[amruthasuri@gmail.com](mailto:amruthasuri@gmail.com)

[www.linkedin.com/in/brsuribabu](http://www.linkedin.com/in/brsuribabu)

Github:

<https://github.com/sb-cloudscience>

<https://github.com/sb-runtimes-sh>

<https://github.com/brsuribabu>

<https://github.com/amruthasuri>

<http://bit.ly/2xdISRC> ( Github)

[Experience Highlights](#_heading=h.gjdgxs) **3**

[Agile Leadership](#_heading=h.30j0zll) **3**

[Skills](#_heading=h.1fob9te) **3**

[Java8-14](#_heading=h.3znysh7) **4**

[Java Virtual Machine ( JVM )](#_heading=h.2et92p0) **4**

[SOA Services](#_heading=h.tyjcwt) **5**

[# 1 Enterprise Common Services](#_heading=h.3dy6vkm) 5

[Spring](#_heading=h.1t3h5sf) **5**

[#1 Web Applications ( Server side and support for AJAX )](#_heading=h.4d34og8) 5

[#2 Batch Jobs](#_heading=h.2s8eyo1) 5

[#3 DB Pollers](#_heading=h.17dp8vu) 5

[#4 Authentication and Authorization Filters using DB/IAM services](#_heading=h.3rdcrjn) 5

[#5 JMS Listeners and publishers](#_heading=h.26in1rg) 5

[Restful APIs](#_heading=h.lnxbz9) **5**

[#1 E-Commerce APIs](#_heading=h.35nkun2) 5

[#2 Master/Metadata API](#_heading=h.1ksv4uv) 6

[#3 Domain Services API](#_heading=h.44sinio) 6

[#4 Integration Gateway APIs](#_heading=h.2jxsxqh) 6

[Search Engine using SOLR/Lucene](#_heading=h.z337ya) **6**

[#1 : Search engine for the matters and the related documents using Lucene](#_heading=h.3j2qqm3) 6

[#2 : Search engine for the company information](#_heading=h.1y810tw) 6

[#3 : Search engine for the events information](#_heading=h.4i7ojhp) 6

[#4 : Search engine for the user locations search - nearby](#_heading=h.2xcytpi) 6

[Angular JS](#_heading=h.1ci93xb) **6**

[# 1 ECommerce App](#_heading=h.3whwml4) 6

[MongoDB](#_heading=h.2bn6wsx) **6**

[#1 Masters and metadata management](#_heading=h.qsh70q) 7

[#2 Event data management](#_heading=h.3as4poj) 7

[#3 Product Catalog management](#_heading=h.1pxezwc) 7

[Tools Development](#_heading=h.49x2ik5) **7**

[Integration experience](#_heading=h.2p2csry) **7**

[Security](#_heading=h.147n2zr) **7**

[Amazon Web Services](#_heading=h.3o7alnk) **8**

[Google Cloud Platform](#_heading=h.23ckvvd) **9**

[Cloud Data Integration](#_heading=h.ihv636) **9**

[Scalability, Availability, Reliability, Monitoring and Performance](#_heading=h.32hioqz) **9**

[Continuous Integration/ Continuous Delivery ( DevOps )](#_heading=h.1hmsyys) **9**

[Infrastructure Automation](#_heading=h.41mghml) **9**

[Testing ( Unit/Integration/Regression)](#_heading=h.2grqrue) **10**

[Microservices](#_heading=h.vx1227) **10**

[Microservices and ServiceMesh](#_heading=h.3fwokq0) **10**

[DevOps](#_heading=h.1v1yuxt) **10**

[Serverless Architectures](#_heading=h.4f1mdlm) **10**

[Kubernetes and Docker](#_heading=h.2u6wntf) **10**

# Experience Highlights

**18+ years of experience in the architecture/design/development of Enterprise applications(web/mobile)/Cloud services using Java, JEE and other integration technologies in the lines of SOA/Micro Services and Event driven architecture models.**

