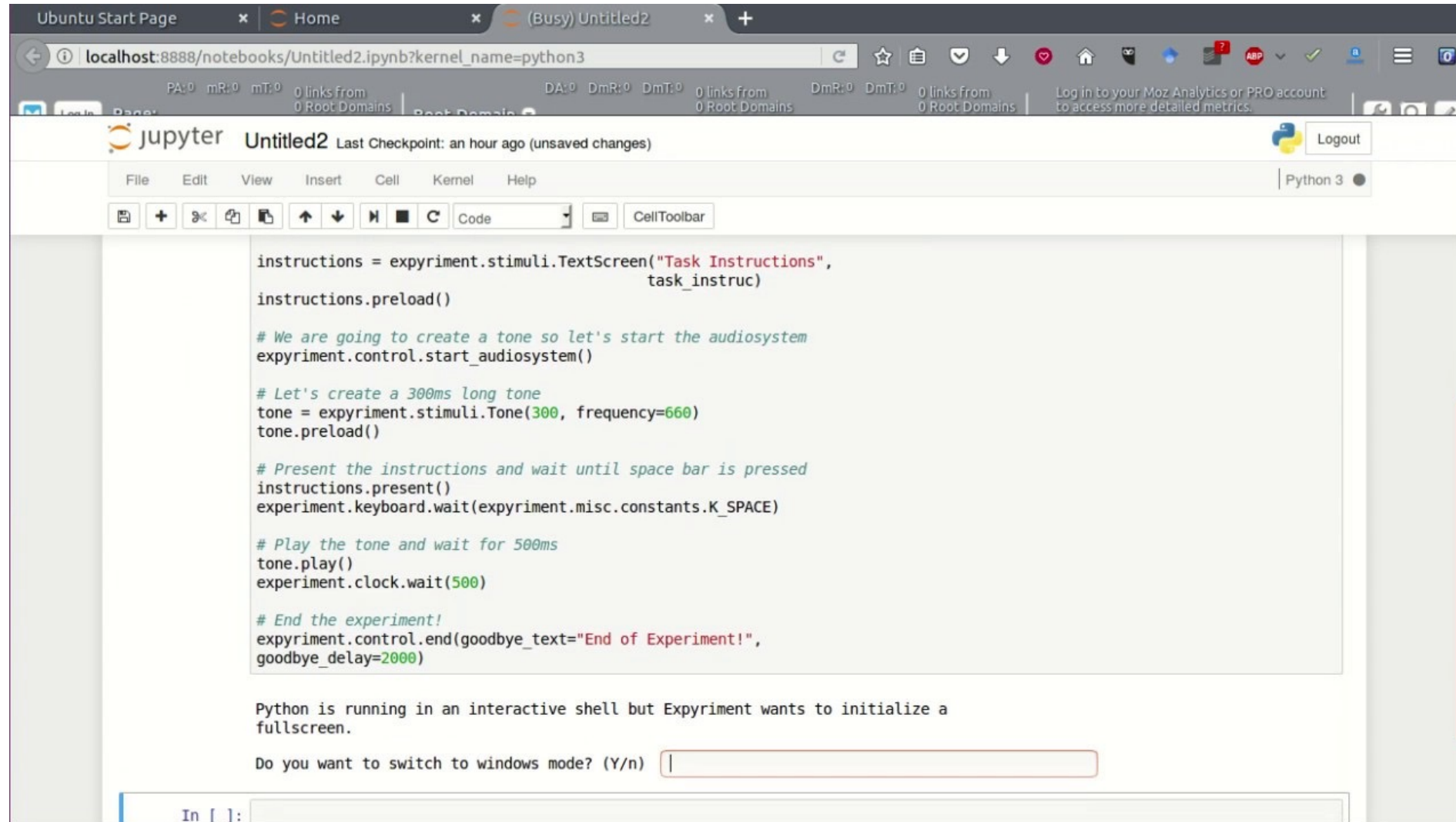
- Hive
- Spark
- Deduplication
- Data Validation
- Data Profiling
- ...

