

[LinkedIn](#) [Github](#) (2018-2021) [Github](#) (2021-till now)

EDUCATION

- Master's in CS at University of Illinois, Urbana-Champaign, GPA - 3.5 2013 - 2015
- BE in Computer Science, BMS College of Engineering, Bangalore, Aggregate: 79%, GPA - 4.0 2005 - 2009
- St. Joseph's College, Bangalore. 12th 98.3% 2003 - 2005
- Cambridge High School, Bangalore 10th 91.7% Till 2003

PROFESSIONAL EXPERIENCE

- * foxpass.com (Headquartered in San Francisco) March 2021 - till now (Designation: Sr Staff Engineer)
 - YC startup providing server and WiFi authentication as a service to SMBs. We were acquired by a unicorn company Splashtop in March 2023. Key member in helping Foxpass reach multiple millions of ARR.
 - I am responsible to ensure our home grown LDAP cluster is highly scalable yet highly cost effective and uses optimum amount of resources. The cluster is home to several customers, serving about 20-25k qps and hosting about 150k persistent TCP connections at any given point of time. LDAP is written using twisted framework.
 - In a very short time, helped reduce the search latency of backend LDAP servers by implementing context aware search indices. Search latency improved by several folds for specific search/read requests. This also brought down the CPU utilization by 30-40%
 - Worked on scalability bottlenecks which prevented us from scaling our LDAP servers. These changes helped to scale our LDAP servers seamlessly. These improvements and search optimizations helped us bag some of our crucial big ticket customers. The changes also helped us scale in a more responsible and cost effective ways.
 - Rewrote the story of LDAP systems. **Built clustered LDAP with partitioning**. Wrote a [blog](#) on the same, published on the company website. This reduced memory utilization on each node by 40%. Reduced DB reads by 75% on the whole. This involved understanding bottlenecks, evaluating, prototyping, implementing the entire clustering solution. It involved making changes to open source libraries. It runs strong today and is serving hundreds of customers. <https://github.com/foxpass/foxpass-pyhelix/commits/main/>
 - Re-architected Foxpass's global RADIUS authentication infrastructure end-to-end. Reimagined the authentication platform to operate behind a global accelerator with lightweight PoPs deployed worldwide, routing users to the nearest instance and significantly reducing authentication latency across geographies.
 - Prototyped/designed/implemented a new line of product that enables chrome books to use us as service for passwordless WiFi authentication. This replaced the old non-scalable ops heavy alternative called GCCC (offered by Google). This new product now spins up an instance without any manual intervention and just by a click of a button. This is proven and has opened up a huge revenue stream for the company. This is the best seamless solution available in the industry.
 - Designed and implemented a new scalable product offering called port based RADIUS proxy from ground up using async python that now enables providing a separate port for each customer to talk to. This is now used in production and used by 100s of customers.
 - Designed and implemented a new scalable product offering called Radsec (RADIUS over TLS) proxy from ground up that now enables us to receive more than 10k connections on a single instance. The product now scales seamlessly and used as an alternative for our old product. Wrote a blog on the same: <https://www.splashtop.com/blog/radius-vs-radsec>
 - Implemented a new product offering called "eduroam" both for UDP and TLS use cases: <https://www.splashtop.com/blog/eduroam-foxpass-secure-global-radius-server>
 - Designed/Implemented the backend for Global lookup feature which helps users access any of our stack (US/EU/AU) seamlessly. Helps new users to be redirected to the platform closer to them and of their choice. To see it in action: <https://console.foxpass.com/login/>
 - Made changes to [open source rate limiter divvy](#) (NodeJS) to accept configs on the fly which is now deployed at scale to rate limit ldap requests
 - Contributed to Golang/python/C open source cryptography library that's used by millions of users around the globe:
 - <https://github.com/smallstep/pkcs7/pull/19>, <https://github.com/bikram990/PyScep/pull/16>, <https://github.com/foxpass/asn1crypto/pull/1/files>
 - Prototype: Enabled proxy protocol parsing in opensource software FreeRadius: https://github.com/foxpass/freeradius-server/tree/dev/grk/parse_pp_poc
 - Introduced DocumentDb for ldap logging purposes. Replaced old non performant logging backend with a complete new backend which made query results to appear 10x faster and with no visible latency from the front end.
 - Created a new python library called 'TTLStore' and available for general use: <https://pypi.org/project/ttlstore/> This is also used in our production for different use cases.
 - Responsible to ensure code quality and successful monthly releases.
 - Tech used: Postgres, S3, Athena, ECS, Docker, Python, datadog, scalyr, rollbar, Django for our console/admin needs.
- Peritus.ai (headquartered in Palo Alto) Jan'18 – Feb'2021 (Designation: Founding Engineer /Tech Lead)
 - 2nd engineer in peritus.ai (Total strength of the company currently: 15); company [acquired](#) by Essenvia
 - Patent: <https://patents.google.com/patent/US10679133B1/en>
 - Peritus.ai provides machine learning insights and automation to scale community intelligence and provide recommendations.
 - Designed and implemented a horizontally scalable multi-tenant continuous data ingestion platform from scratch. Grew the team b 5 engineers. This used s3/google storage, athena/big query and postgres as part of the solution. The data platform is responsible for getting data in realtime from 300+ customer community websites including stackoverflow.com, github.com, discourse.org
 - Created libraries to read data, create tables/views for Athena and GCP programmatically for the data sitting in s3/google storage. Automated (made it repeatable) the whole process of normalizing data from various sources and exposing it through one final view.
 - The data exposed through the final view served as an input for the ML/IR pipeline steps.