**As a Java/J2EE application architect**

* Experience in architecting/designing the SAAS/Enterprise applications (web/mobile) using the SOA/Micro services and event driven architectural styles.
* Experience in delivering the enterprise applications from requirements to delivery
* Experience in the building application search using SOLR/Lucene
* Experience with NoSQL stores like MongoDB
* Experience of nearly 5 years as an architect working across teams in the organization
* Experience Enterprise Application and Integration Patterns, OO Patterns, J2EE design patterns
* Code Quality governance across teams using Sonar/PMD/FindBugs
* Setup Code Quality guidelines/best practices relating to development/design/architecture
* Architectural/Design reviews for various projects
* Experience with Amazon Web Services ( AWS )

# Agile Leadership

* SCRUM and Kanban Sprints using VersionOne and JIRA Agile – Experience with Sprint Planning, Sprint Burndown, Sprint Retrospective to Backlog
* GitOps with GitHub and GitLab for code repositories with Continuous Integration and Continuous Delivery with Jenkins, Github actions and GitLab CI/CD
* Continuous Integration with Jenkins, GitHub actions and GitLab with unit tests, code quality review using Sonar Qube, PMD, FindBugs, CheckStyle
* Continuous Delivery using Jenkins, GitHub actions and GitLab with automated regression tests using Selenium, automated user acceptant tests using FitNesse and Geb functional testing frameworks
* Automated Infrastructure Provisioning with DevSecOps with Ansible, Chef, Puppet

# Skills

**Languages**

Java, C, HTML, Javascript, CSS, SQL, PL/SQL, XML ( XML Schema/XPATH/XSLT/XSLFO ), JSON ( JSONPATH)

**JEE Components**

Servlets/JSP, JSF, EJB, JAX-RS, JAX-WS, JAXB, JMS etc.

**Frameworks**

Spring Boot, Spring Data (Mongo/SOLR/JPA), Spring Integration, Spring Batch, Spring MVC, Spring Web Flow, Spring ORM, Spring JDBC, Spring JMS, Spring Cloud, Spring REST HATEOAS, Struts, Hibernate, Drools, Camel etc.

**Javascript Frameworks** - Angular JS, ReactJS, JQuery, Twitter Bootstrap etc.

**Tools/Libraries**  
 Joda datetime, Apache Commons, Dozer, Jackson, etc.

**Configuration management tools**

Git, SVN, VCS etc.

Github, GitLab

**OS Platforms**

Linux(Ubuntu/RHEL), Windows

**Databases**

SQL : Oracle, MySQL, MariaDB

NOSQL : MongoDB, Redis

**Search**

SOLR/Lucene

Elastic Seach Engine

**Messaging queues**

Sonic MQ

Weblogic

IBM MQ

Active MQ

**Streaming Platforms**

Apache Kafka

Amazon Managed Service Kafka

**Security**

Spring Security OAuth2

Oracle Access Manager

OWASP Security guidelines and practices

**Deployment**

Amazon Web Services

Application Servers : Weblogic, WebSphere , Spring TC, Tomcat

**Monitoring**

HP Sitescope, Nagios, Splunk etc.

# Java8-14

Java 8 - Functional Programming Streams and Lambda Expressions, Type Inference

Java 9 - Modules Reactive Streams HTTP2

Java 10 - java.util new features with Optional and Collections

Java 11 - String and HTTP Features

Java 12 - Switch Expression String Features Stream Features

Java 13 - Text Blocks Switch Expressions Enhancements

Java 14 - Switch Expressions (Standard) – JEP 361

Pattern Matching for instanceof (Preview) – JEP 305

Helpful NullPointerExceptions – JEP 358

Records (Preview) – JEP 359

Text Blocks (Second Preview) – JEP 368

# Java Virtual Machine ( JVM )

* Experience with deep understanding of JVM internals
* Garbage Collectors tuning for latency and throughput
* Experience with Bytecode instrumentation and manipulation with Dynamic proxies, ASM and CgLIB
* Experience with reverse engineering with bytecode
* Experience in turning the JVM memory model for GC tuning for better latency and throughput

