# Yong Wei

*Make changes*

Email : weiyong1024@gmail.com
Mobile : (+86) 173-7657-1024

## Education

- **Master - EE - Beihang University** Beijing, China
  *Exempt Entry Exam; Finish courses and research publishing mission during the 1st semester* *Sep. 2016 – Jan. 2019*
- **Bachelor - EE - Beihang University** Beijing, China
  *Exempt NCEE; Honors College(top 50 among 3000+ freshmen); GPA 3.7* *Sept. 2012 – Jun. 2016*

## Competitive programming

- **Google Code Jam Kickstart 2017**
  *Top 5%, rank 108th globally.* ***scoreboard, id:WeiYong1024***

## Career

- **Mindverse AI(Company of the LLM agent platform MindOS)** Hangzhou, PRC
  *Infra Tech Lead (4-person team), System Design, Kubernetes, Java, OpenAI, Azure, Tencent Cloud* *Jun. 2022 - Now*
  - **Technical Responsibilities - Extreme Attention to Detail, Pursuit of Best Engineering Practices**
    - **AiInfra - The foundational layer for large models in business applications**
      - **OpenAI Base Layer:** Designed and built mid-tier services to manage OpenAI and Azure OpenAI assets, encapsulating their services. Responsibilities included, but were not limited to, managing Azure OpenAI service regions, instances, deployments, and model versions, as well as pooling OpenAI accounts with proactive/passive health checks, weight distribution, and other availability mechanisms.
      - **Mutil-LLM Encapsulation:** Provided encapsulation services for the algorithm team, integrating third-party SaaS large models such as OpenAI GPT, Google Gemini/Palm2, and implemented cost monitoring for various business lines and features.
      - **GPU Computing Scheduling Platform:** Researched and designed a GPU compute scheduling platform for training tasks as a technical reserve, which was not implemented due to the company's strategic focus on AI-Agent layer only.
    - **Production Tools & DevOps - The complete set of production environment and tools**
      - **Multi-cluster Environment Management:** Designed and built multiple environment cluster architectures based on Kubernetes; containerized all microservices and deployed them on Kubernetes; provided a Helm-based distributed GitServer (Gitlab), and developed a company-wide code review mechanism based on OwnershipReview and ReadabilityReview to strictly ensure the engineering maintainability of the entire codebase; constructed a distributed CI/CD pipeline (Jenkins) using Helm, Jenkins, Python, Shell, etc., which includes packaging, approval, deployment, and regression, with scalable parallel build and release capabilities.
      - **Production Tool System Construction:** Selected, designed, and built internal production tools. Responsible for internal networking, internet access management; unified distributed configuration centers (Apollo, Nacos), rate limiting (Sentinel), etc.; set up operation tools (Rancher, JumpServer, Octant); managed offline data tools (MySQL, Redis, Clickhouse, Superset, Grafana); and logging and application performance monitoring tools (Loki, DataDog, Tencent Cloud CLS, Skywalking). Provided all necessary tools for technical research and development.
      - **Cloud Resource Management:** Managed the lifecycle of company's IaaS, SaaS resources, user permission management, and controlled costs for cloud and third-party services. Designed, created, and maintained cost dashboards.
    - **Data & BI —Managing data assets, experimental tools, data-driven decision making**

- **Online and Offline Data System:** Built and maintained an online/offline data stream based on OLTP (MySQL) -> CDC (external supplier) -> OLAP (Clickhouse) -> DataVisualization (Superset+Grafana), used for data development, anonymization, etc., and an asynchronous data stream based on message queue (RocketMQ) -> basic service (Springboot), for user behavior data analysis. Served as a dashboard and reporting tool for product, development, and operations teams.
- **A/B Testing Tool System:** Selected, designed, built, and maintained a complete experimental tool system including experiment configuration/result analysis platform (GrowthBook) + frontend tracking (GoogleAnalytics) + backend tracking (implemented in Springboot, including engineering and algorithms). Supported experiments and feature toggles for front-end, pages, and algorithms.
- **Product Privacy Compliance:** Implemented a series of privacy compliance standards based on Vanta, including data anonymization, internal training, etc., to ensure the company's product MindOS met USDP and GDPR standards.
- **Business Layer Development - Providing fundamental services for business development**
  - **Middle Platform Services:** Designed and built mid-tier basic microservices, offering common foundational services to business systems like backend encrypted storage, user privacy database, web crawlers, offline data warehouse management, etc., and delivered them in forms of RestfulAPI, secondary libraries + RPC interfaces, etc.
- **Business Support**
  - Led mobile app mini-programs and web product MindOS's subsystems & full-link load testing, and safeguarded the global launch processe; prepared and established SOPs for toB public cloud/private cloud/private deployment; supported service architecture migration across countries/cloud regions as per business needs.
  - Promoted a company-wide blameless postmortem culture, organized safety production weekly meetings.

