Product Blog: Tardis

Content:

1. Background and Problem Identification:

The data engineering group at Conde Nast was established with high expectations to improve the efficiency of data science, business intelligence, and global consumer marketing divisions in a successful run of subscription LTV models, various real-time dashboards and live ad-campaigns. Having a team to build reliable, scalable, and repeatable practices into the data platform directly support the ad and subscription revenue management.

Until early 2020, there were no set standards for the data engineering processes and systems. With the organizational growth it was identified that there was no mechanism of intercommunication, a record of Extract, transform, load (ETL) process as well as internal and external process status communication.

Additionally, how does the business get an awareness of the completion of a process, that the data is available for their consumption to be used in their decision-making activities. (It was mainly an expectation that in the morning US business hours most of the datasets will be available in the pipelines, but with no confirmation.)

Summarizing the needs:

- A central system to monitor the health of pipelines, processes and dashboards.
- Ability to log audit trail for our ETL process - a way to capture the important data points in pipelines and processes.
- Resolving dependencies - The sources/pipelines/processes can be dependent on each other and can collectively corresponsd to a variety of projects or business needs.
- Priority levels - Setting up grouped dependencies and defining hard and soft dependencies for downstream to progress.
- Alerts - To hold necessary data to build an alert framework and maintain the definition of alerts

2. Business impact of the problem:

Our ad campaigns, editorial dashboards, and subscription models are running on real-time GA, GAM, Email, and various other consumer data. An un-notified delay in data load, data pipeline failure, or issue with external vendor partner – later troubleshooting the root cause, and manually solving the issues are time-consuming and require additional inventory for making a campaign successful.

Issues with the health of the business-critical data pipelines can cause deterioration of business performance, revenue loss and decline in customer satisfaction, employee productivity, or brand reputation.

There was a need for a mechanism that should trigger the data to downstream applications post data retrieval and provide insights in near real-time to our business users to state the effectiveness of a campaign and dash. This would help in quick decision-making like targeting parameters, and right segment alignment of users. This will be both cost-effective and ensure seamless business performance.

3. **What did it require to solve the problem?**

"Single source of truth for all the problems."

A centralized system that enables business users to understand the health of the processes and pipelines. It assists to build out seamless orchestration among the pipelines and builds a logical grouping of processes as well as pipelines.

We named it **Tardis**.

The major requirements/concepts of Tardis:

- A central process through which every system talks, to elaborate we had to build a pivot point for any communication that goes in-and-out.
- To design a system of this concept - the challenges were:
    - Where does it fit? What would be the endpoints through which different communications can take place?
    - How easy would it be to come up with a system like this?
    - We also had to consider what features to add for other systems to communicate out of their boundaries?

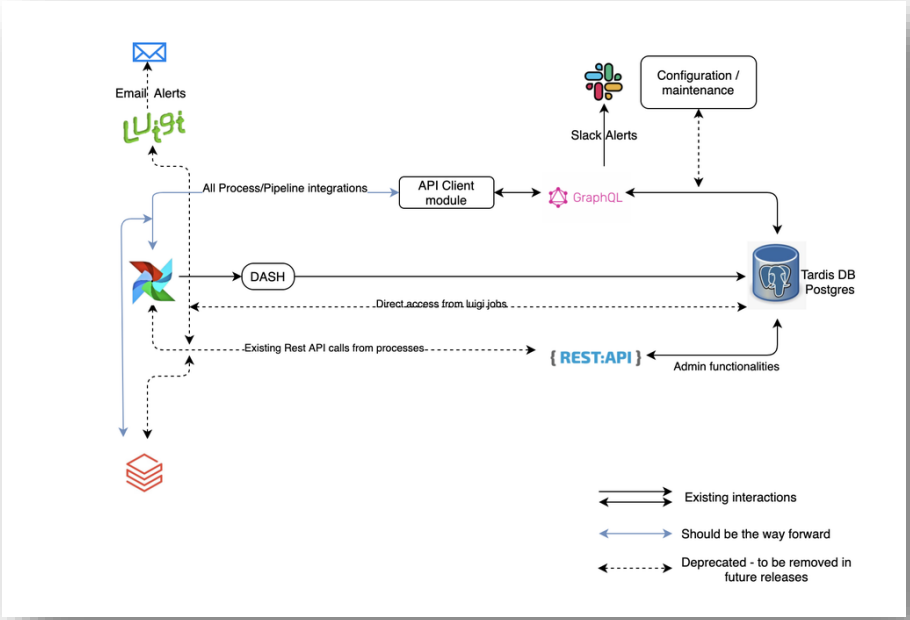
To solve these:

- We built an API layer which was the most common framework available to incorporate the communication between two digital systems. The API endpoints can serve as a gateway for in-and-out of the systems.
- We dealt with a central database – Postgres. This was our system of records of all our events that are various happenings in pipelines and downstream processes.
- On the top of the central database, an API engine was implemented which exposes communication from-and-to Tardis.
- A User interface that presents out various forms of data reliability matrix such as the status of pipeline run, the success rate of ETL processes, failures rate of ETL processes, and types of failures.
- A global support team to fix the pipeline issues, managing the pipeline manual run updates.

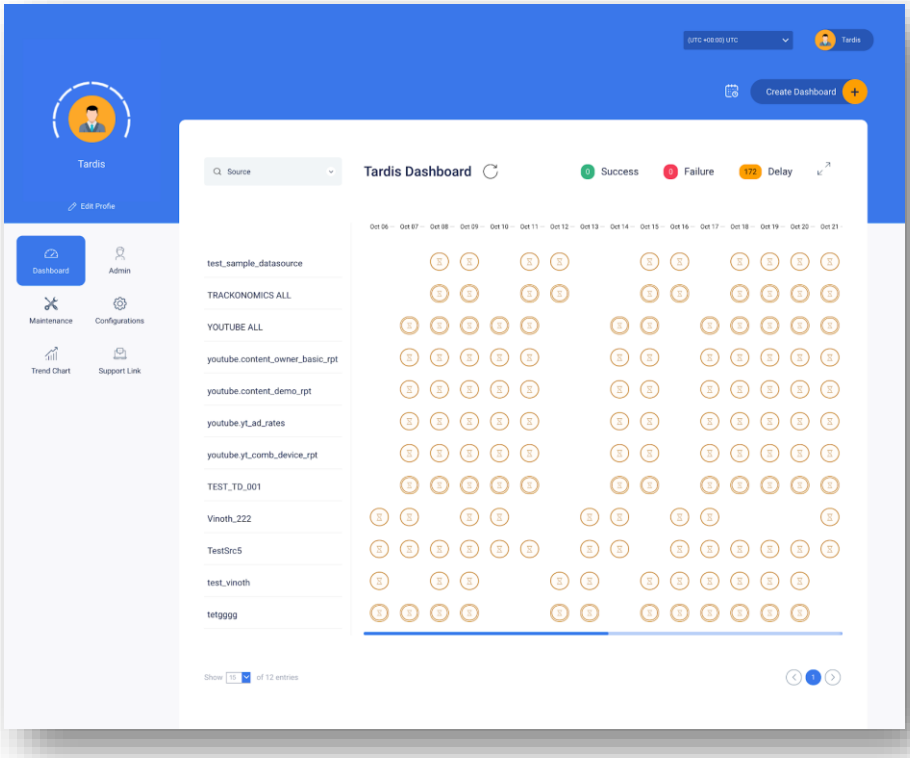
Summarized Product Goals:

- Build an application that enables users to understand the health of the processes and data pipelines of Conde Nast.
- It should build out seamless orchestration among the pipelines and build a logical grouping of processes as well as pipelines.
- The different statuses of the sources can be sent in the form of alerts via slack and emails.