# SOA Services

## # 1 Enterprise Common Services

* Enterprise common services like Billing/ Ordering/ Subscription/ Payment/ Customer Information etc are implemented using SOAP/AXIS2 and Spring Web Services(JAX-WS)
* These services are exposed using the enterprise service bus Oracle Service Bus as proxy services for the various clients/apps in the enterprise

# Spring

## #1 Web Applications ( Server side and support for AJAX )

* Web applications are build using Spring MVC/Webflow
* Web controllers returning both html/json using Spring MVC annotations

## #2 Batch Jobs

* Batch jobs for uploads etc using Spring Batch using the transaction control
* Retry jobs for various failures in the applications like events process failures etc.

## #3 DB Pollers

* Polls on to the database using Spring Integration pollers and executes the Spring integration flows talking to the various application components like order status change in db reflecting in various other systems.

## #4 Authentication and Authorization Filters using DB/IAM services

* Security Filters,AuthenticationProviders and role based access management voters including hierarchical roles. using Spring Security
* User services using DB/Oracle Access manager and Spring Security

## #5 JMS Listeners and publishers

* Spring JMS listeners for various application events for both queue/topic based and with durable subscription
* Spring Integration JMS inbound and outbound adapters for listening and publishing the JMS events.

# Restful APIs

## 

## #1 E-Commerce APIs

* API backend is developed using micro services style around the business domain to be consumed by the various channels web/mobile etc.
* the APIs for the product catalog/pricing/payment and other steps in the ecommerce transaction
* Implemented as Level 2 services
* Transaction flows tried to be in Level 3 services (**HATEOAS**)
* The APIs are implemented using Jersey/Spring.

## #2 Master/Metadata API

* Single API which serves the needs of the master/metadata for the various web/mobile apps
* The APIs are implemented using Jersey/Spring.

## #3 Domain Services API

* the Restful API for the domains registration transaction flow like .com/.org/.net registration.
* The APIs are implemented using Jersey/Spring.

## #4 Integration Gateway APIs

* The Integration Gateway APIs integrates various SOAP/REST web services from various providers in unified manner for aggregation/comparison of prices and for the transaction fulfillment for a E Commerce Project.
* The APIs are implemented using Jersey/Spring

# Search Engine using SOLR/Lucene

## #1 : Search engine for the matters and the related documents using Lucene

* the full text search is build using Lucene to index the richer documents like word/pdf etc.
* the index is around of size 1TB including the documents
* hot backups using SnapshotDeletionPolicy
* re-indexing jobs for the matching specific criteria

## #2 : Search engine for the company information

* full text search implementation for the companies information nearly million records

## #3 : Search engine for the events information

* full text search implementation for the events information with richer schema including dynamic fields

## #4 : Search engine for the user locations search - nearby

* geospatial search using the SOLR geospatial search bounding box search for the specified location (lat/long)
* the user locations are indexed from the mobile app location change events.

# 

# Angular JS

## # 1 ECommerce App

* Angular JS based web app which consumes the REST apis as a single page application

# MongoDB

## #1 Masters and metadata management

* Managing application masters and metadata for overgrowing fulfillment engine integrations with the various providers using MongoDB as schema less(flexible schema) data store.
* application repository layer using Spring Data MongoDB

## #2 Event data management

* managing the events information with the richer schema handing the custom attributes at the various levels company/user level.
* application repository layer using Spring Data MongoDB
* sync the events data to the events index stored in the SOLR

## #3 Product Catalog management

* Product catalog management for ECommerce system with flexible schema handling the category specific attributes and the dynamic schema
* Application repository layer is built using Spring Data MongoDB with custom converters

.