# Optimization - Modules and Utils

- Modules
- Utils



# Optimization - OOP



```
instructions = expyriment.stimuli.TextScreen("Task Instructions",
                                             task_instruc)
instructions.preload()

# We are going to create a tone so let's start the audiosystem
expyriment.control.start_audiosystem()

# Let's create a 300ms long tone
tone = expyriment.stimuli.Tone(300, frequency=660)
tone.preload()

# Present the instructions and wait until space bar is pressed
instructions.present()
expyriment.keyboard.wait(expyriment.misc.constants.K_SPACE)

# Play the tone and wait for 500ms
tone.play()
expyriment.clock.wait(500)

# End the experiment!
expyriment.control.end(goodbye_text="End of Experiment!",
                       goodbye_delay=2000)

Python is running in an interactive shell but Expyriment wants to initialize a
fullscreen.

Do you want to switch to windows mode? (Y/n) |
```



```
class Vehicle:
    vtype = ''

    def start(self, x):
        self.vtype = x
        print('The', self.vtype, 'started')

    @staticmethod
    def cleanVehicle():
        print('Cleaning the vehicle')
```

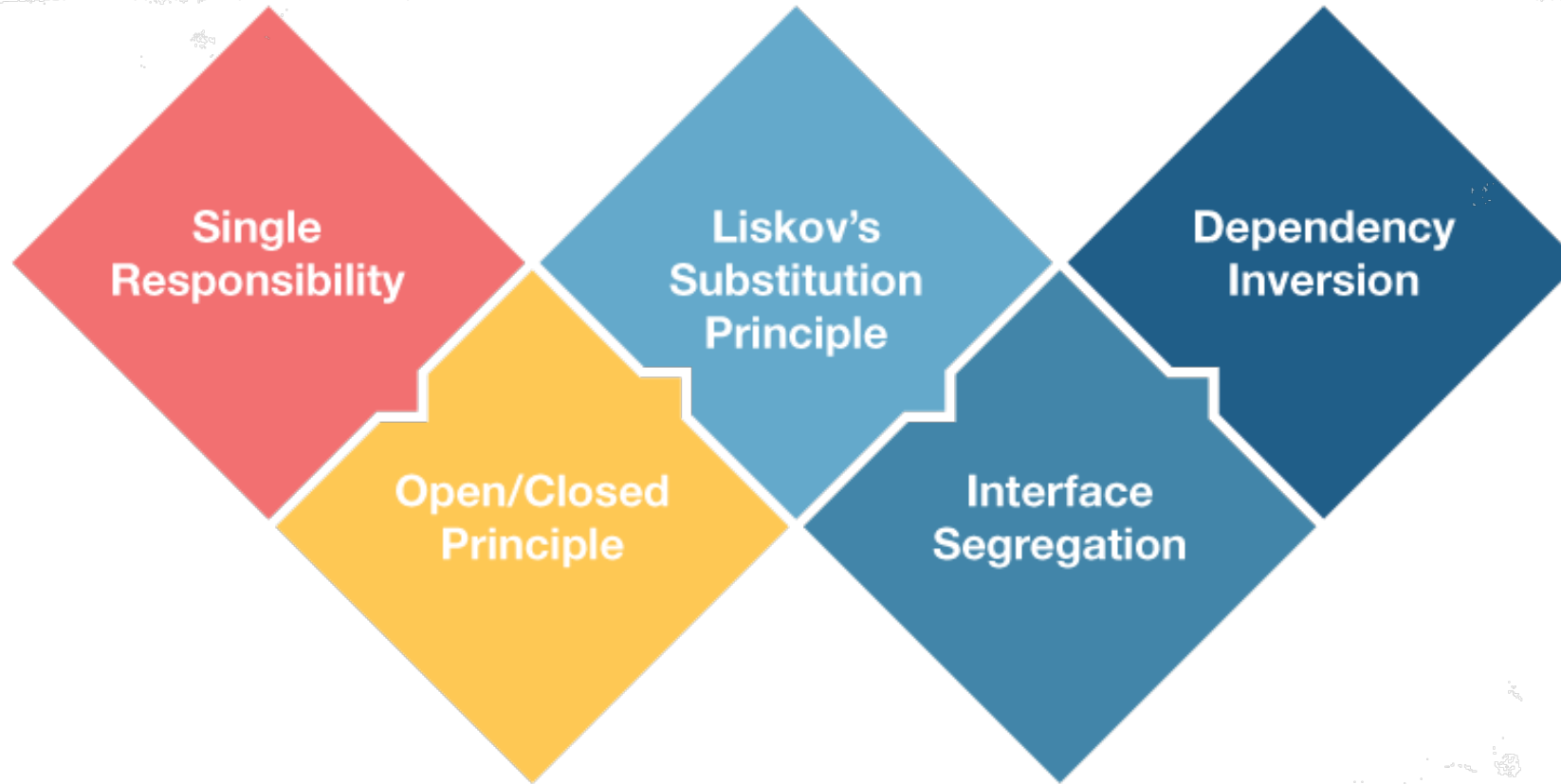
#Auto class extends Vehicle class

```
class Auto(Vehicle):
    num = 2

    def open(self, x):
        self.num = 4
        print(self.num, 'doors were opened')
```

```
auto1 = Auto()
auto1.start("car")
auto1.open(4)
```