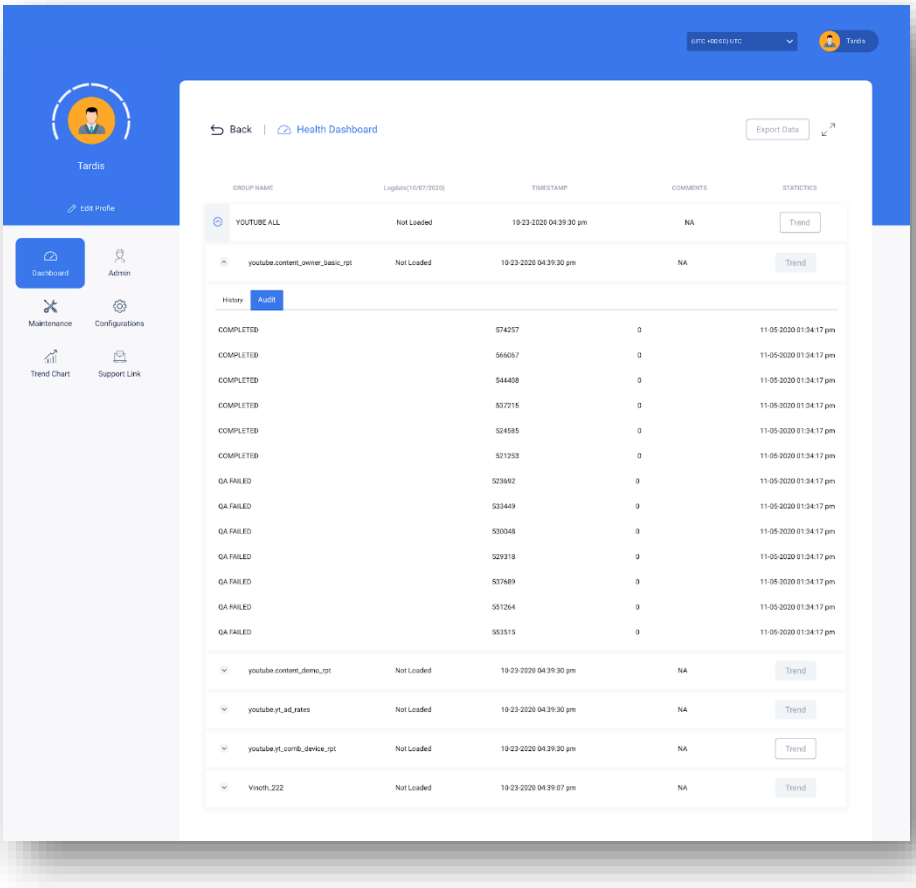
Architecture:



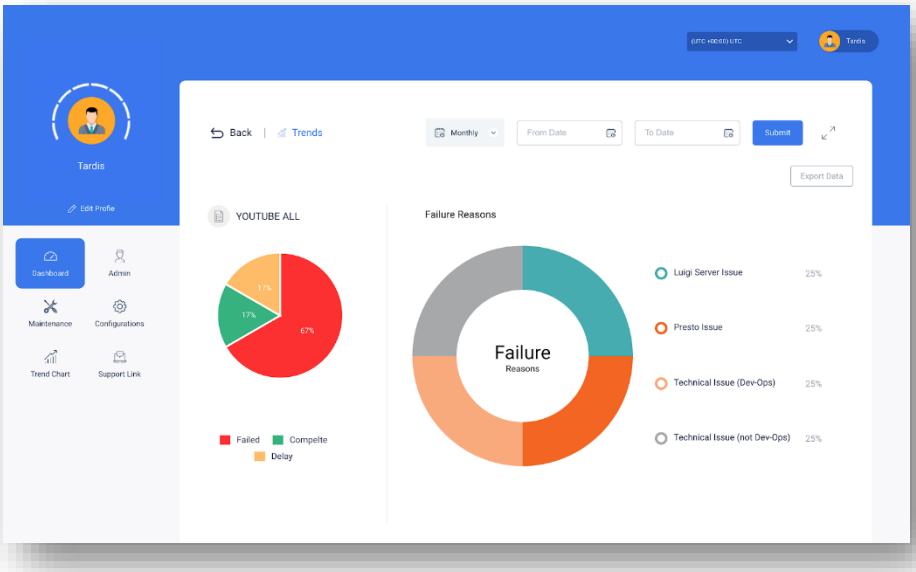
Snapshot of the Application:



Data Pipeline Sources – Detailed View:



Performace Trend Analytics:



4. Knowing the business impact of the solution?

We used three parameters to understand the impact:

- Business user and stakeholder feedback
- Understanding the Outcomes
- Measuring the usages

To summarize the overall impact:

- In a quarter after the release our global engineering, campaign, consumer marketing, business intelligence, data science and AI infrastructure teams started using Tardis daily and adopted as their prime platform to check details of data.
- Establishment of a 24x7 global support team to maintain the data pipelines.
- 30% improvement in ad-revenue through B2C campaigns.
- We identified the non-performing vendor side API by observing the data trend and created another route, that has helped businesses to get more real-time commerce data.
- **Critical feedbacks:**
    - ***From leadership:*** A need to ensure data reliability and transparency to enable our teams to scale and meet the demand of use cases across the enterprise. - A new product is in ideation, which is a monitoring and traceability solution that connects our internal & external systems to enable tracking of data downtime that occurs in downstream dependencies.
    - ***From Business User:*** An update on digital ad traffic availability required – We implemented AdOps process on top of Tardis, that makes users easier to know when digital ad traffic is available for their consumption, they get notified via slack and email, and check the brand trends in the dash. This supports their business decisions.

5. The team

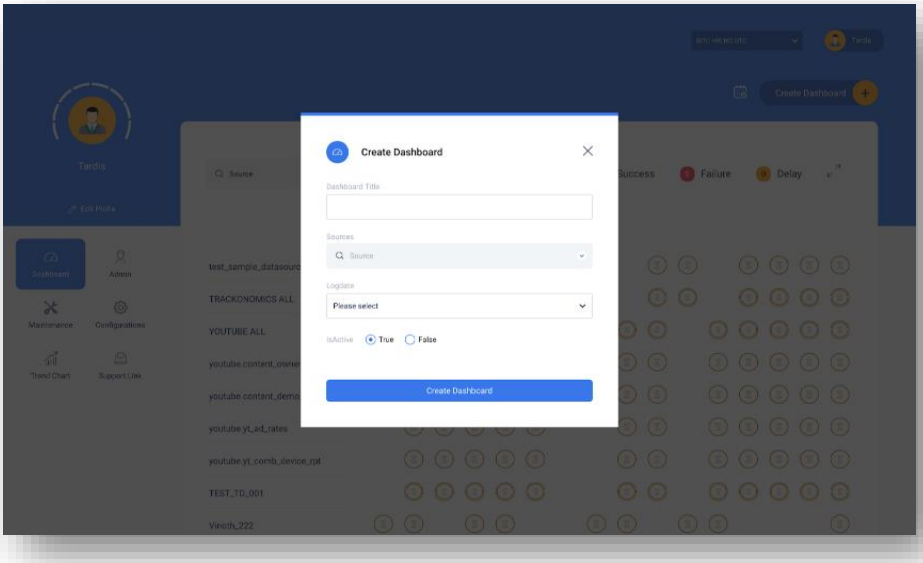
Team Member	**Responsibility and contribution**
Data Engineering Lead	• Developed, constructed, and maintained the pipeline architectures • Aligned architecture with business requirements • Review and manage works of Data Engineers
Data Engineers (2)	• Perform the ETL process and share update on each steps in the process • Existing functionality enhancements: It includes the movement of Tardis related jobs from Luigi to Airflow, higher configuration oriented data model, and optimizing the DB functions • Work to improve data reliability, efficiency and quality
Backend Engineer (2)	• Understand the end-user requirements and application architecture and create APIs. • Conceptualize and implement solutions for data storage. • Configure the backend usability of all front-end side application.
Data Solution Architect	• Complied with all technical and business requirements, created an application framework • Guided the tech teams on use of technologies.
Designer	• Understood the requirement, and created a visualization of the user interface.