# Tools Development

**#1 Code generation engines**

* JSP/HTML generations from the workflow engine definitions
* Graph/SQL markup/statements generation using JDBC/velocity template engine

# Integration experience

* Integration gateway using SOAP/REST services integrations.
* Application Integrations using Weblogic JMS/Sonic MQ/IBM MQ
* Integrations with third party systems like Oracle EBS, Siebel CRM, and SalesForce etc.
* MongoDB – Solr Search Server integration
* RDBMS– Solr Search Server integration
* Integration with Document management system like Documentum
* Integration with COM components jCom bridge
* EAI through the RMI and Remote EJB-SLSB ( RMI/IIOP) and EJB-MDB
* EAI through Spring Integration with DB pollers and JMS adapters
* Web Scraping on the State Websites for data/creating the simple workflows from the State Websites

# Security

* Microservices security using the API gateway using Kong and Okta Identity Platform
* Microservice security using OAuth2 and OpenID Connect for authentication and authorization
* OWASP Top 10 Security Vulnerabilities with Web Application Firewall and Spring Security
* Experience with Security attacks like DDoS, Phishing etc
* Spring Security authentication flows for form login/anonymous/token based authentication and remember me services.
* Spring Security custom filters and authentication providers implementation
* Spring Security access management using roles ( granted authorities ) and hierarchy of roles and voter based access management - URI authorization - Functional security
* Functional Security through custom tags enabling/disabling the action buttons in the UI page.
* JAAS authentication and authorization
* SSO libraries for the client applications to authenticate and authorize
* Authentication flows in ESB for secured access
* OWASP Top 10 security guidelines implementation  - SQL Injection, XSS, CSRF, Session Hijacking etc.
* PCI security standards implementation
* Security testing using OWASP Zap Security tool and mitigating the risks like SQL injection, remote code execution ( shell scripts) etc.
* WSO2 IAM server experience for XACML/SAML/OAuth2
* REST API authentication interceptors implementation using Jersey.
* Data Security – PCI/DSS for payment card information encryption and masking
* Data Security – GDPR
  + Authentication through MFA and Geo restrictions for authentication through OpenID Connect
  + Access Control through OAuth2 with Geo restrictions
  + Fine grained access control for the resources through attribute level access control
  + Data Privacy through the storage selection
  + Data Privacy through the secrets/keys management for encryption for data at rest, data in transit and data stored for access
* Data Security - CCPA
* Information Security – ISO 27001
* Encryption and cryptography standards with SSL/TLS, RSA, JWT Tokens using libsodim, JWT, Amazon KMS

# Amazon Web Services

* Message transmission between on-premise systems to the cloud SAAS to the apps using Amazon SQS
* Amazon S3 for Amazon SQS large payloads
* Amazon Route53 for DNS management
* Amazon SES for transactions emails
* Amazon EC2 for Cloud services hosting
* Amazon RDS for Oracle using Multi-Zone Availability
* Amazon Cloudwatch for alerts
* Amazon Elastic Kubernetes Engine
* Amazon Elastic Container Service
* AWS Fargate
* VMWare Cloud on AWS
* Amazon API Gateway and Amazon App Mesh using Envoy-Proxy
* AWS IAM and Federated IAM providers
* AWS and Thales E-Security (Vormetric)
* AWS Web Application Firewall
* AWS and F5 Web Application Firewall
* Amazon CloudWatch with ELK
* Amazon CloudWatch with Splunk
* AWS Secrets with Hashicorp Vault
* Amazon Redshift and Amazon Redshift Spectrum
* Amazon S3 and S3 Datalake Formation
* Amazon Athena
* AWS Snowball Family
* Amazon Transit Gateway
* Amazon Direct Connect
* AWS Outposts

# Google Cloud Platform

* Serverless with Cloud Run, Cloud Functions
* Google Anthos for Hybrid Cloud and Multi-Cloud
* Google App and Compute Engine
* Google Kubernetes Engine
* Google BigQuery

