Jayesh Sidhwani

CONTACT

Mumbai, India
jayesh.sidhwani@gmail.com
+91 9970112010

I started my career by co-founding InteractEasy - a SaaS based internet startup. Post that I've had experiences working with early-stage and mature startups for around 4 years.
Over the course of my career I've had opportunities to architect a monolithic code into microservices, build scalable data pipelines, build scalable & reactive data-science pipelines and build an infrastructure that enabled inter-service communication that ***scaled up to 200K QPS***

EXPERIENCE

Grofers
February 2016 — Present

Lead Engineer

Grofers is a hyper-local e-commerce startup.

I started the data infrastructure team at Grofers. Over the last few months I have worked on the following projects:

- Date Warehouse
  When I joined Grofers, the team had multiple micro-services and each service maintained their own database. An analyst used to query multiple databases to build reports, introspect anomalies etc. This did not scale well. I built scalable data pipelines that consolidated data from multiple sources into a common data warehouse - Redshift.
  The data pipelines only propagated the changed-data each time it ran. This enabled the warehouse data to be refreshed near real-time. Today, the data in the warehouse refreshes every 30 mins. This enables the business analysts to react on the data quickly
  To build the data-pipelines, I have worked with technologies such as: GoLang, Spark and Redshift
- Nyala
  Nyala is an inter-service asynchronous event-driven messaging system that is used for inter-service communication. It provides high-level abstractions to reliably deliver messages between micro-services.
  Without Nyala, micro-services would do resource-blocking asynchronous communication to share data. Such a communication pattern caused many catastrophic events; sometimes causing a domino effect to bring all the services down because of a fault in a single service.
  Nyala simplifies the communication pattern. When a service drops a message, any service can consume it whenever it wants.
  Nyala is architected to be a highly-available, durable, high-throughput & high-latency message queue.
  Nyala gives a throughput of **200K QPS at peak.**
  A scalable system like this needs a comprehensive monitoring and alerting framework. I have used StatsD and Grafana to achieve that. *To build Nyala, I have worked with technologies & frameworks such as: Spark, GoLang, Docker, Kafka, StatsD, Grafana*

TinyOwl
December 2015 — January 2016

Senior Software Architect

TinyOwl is an online food delivery startup

I joined as a software architect to re-architect the existing monolithic application to multiple micro-services. The existing monolith caused plenty of scaling issues. I worked on the following projects:

- Re-architect the existing monolith
  The existing monolith was broken down into multiple micro-services. The entire system became a polyglot where each service was written in a language/framework that best suited its needs.
- Product Intelligence
  At this point, TinyOwl did not have any intelligence in the app. Being in the food industry, it gave TinyOwl immense opportunities in understanding the taste patterns of the users and making the entire application highly context driven.
  With this objective, we started the Product Intelligence team at TinyOwl. I lead a team of 10 members.
  - Data Warehouse
    We built data pipelines that consolidated data from multiple data sources into a common data-warehouse, Redshift. The data-pipelines were batch jobs that refreshed the data once a day.
    These pipelines provided data to the data-science team.
  - Mood Graph
    We modelled user's taste preference in a taste graph. The graph mapped a user to the different types of food, ingredients, specialities etc. The various traversals on this graph helped us identify the mood/preferences of the user and present him with the food choices he liked.
    The data science team used this information to build various models to personalise the user's experience.

Shopsense
June 2013 — December 2015

Lead Engineer

Shopsense is a platform that brings the beauties of the e-commerce world to brick and mortar retail stores. It enhances the user's in-store shopping experience by augmenting their interaction with enhanced search, product discovery, product recommendation etc. via a mobile app or in-store kiosks.

I was the first tech-employee at Shopsense. My initial days were spent in designing the architecture of the application. The backend infrastructure has an hybrid-architecture. This was to ensure that no real-time calls were made to the server to fetch product data. This led to interesting things that made us mimic the store data on the local machine near real-time while saving on the network communications as much as possible

I have worked on the following projects while at Shopsense:

- Lemur
  Lemur is a semantic search and product recommendation engine. Search is written using Solr. It uses NLP to extract the contextual information from the search query before querying Solr. Search also had auto-suggestions based on word-collocations.
  The Recommendations are based on Collaborative Filtering algorithms, Basket Analysis of the transaction data and the taxonomy graphs of the products. The recommendations ensure better discovery of products in store - both breadth-wise and depth-wise. The biggest challenge with Lemur was to avoid Long-Tail recommendations.
- Owl
  Owl is a real-time data ingestion platform. It ingests data coming from different sources (POS, ERP, CRM et. al.) into the database. During the process it cleans the data, massages the data and generates the statistical summary to understand the health of the data. The biggest challenge was to build an infrastructure that handles large number of incoming data. At the current capacity, OWL handles roughly 400 data files per minute - each file being ~100 kb big.
- Beat
  Beat is a real-time data visualisation platform. It represents the large volume of inventory, transaction data in charts. The businesses can do cohort-analysis on the charts by doing custom pivots. Beat is like a tableau. The biggest challenge with Beat has been to make queries run as fast as possible and to hit the database as few times as possible!

Fab Inc. August 2012 — June 2013

Software Engineer

Fab is one of the fastest growing E-Commerce market place for everyday designer products.
I was a part of the Consumer Team that builds the Fab.com web application used by millions of people worldwide. I worked on Rails.
I joined Fab when it was operational only in US and have been a part of the team that scaled it to cater to people in Germany and UK. I have got a working experience in building scalable web applications.

InteractEasy November 2011 — June 2012

Co Founder

InteractEasy is a plugin for websites that enabled enhanced customer support to the customers.
InteractEasy was a customer-support plugin for e-commerce websites. Customers used it on the websites to resolve their pre / post sales queries over VoIP without leaving the website.

InteractEasy was used by 25+ paid businesses across US, India and Middle East.

My role here was to build a Voice Engine that enabled cross browser, cross platform voice conversation.

SKILLS

- Languages
  Scala, GoLang, Python, Ruby
- Frameworks
  Spark, Storm, Pig, Django, Rails
- Database
  Postgres, Redshift MySQL, Redis

EDUCATION

Computer Engineering 2007 — 2011

Vivekananda Education Society's Institue of Technology

Secondary School Certificate 1992 — 2005

Fatima High School

ACHIEVEMENTS

Get Shit Done! Award:

Fab thrives on 2 principles - 'Make Mistakes' and 'Get Shit Done'
I was awarded the 'Get Shit Done' award for gleefully doing what came my way.

Sir Ratan Tata Scholarship:

Received the Scholarship in June 2008 for academic excellence during my engineering studies.

Best Student:

I was awarded the Best Student of the school for the academic year 2004-2005

Microsoft Student Partner:

I was selected by Microsoft to represent the Microsoft in the local Mumbai Student Community.
As a part of the programme, I took workshops on Microsoft technologies, held meet-ups etc.