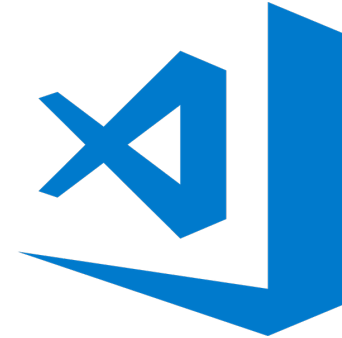
# Optimization - OOP



# Optimization - Design Patten

**sonarlint**

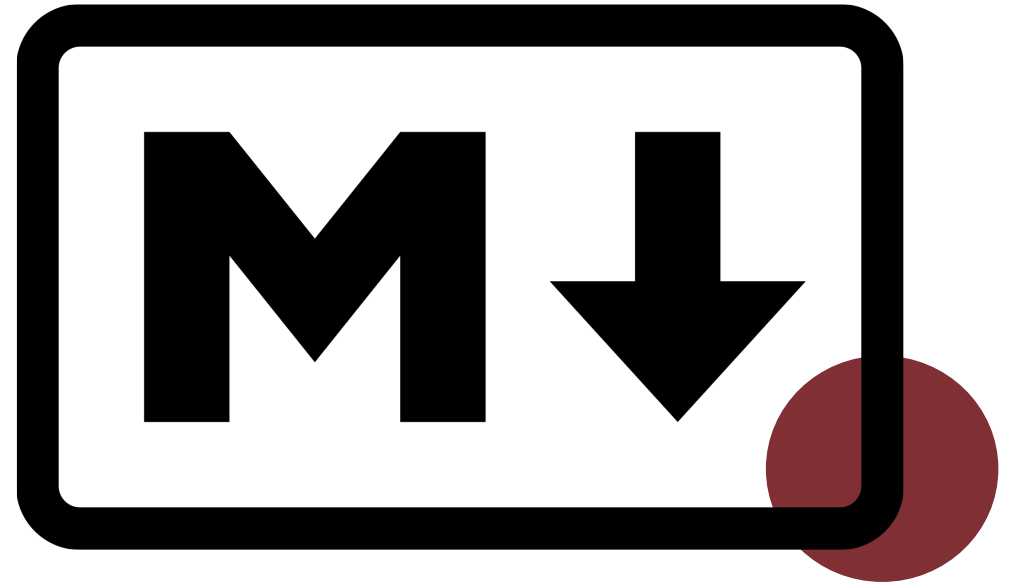
 **Pylint**  
Star your Python code!



