

# Ji Liu

✉ 709327148@qq.com · ☎ (+86) 150-0019-3830 · 🌐 GreyPlane ·

## ABOUT

---

I'm a self-learned programmer who has a bachelor's degree in Communications, working across almost the whole lifecycle of modern Web application development(frontend, backend, bigdata, devops), but especially focusing on Functional Programming and Stream Processing. And with some academic interest in Programming Language Theory and Distributed Systems, read some papers, and do a little bit of research as a hobby.

## WORK EXPERIENCE

---

### Convertlab Inc.

06/2020 – 03/2021

(Software Engineer) Application Team

- develop a low-code form design application which allows user to use simple drag and drop method creating complex form with relation between questions that can be later rendered on mobile devices.
- application development based on React and Redux, with UI component library Ant Design, using Typescript.
- integrate Typescript toolchain into the existing webpack build pipeline, enable our team to migrate to Typescript gradually.
- develop UI components library based on ant-design.

### Convertlab Inc.

04/2021 – 08/2023

(Software Engineer) Data Team

- the only two developers who create and maintain our own Flink-like streaming processing platform which is based on Akka and Scala, and uses MongoDB as persistence layer.
- often in charge of design, implementation whole project or module no matter which part of technology it uses or area it belongs, and report to high level managers directly.
- regular application development and maintaining(in both frontend and backend with multiple programming languages) for various modules across whole product.
- design the required schema for various storage and implement analytical SQL for Impala or batch process job(Spark).

## PROJECTS

---

### Reactflow

a low-latency, high-throughput streaming processing system, represent the whole journey of a customer in a campaign which is the implementation of our clients' marketing strategy(when customer did what and then we should do what if he meets some criteria), also coordinate other functionalities of application to meet the goal.

- the selling point of our main product - digital marketing cloud DMHub.
- able to support thousands of instances and thousand million of customers running simultaneously and stably.
- enhancing existing product with the ability to choose multiple criteria for customer to enter the flow and set arbitrary number for the maximum reentrance times for one customer at a flow.
- implement an algorithm to validate topology definition of a flow such like if it has cycle or other our own business logic requirements with slightly modified topo sort.

### Knowledge Graph

according COO's schedule, create a platform for saving and editing our knowledge into databases, which can be retrieved by certain pattern, in such we can build company's knowledge system that reducing the efficiency lost between knowledge transferring therefore increase productivity.

- research the whole concept of so called knowledge graph and how to use it to represent our goal practically.
- represent every concepts as ontology(works like category of things) and entity that belongs to and constraints by certain ontology, and modelling it by two separate labelled property graph which store in Neo4J.
- implement a graph visualization front-end application that support showing, searching, modifying, connecting entity graph by React, Typescript and AntV.
- implement the backend service based on Neo4J which utilize Neo4J's ability to import ontology definition from Protégé, and use mostly only Cypher query language to implement CRUD functionality of entity graph, featuring complex graph joining because every modification on entity graph must be validated by checking ontology graph, also implement basic search logic such like shortest path between two entities according to a certain type connection.

### Message Engine

a middleware that unifies the approach of sending through sms or other channel that can reach out end users, made the message sending process to be robust and reliable, and the integration of third-party service provider much easier, also automatically get the common features like batching request or delay the sending if it was triggered at a time that will disturb customers.

- design the high-level architecture, divide whole project into server side and sdk library for client usage, and utilize Kafka Streams to implement stateful functionality and guarantee the effective exactly once delivery semantic.
- implement the batching logic using Kafka Streams, optimize the performance by designing our own data schema that working directly on bytes instead of ser/der JSON everytimes when saving or retrieving message from kafka state store.
- implement the disturbing time check and delay functionality, using HBase as storage.
- research how to reduce the duplication caused by Kafka rebalance with low cost, implement it into our own kafka consumer wrapper that used by sdk.

## Data Permission

working directly with CTO, reimplement our way to apply permission control to data accessing, make it more like a standard ABAC system.

- implement a SQL AST tranformation pass which compile original SQL into permission applied SQL, which include more clauses for checking if data was accessible for current user or department, or if data was sharing to the current user manually.
- implement several Spark jobs for computing shared permission based on user customized rule.

## PERSONAL PROJECTS

---

### AlgebraicGraph

<https://github.com/GreyPlane/algebraic-graph>

algebraic graph implementation in Scala

- read, implement both associated type families and plain algebraic data types encoding and bunch graph opearations describe in paper using Scala.
- Using encoding describes in this paper to build a library for constructing the topology of our Flink-like component programmatically, is an elegant solution because the good algebraic properties holds by the encoding, implement both TypeFamilies-based and ADT-based encoding in Scala.

### A=B

<https://github.com/GreyPlane/AeqB>

interpreter for esolang A=B in Haskell

- Using Free Monad and implements via the approach that described in paper data type a la carte.

## SKILLS

---

- **Program Language:** multilingual (not limited to any specific language), comfortable with Haskell Typescript Scala (in random order).
- **Web Frontend (React):**
  - Experienced in React and Hooks API.
  - Experienced in Typescript
  - Experienced in the ecosystem around React, includes but is not limited to routing, state management, query library, can delivery project independently.
- **Backend:**
  - experienced in using Scala and Akka ecosystem's most commonly used components like actors, streaming, HTTP to develop Web application or middleware. and also Akka contributor.
  - experienced in Cats ecosystem libraries, Knowing how to do generic programming by using Shapeless.
  - experienced in using Groovy and Grails, GORM to develop Web backend service.
  - experienced in using Java and Spring, Springboot, Mybatis, Mybatis plus and JPA, also use Google guava as util library to develop Web backend service.
  - experienced in using MySQL, Redis, Elasticsearch, MongoDB as application storage, also Kafa as Message Queue.
  - understanding Microservices and containerization technology like K8S and Docker, can delivery whole project independently.
- **BigData:**
  - have solid understanding of basic topics about the distributed system such as consensus algorithm, delivery semantic, read some of founding paper in this area.
  - have deep knowledge of Kafka, also have researched several topcis on streaming processing and have read related papers and books.
  - use components in Hadoop ecosystem like HBase, HDFS, also Impala, Kudu.
  - write spark job to process data, in both RDD API and Spark SQL, have solid understanding of how spark works on the high-level.
  - certificated Cloudera Hadoop System Administrator.
- **Functional Programming:**
  - Capable to adapt functional programming technique in the real world project.
  - Able to solve the complicated problem that involves advanced topic about type theory or category theory.

- **Evangelist:**
  - teaching Typescript for colleagues, helping them to solve problems associated with types.
- **CS Basic:** learned CSAPP and SICP, has solid CS foundation, also learned some PL theory.
- **Development Tool:** Can adapt to any editors/OSs, usually use Vscodc, IntelliJ, Emacs under MacOS.

## EDUCATION

---

College of Publication, **Shanghai Publishing and Printing College** 08/2015 – 06/2018

Bachelor of Communications, **University of Shanghai for Science and Technology** 08/2018 – 06/2020

## MISCELLANEOUS

---

- Languages: English - fluent
- **3 kyu** on CodeWars, primarily in Haskell, Agda and Idris