- **Alibaba Cloud** Hangzhou, PRC
  *Senior Engineer, Backend, Java, Springboot, Alibaba Cloud* *Aug. 2020 - May. 2022*
  - **Traffic Scheduling Middleware:** : In the group's high-availability team, responsible for Alibaba Cloud's internal traffic scheduling middleware. Utilizing Alibaba's internal PandoraBoot (an enhanced version of Springboot), provided container granularity second-level health monitoring and3$\sigma$outlier detection on the cluster. Leveraged the weight table mechanism of Alibaba's internal RPC framework HSF for minute-level traffic scheduling and recovery linkage of microservices. This system prevented cluster avalanches caused by single-point failures and was also used for JIT preheating. During my tenure, the product was integrated into all core applications of Alibaba Group, supporting over 400,000 containers, and served as a core security component safeguarding the peak traffic of 600,000 QPS during the 2020 Double Eleven shopping festival.
  - **Private Cloud Management Platform**: As part of the P8 team, the goal was to build a private cloud management system for government scenarios. Developed using the Java framework Springboot, the system is divided into three subsystems: application operations, resource operations, and general management. I was responsible for backend development of the application operations center subsystem. The subsystem’ s base layer included cloud resource listing, monitoring and inspection, usage statistics, custom interface inspection, SQL inspection, etc. On top of this, we built expert reports such as slow SQL statistical analysis, business dashboards, and health check reports. This system is currently commercially deployed in over 6 locations nationwide, supporting projects like elementary school admissions in a provincial capital city in 2021 and nucleic acid testing during the 2022 Spring Festival.
  - **Lightweight Delivery for Government Cloud Management**: As a new P9 team member, the goal was to deliver private clouds for government use in 7 regions. Initially, developers used the Rainbond interface to deploy microservices from different teams onto customer K8S clusters, but the complicated configuration led to long delivery cycles. To address this, I developed a lightweight deployment delivery solution using docker-compose, based on generic environment initialization scripts and delivery step documents. By maintaining environment variables for different customer sites, we centralized control and integrated over 20 microservices from various older teams. I edited the WBS and SOP, organized training for outsourced staff on the delivery process, and trained outsourced individuals could deliver or upgrade a site in 2 days. With this solution, the team’ s monthly commercial output capacity increased from single to double digits.

- **Pony.ai** Beijing, PRC
  *Engineer, Infra, C++, Python, Linux, Bazel, Strict Code Quality Standards* *Feb. 2019 - Mar. 2020*
  - **Driving Recording Toolchain**: Autonomous vehicle road tests required an accompanying engineer to record issues via an external keyboard. To eliminate the need for this, a driving recording toolchain was developed using microphones and physical buttons as the hardware solution. Firstly, an input device interface library evdev, licensed under BSD, was integrated into the Bazel main project. Based on this library and Linux's ALSA audio driver, a daemon was written responsible for hardware detection, listening for signals from external buttons and the vehicle system process's pipeline, triggering and stopping microphone recording, and storing issue information as audio files on the industrial computer during the drive. In the data processing stage, a speech-to-text service was set up on GCP. Python scripts were used to extract the contents of the audio files, and Google's Protobuf tool was used to serialize the original information into the QA data stream.
  - **Onboard Car Voice System**: The original onboard voice module used Google's Pico TTS library to play hardcoded voice content text in real-time during driving, leading to issues of language monotony and repetitive consumption of computing resources for TTS processes. The new onboard voice system uses an audio file-based workflow, aimed at reducing the computational load on the vehicle's industrial computer and supporting voice I18N. For this, 1. A Python-based internal CLI tool, `car_sound_utils` (referred to as CLI), was developed for engineers to create and modify existing corpora and manage voice pack versions. First, an entire TTS service built on AWS using CloudFormation and Lambda functions was encapsulated in the CLI for creating audio commands. Then, commands for uploading and downloading voice files and uploading voice packs to the internal storage server were added to the CLI, along with a caching mechanism to speed up uploads and downloads. 2. In the onboard system, the main process initializes the voice module using Linux's ALSA driver to load the audio stream into RAM, and plays it when the voice module receives a playback request message.

## Internship

- **Airbnb**: In the summer of 2018, Airbnb expanded into the Chinese market, and as one of the first 8 interns in the Mainland China office, developed the domestic version of the annual host review page with full-stack tech.

- **Megvii(Face++)**: During the cooperation between Megvii and VIVO in early 2018, participated in the development of the facial recognition module for the X21 model and conducted mobile model search optimization based on InceptionV3(a type of CNN) for mobile devices.

## System experience & Tech stack

- **System experience**: LLM agent platform, Cloud computing, HA, DevOps, Self driving vehicles, etc.

- **Tech stack**: Java, Python, Kubernetes, C++, etc.