# Optimization - Formatting

# Optimization - Docs

- Docstring - Sphinx
- Data Catalog – Markdown



# Docstring

```
# -*- coding: utf-8 -*-
"""Example Google style docstrings.

This module demonstrates documentation as specified by the `Google Python
Style Guide`. Docstrings may extend over multiple lines. Sections are created
with a section header and a colon followed by a block of indented text.

Example:
    Examples can be given using either the ``Example`` or ``Examples``
    sections. Sections support any reStructuredText formatting, including
    literal blocks::

        $ python example_google.py

Section breaks are created by resuming unindented text. Section breaks
are also implicitly created anytime a new section starts.

Attributes:
    module_level_variable1 (int): Module level variables may be documented in
    either the ``Attributes`` section of the module docstring, or in an
    inline docstring immediately following the variable.

    Either form is acceptable, but the two should not be mixed. Choose
    one convention to document module level variables and be consistent
    with it.

Todo:
    * For module TODOs
    * You have to also use ``sphinx.ext.todo`` extension

.. _Google Python Style Guide:
   http://google.github.io/styleguide/pyguide.html

"""
```

# Auto Data Catalog with Markdown

The screenshot shows a web browser window with the title "Welcome to My Project's documentation! — My Project v1 documentation". The address bar shows the file path: "file:///Users/alfredo/tmp/my\_project/\_build/html/index.html". The page content is as follows:

My Project v1 documentation » [next](#) | [index](#)

## Table Of Contents

Welcome to My Project's documentation!  
Indices and tables

### Next topic

[This is a Title](#)

### This Page

[Show Source](#)

### Quick search

Enter search terms or a module, class or function name.

## Welcome to My Project's documentation!

Contents:

- [This is a Title](#)
  - [Subject Subtitle](#)
  - [Inline Markup](#)

## Indices and tables

- [Index](#)
- [Module Index](#)
- [Search Page](#)

My Project v1 documentation » [next](#) | [index](#)

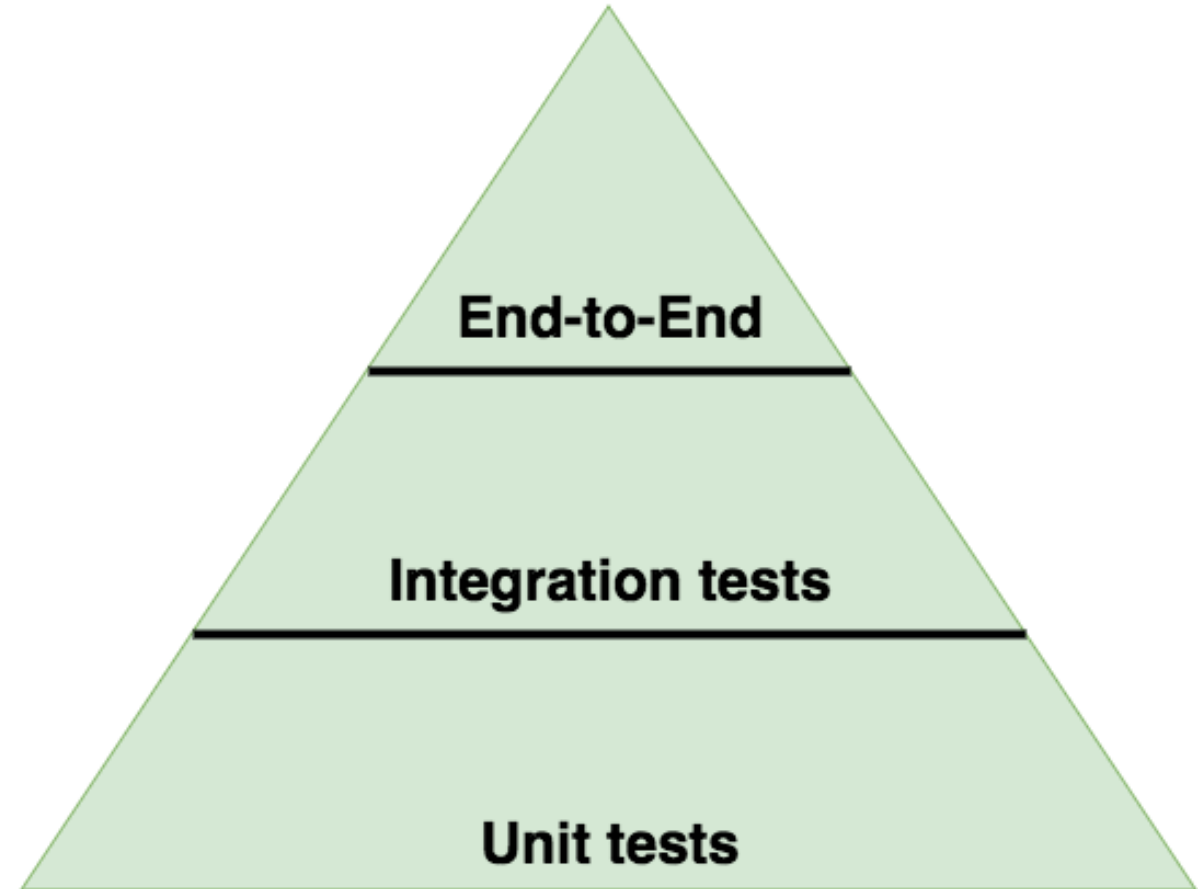
© Copyright 2010, Alfredo Deza. Created using [Sphinx](#) 1.0.5.

