



CAREER PROFILE

I'm a **Data Architect** in the Data Architecture team at lastminute.com Group. I design and own the **Google Cloud** ecosystem that hosts our heterogeneous data: streaming events from Pub/Sub, feeds from third-party partners, and replicas of relational databases, landing on Cloud Storage and **BigQuery**. Analysts and data engineers build on it every day.

My work sits where architecture meets platform engineering: deciding how data lands, how it is stored, and how it is served. That means picking the right serialization format for each stage (**AVRO**, **Parquet**, **ORC**), designing the IAM model that keeps consumers on curated datasets instead of raw ones, and describing the whole estate as code in **Terraform** so it stays reproducible rather than hand-made. Alongside BigQuery the platform still runs on **Vertica**, so most designs have to hold in both.

AI is part of my daily toolchain, not a side experiment. I work with **Claude** and **Gemini** every day, and I bring Claude into how the platform is run: agentic tooling that reviews pipelines against our internal standards, queries the warehouse, and turns platform knowledge that used to live in people's heads into something applied consistently. I care just as much about the engineering discipline around it: context design, guardrails, and a human accountable for whatever ships.

I started out in 2015 in Rome, building on-premise Hadoop and Spark platforms, then moved to Switzerland to work on cloud data infrastructure at scale. That background still shapes how I reason about distributed systems.

120k+

events per second
sustained by the
streaming ingestion I built

60B+

sensitive records
pseudonymised for GDPR
compliance

529

tables replicated from 58
sources into BigQuery



EXPERIENCES

Data Architect

lastminute.com Group · Data Architecture

Jan. 2026 - Present

I own the architecture of the **Google Cloud** ecosystem that hosts the group's heterogeneous data and serves the analysts and data engineers who build on it: Pub/Sub event streams, third-party partner feeds and relational database replicas, landing on Cloud Storage and **BigQuery** as **AVRO**, **Parquet** and **ORC**.

The event platform is contract-first. An Avro schema is validated for backward compatibility in CI, and the pipeline then provisions everything the event needs on its own: the Pub/Sub topic and its streaming subscription, the BigQuery table and views, and the **Vertica** counterparts. Product teams onboard a new event with a merge request instead of a ticket to my team.

The estate is described in **Terraform** and applied only through pipelines, never from a laptop: projects, IAM, BigQuery datasets, GKE, Cloud SQL and buckets, with automated security scanning as a gate before apply. Access governance is part of that model, keeping analysts on curated datasets and out of the raw and staging layers.

AI is part of how the platform is run, not a side experiment. I contribute to building the **Claude** skills and context the team works with, and use **Gemini** alongside them: tooling that reviews Airflow DAGs and Terraform against our internal standards, queries the warehouse, and turns platform knowledge that used to live in people's heads into something anyone can apply consistently.

Senior Data Engineer

lastminute.com Group · Data Platform

Nov. 2017 - Dec. 2025

I built the **Scala** tool that turns Avro event schemas into provisioned infrastructure across **Vertica** and **Google Cloud**. It carries an **ANTLR4** grammar that parses existing Vertica DDL into an AST, so views could be evolved safely as schemas changed instead of being rewritten by hand.

I contributed to the Airflow estate of roughly **190 DAGs** orchestrating ingestion into **BigQuery** and Vertica, from relational replicas to partner feeds. I deployed Airbyte on GKE Autopilot with a Cloud SQL backend and tuned it to replicate **529 tables from 58 relational sources**, mostly MySQL, into BigQuery.

Earlier on, when I joined as a **Data Engineer**: a realtime ingestion from Pub/Sub to Cloud Storage sustaining **120k+ events per second**, a Spark job that pseudonymised **60+ billion** sensitive records for GDPR, and Scala services pulling data from REST APIs, SFTP, S3 and Google buckets.



Dario Santilli

Data Architect

CONTACTS

✉ santilli.dar@gmail.com

📍 Province of Como, Italy

🌐 dariosantilli.pages.dev

🌐 [dariosantilli](https://dariosantilli.com)

📄 [resume](#)

📄 [printable resume](#)

EDUCATION

Bachelor's degree in Computer Engineering

University of Roma Tre
2010 - 2013

Master's degree in Computer Engineering

University of Roma Tre
2013 - 2015

LANGUAGES

Italian

English

For a telco customer I was responsible for the design, installation and maintenance of Hortonworks Data Platform on Dell EMC Isilon.

I built a near real-time ingestion storing **25M+ rows per minute** on Apache HBase, covering mobile-user signals joined with ISTAT open map data, and a multi-tenant platform where clients filtered it into sliding-window heatmaps of population movement across Italy.

CTO & Co-founder
idSoul

2014 - 2015

Co-founded a **toys-to-life** startup: an app that turned a child's own toy into the main character of a game, recognised through the tablet camera. Competing with Disney Infinity and Lego Dimensions, our angle was full customisation.

SKILLS

GOOGLE CLOUD

BigQuery

Pub/Sub

Cloud Storage

IAM

Composer

GKE

Cloud Run

Cloud SQL

Compute Engine

Monitoring

DATA PLATFORM

Vertica

Airflow

Airbyte

dbt

Spark

AI

Claude

Gemini

Agentic workflows

Model Context Protocol

INFRASTRUCTURE AS CODE

Terraform

GitLab CI

Helm

Docker

FORMATS & MODELLING

AVRO

Parquet

ORC

Schema evolution

Data governance

LANGUAGES

Scala

Shell

SQL

Python

I authorise the processing of the personal data contained in this CV for recruitment and selection purposes, under EU Regulation 2016/679 (GDPR) and the Swiss Federal Act on Data Protection (FADP).

STUDYING

- Google Cloud Platform
- Terraform
- Gemini
- Claude
- Model Context Protocol

INTERESTS

- Volleyball & Beach Volley
- Playing guitar
- mtg