- o The whole data ingestion platform was deployed on Kubernetes.
 - o Conceptualised, designed and implemented Rules engine. From wire framing the front end design to leading and shipping the entire project. Implemented the backend completely. This allowed authors to author rules that can be applied to terabytes of machine data.
 - o Implemented a search solution using 100% open source software (apache license) SourceGraph which lets people search code across thousands of repositories. Productionized and integrated it with our existing search solution.
 - o Setup a robust deployment process for processing the data through our ML pipelines so that anyone could do it.
- o Contributed to several components of the code base from Data platform to infrastructure to backend.
 - o Conducted several experiments to evaluate and scale Dgraph. Dgraph was used as a knowledge graph to store machine configs to answer some important questions.
 - o Conducted several experiments to scale solr and came up with a strategy to scale solr
 - o Evaluated kubeflow as a platform for model training and deployment.
 - o Introduced/wrote libraries in our code base so that mongodb could be used as a feature store to store and use the features extracted as part of the ML pipeline step for model training.
 - o Introduced Redshift to process large volumes of data sitting in s3.
 - o Mentored interns, entry and senior level engineers.
 - o Java and Python are the languages used.
 - o Kubernetes was used for the all deployment purposes
 - o Have hands on experience on writing docker files for containerized deployment in kubernetes. Experience in docker-compose/helm charts.
- Fisdome (fisdome.com) Jan'17 – Jan'18 (Designation: AVP, Engineering)
 - o Fisdome raised ~46 million USD and were [acquired](#) recently in an all cash deal for \$150 million.
 - o Built back-end for the wealth management app
 - o Setup engineering best practices (code reviews, test setup (jenkins, jmeter), alerting, monitoring, sentrysetup)
 - o Introduced MySQL based on their data needs which decreased latency of serving the data by several folds [3s to 400ms] and rewrote components due to this change.
 - o Wrote the whole back-end for the do it yourself feature which has seen increased engagement and investments since the launch
 - <https://www.fisdome.com/resources/fresh-fisdome-april/>
 - Also launched top trending funds based on the demand
 - o Introduced graphs in the mobile apps serving from the back-end using high-charts.
 - o Automated payments of sub broker commissions end to end without any manual intervention for raising invoices
 - o Prototyping efforts:
 - Used computer vision library, OpenCV, to crop images. For reducing work load on ops team
 - Prototyped WhatsApp based customer on-boarding using the library yowsup <https://github.com/tgalal/yowsup>
 - o Worked on building an AI chat bot using Google's <https://api.ai/>. This is to automate the repeated customer questions.
 - Before narrowing down on this, prototyped several others (wit.ai, luis.ai, motion.ai)
 - o Google app engine is used for all the infrastructural needs
 - o Mentoring engineers to deliver high quality product Languages used: Python, Javascript
- Uber, San-Francisco Jan'15 – Feb'16 (Designation: Software Engineer 2)
 - o One of the early engineers to join the effort of building the infrastructure team at Uber
 - o 2nd engineer to join the effort of building Kafka platform, helped grow team from 2 -20 in a span of 6 months
 - o Built and scaled the logging/events infrastructure using Kafka (messaging bus) ground up. Built it across US and China and made them inter dc aware
 - o Built 4 clusters each of 16 nodes across US and china data center, each cluster getting a traffic of 30-40k qps
 - o Built monitoring and alerting for the infrastructure
 - o Built logging libraries in Golang to produce messages to Kafka
 - o Migrated the complex kafka 7 architecture to kafka 8 architecture
 - o Wrote tools to consume messages from kafka in a restricted manner between given two timestamps, which is used companywide to consume messages from kafka
 - o Wrote many other tools to help in operational needs of Kafka distributed systems
 - o Got promoted within 6 months
 - o Languages used: GoLang and Python
- Uber, Bangalore Feb'16 – Jan'17 (Designation: Software Engineer 2)
 - o One of the first engineers to join the Bangalore Uber team
 - o Helped Uber setup engineering in Bangalore. Helped grow from 0-20 engineers
 - o Designed, implemented and successfully shipped the vehicle leasing product ground up (Platform to provide leasing opportunities to drivers)
 - o Introduced Kafka, samza, spark in Uber Bangalore to solve the leasing problem in India
 - o Designed and implemented a framework to scale vehicle leasing service using SWIM protocol (gossip variant) and sevnup (<https://github.com/uber/sevnup>)
 - o Wrote technical specs to evaluate GPS vendors.
 - o Vehicle leasing service was built using Golang
- NVIDIA, Santa Clara May '14 - Aug'14 (System Software Intern in Android Kernel Development Team)
 - o Designed and Implemented a framework in Android kernel to analyze and co-relate kernel events with architectural events. Generated visualizations using the data generated from the framework, identified hot-spots and pointed out architectural anomalies when co-related with the kernel events
- Graduate Student Researcher (Systems Research Group in UIUC) July '13 – Dec '14
 - o Built infrastructure to analyze heap growth pattern of an application. Memory footprint of an OS running on a VM is extracted using virtual machine introspection tools provided by Virtual-box. Heap dumps of the applications from the memory image is extracted using a tool called volatility. Raw heap is then reconstructed to visualize the data structures. This reconstruction helps us understand the transformations that the heap undergoes at various points of time.
 - https://github.com/grkrthk/heap_analyzer_new