# Cloud Data Integration

* Informatica Data Integration with Amazon S3 Data Lakes and Amazon Redshift Spectrum using AWS DataSync
* AWS Glue Crawlers for Data Catalog and ETL process with Spark Jobs underneath
* Amazon Athena interactive queries
* Amazon EMR for data processing and analysis
* Amazon S3 and Amazon Redshift Data Integration for Eurex and Xetra datasets
* Google BigQuery integration with Ocean datasets ( POC)

# Scalability, Availability, Reliability, Monitoring and Performance

* Experience with scaling the servers on-promise and on the cloud using AWS to support the transactions load
* Monitoring the applications and services using Splunk, ELK, Nagios, Amazon Cloud Watch, HP Quality center etc
* Production debugging and tracing the code with Splunk, ELK and OverOps
* Experience with performance profiling, debugging with JProfiler, JProbe, New Relic, App Dynamics etc
* Amazon RDS scaling with Multi-Zone availability
* Database tuning with indexes and queries tuning
* Caching with EHCache, JCache with Spring Cache abstractions improve the latency across the requests and within the request
* Experience with Capacity planning and auto scale with Amazon Load balancing strategies

# Continuous Integration/ Continuous Delivery ( DevOps )

* CI/CD with Jenkins, Anthill and Gitlab
* Experience with deployment pipelines containing unit tests,integration tests, functional tests, security testing using Sonar Qube
* Experience with container deployments using Docker and Kubernetes
* Experience with linux shell and bash scripts and windows batch scripts

# Infrastructure Automation

* Terraform for Hybrid and Multicloud
* Amazon CLI for bash/powershell scripts
* Amazon SDKs for Python/Java
* Redhat Ansible

# Testing ( Unit/Integration/Regression)

* Unit testing with Junit, TestNG and Mockito
* Integration testing with Spring Tests and TestNG
* API regression tests using SOAP UI for SOAP webservices
* API regression tests using SOAP UI for RESTful API
* REST API unit tests using PostMan

# Microservices

* Design the Microservices using Domain Driven Design – Domains, Subdomains, Bounded Contexts and Context Maps
* Microservices Orchestration and Choreography using API Gateway and Synchronous, Messaging and Event driven architectures using Amazon API Gateway , Kong API Gateway
* Microservices continuous integration and continuous delivery using Jenkins, GitLab and Github
* Microservices runtimes using GCP GKE, Amazon EKS, Amazon ECS, AWS Fargate and Azure AKS, Redhat Openshift, VMWare Tanzu for Cloud, Hybrid clouds and onpremises clouds
* Microservices runtimes using Kubernetes and Dockers for on premises infrastructure with VMWare Tanzu and Redhat Openshift

# Microservices and ServiceMesh

* Amazon EKS with Istio and Prometheus and Grafana
* Amazon EKS with ELK stack for observability
* Google Kubernetes Engine with Istio and Stackdriver
* Google Kubernetes Engine with Istio and Prometheus

# DevOps

* Terraform for AWS Fargate, Amazon EKS, Amazon ECS, Google GKE, Google CloudRun
* Ansible
* Chef
* Puppet

# Serverless Architectures

* AWS Lambda and AWS Lambda Edge
* AWS Fargate
* Google Cloud Functions
* Google Cloud Run
* KNative
* Serverless Framework
* Spring Cloud Serverless Framework

# Kubernetes and Docker

* Amazon EKS for containerized microservices and applications
* AWS Fargate with Amazon EKS for containerized microservices and applications with serverless
* Google Kubernetes Engine for containerized microservices and applications
* Google Cloud Run for serverless applications
* KNative for serverless functions and applications
* Service Mesh integrations with Istio and envoy
* Managed Services – VMWare Tanzu and Redhat Openshift
* PAAS – CloudFoundry and Heroku
* Managed Kubernetes on Digital Ocean