A woman with shoulder-length brown hair, wearing a dark, long-sleeved sweater, is focused on her work at a large white table in a factory or workshop. She is handling a piece of blue denim fabric. The table is cluttered with various items, including a measuring tape, a pair of blue jeans, and other fabric pieces. In the background, there are industrial machines, including a sewing machine and a large, illuminated, stylized letter 'O' made of small lights. The overall atmosphere is professional and industrious.

# III. Auto-testing

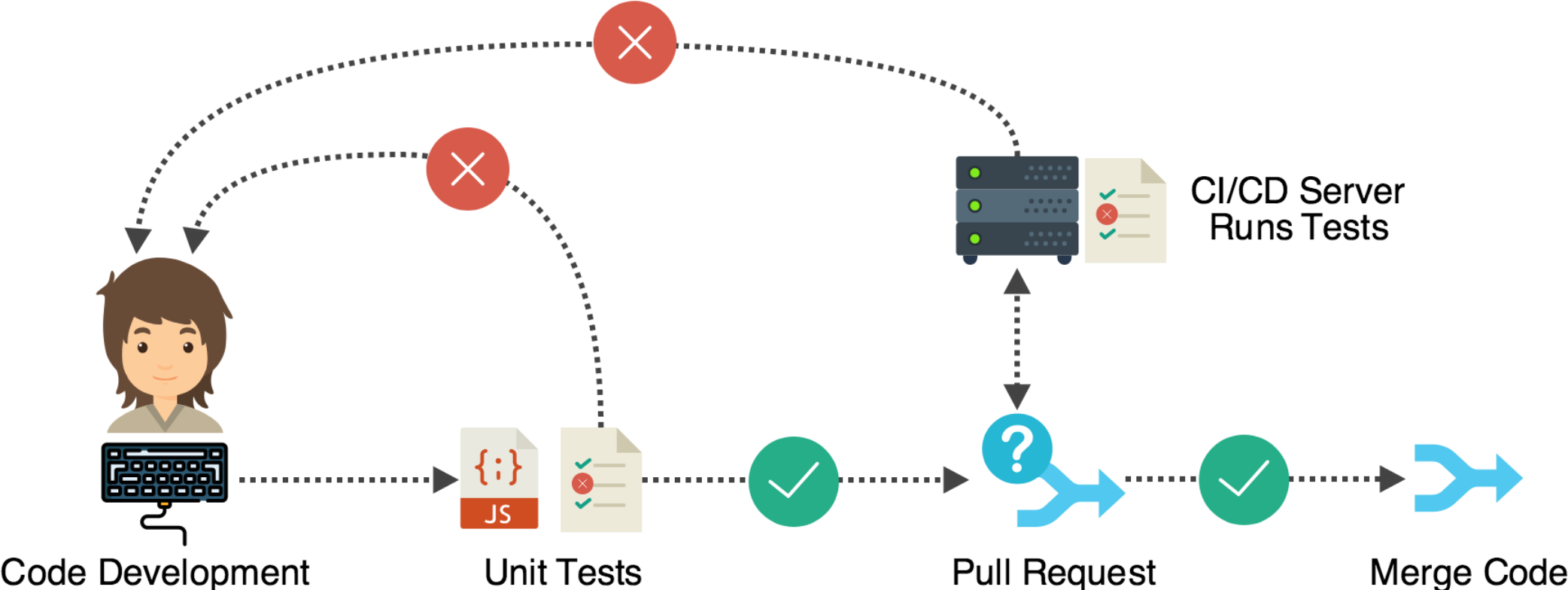
# Auto-Testing

- **Unit tests**
- **Integration tests**
- **End-to-End tests**
- **Smoky tests**





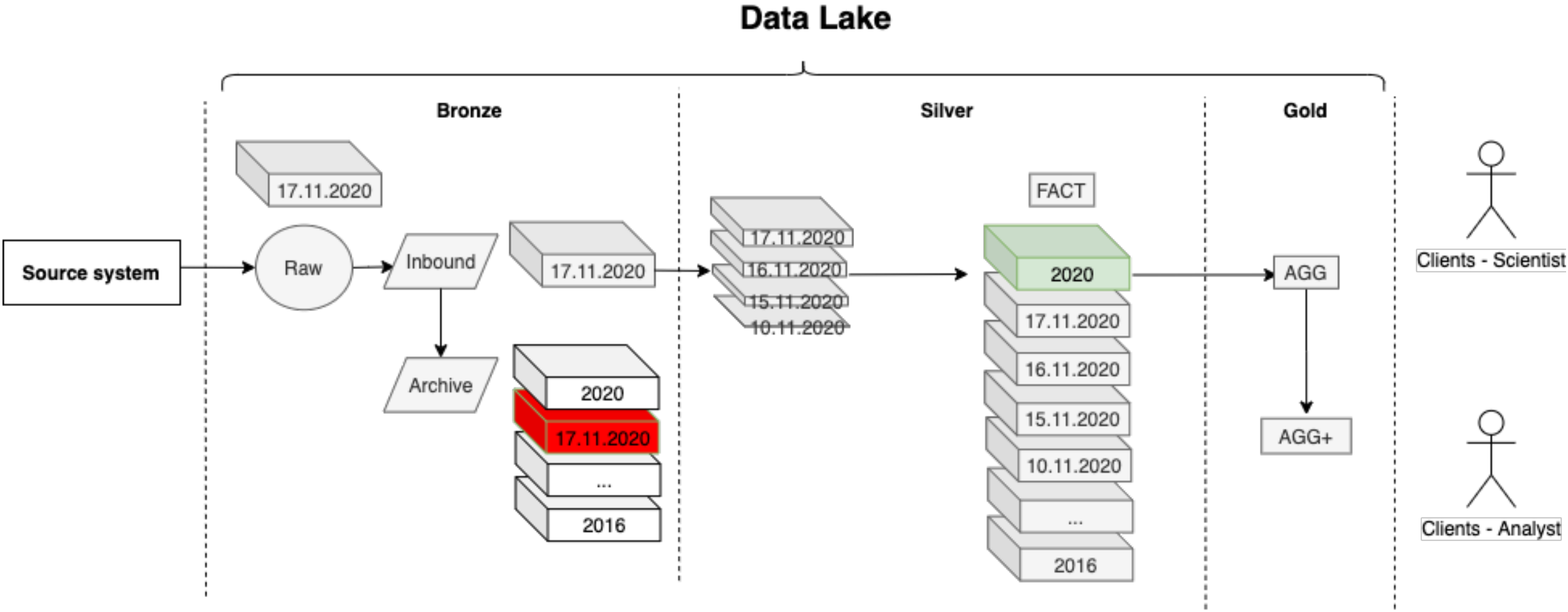
# Unit Tests



# Integration Tests



# End-to-End Tests



# Smoke Tests



**READY FOR  
THE NEXT  
167 YEARS**





THANK YOU

# Feedback

Your feedback is important to us.  
Don't forget to rate  
and review the sessions.