- http://tcipg.org/research_Forensics.html
- Research Poster: http://tcipg.org/sites/default/files/posters/Poster_IW2013_Forensics.pdf
- https://tcipg.org/sites/default/files/posters/poster_iw2014_forensics.pdf
- NetApp India, Bangalore, India July '09 – July '13
 - Development experience in kernel and distributed file system called WAFL (write anywhere file layout).
 - Authored 4 technical invention papers and have been awarded Patent incentive award for all of them.
 - Have worked on file system recovery issues and have provided patches aiding file system recovery.
 - Have worked extensively in SAN area. Have a good exposure in SCSI and FC protocol. Have provided root cause analysis for interoperability issues and have provided fix for the same.
 - Delivered tech talks, mentored interns, have been awarded for building a tool called i-sense which saved a lot of engineering resources. This tool automated the file system damage assessment and recovery time. Languages used: C
- Projects executed at UIUC
 - HEVC Intra Frame Prediction using GPU: Language CUDA
 - 'Helenus – ' Distributed Key-Value store: Language: C
 - [Add constraint feature in sqlite3](#): The project adds a new functionality ADD CONSTRAINT to open source DBMS, SQLite3
 - [Mitigating Impact of Heterogeneity Across Power-constrained Nodes on Parallel Applications through Load Balancing](#)
 - [Growth Pattern and Pointer Analysis of Heap to Detect Security Compromises on Generic Applications](#)