	• Worked closely with product team and improved design based on the requirement and feedback.
Frontend Engineer (2)	• Developed the application on react framework based on the design finalized. • Contributed in design feedback. • Integrated the APIs with developed designed and developed a working application
QA	• Validation of data, design and functionality in both staging and production
Product Managers (2)	• Managed the entire product lifecycle and product roadmap. • Worked on defining the product vision and goals. • Defined the releases, evaluated ideas, prioritized features and performed UAT. • Worked with Tech, architecture and design team in finalizing the architecture and design. • Communicated with stakeholders and leadership on progress, feedback and launch

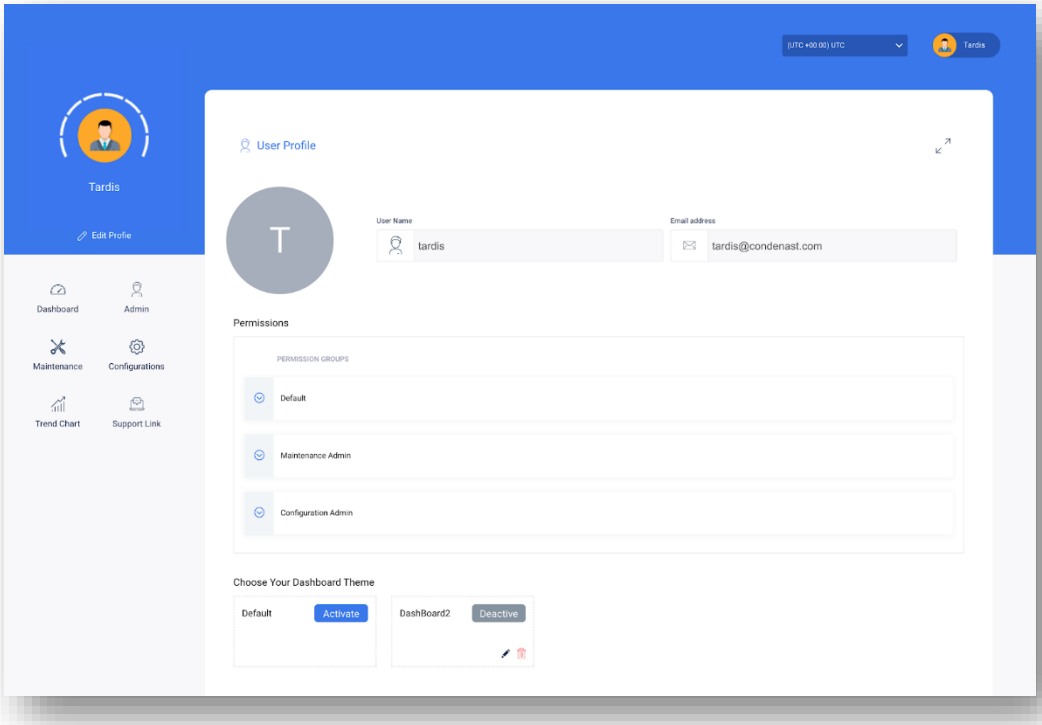
Additional Information:

- Customized Dashboard Feature:
  This feature allows to create customized dashboards and save it. These dashboards have the following features:
    - Options to choose the sources, the window of log dates, the statuses and group by which of these categories.
    - The user can choose from the saved dashboards. If not, the default dashboard will be shown which shows statuses for sources for the current date.
    - Source Name should be a drop down where multiple sources can be chosen.
    - Log Dates can be the following,
        - Daily
        - Daily + previous day (day + day-1)
        - Weekly (day + day-6)
        - Monthly (day + day-30)
    - This dashboard offers an additional reporting feature which is quite useful in sending specific reports to vendors, performance reports and leadership meetings.

  Visualization:
- Creating a custom dashboard:



- Edit Profile: (Activate, Deactivate Modified and Delete Customized Dashboards):



Thank you!