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 1^{st} 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* 108^{th} *globally.* ***scoreboard, id:WeiYong1024***

Career

- **Mindverse AI(Startup 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 and3σoutlier 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.

魏雍
Make changes
weiyong1024@gmail.com
(+86) 173-7657-1024

教育背景

- **硕士 - 北京航空航天大学 - 电子与通信工程** 中国，北京
  保研; 研一上学期发核心期刊达毕业标准; 北京市优干 *2016 年 9 月 – 2019 年 1 月*
- **本科 - 北京航空航天大学 - 电子信息工程** 中国，北京
  物理竞赛保送; 沈元荣誉学院 (入学 top50/3000+); 北京市三好; GPA 3.7 *2012 年 9 月 – 2016 年 6 月*

编程竞赛

- **Google Code Jam Kickstart (谷歌 2017 全球校招赛)**
  全球 108^{th}, 中国 18^{th}, 前 5% **计分板链接** ***(id - WeiYong1024)***

工作经历

- **心识宇宙（大模型 Agent 平台创业公司，产品 MindOS）** 中国，杭州
  基础设施主管（4 人团队）、基建系统设计、*Kubernetes*、*Java*、*OpenAI*、*Azure*、腾讯云等 *2022 年 6 月 - 至今*
  - **技术职责——极至细节，追求最佳工程实践**
    * **AiInfra ——公司业务层的大模型能力底座**
      · **OpenAI 底座层：** 设计并构建中台服务管理 OpenAI 和 AzureOpenAI 资产并封装其服务。包括但不限于管理 AzureOpenAI 服务的区域、实例、部署以及模型版本，以及对 OpenAI 账号池化并在上层进行主动/被动探活、权重分配等可用性机制。
      · **多大模型封装：** 针对算法团队封装 OpenAI GPT、Google Gemini/Palm2 等第三方 SaaS 服务大模型的调用层服务，并做各业务线、功能点的成本监控。
      · **GPU 算力调度平台：** 调研设计针对训练任务的 GPU 算力调度平台作为技术储备，但由于公司将战略定在只做 AI-Agent 层而未实施。
    * **生产工具 &DevOps ——提供全套生产环境与工具**
      · **多套集群环境管理：** 设计并构建基于 Kubernetess 的多套环境集群架构；全部微服务容器化并上 Kubernetes；提供基于 Helm 的分布式 GitServer（Gitlab）, 于其上设计、构建并在全司推广基于 OwnershipReview 和 ReadbilityReview 的代码审核机制，从流程上严格保证全司代码库代码的工程可维护性；基于 Helm、Jenkins、Python、Shell 等工具构建分布式的包含打包、审批、发布、回归一体的 CICD 流水线（Jenkins），具备可伸缩的并行构建发布能力。
      · **生产工具体系构建：** 选型、设计并构建公司内部生产工具。负责公司内部组网、科学上网等网络环境管理；设计并统一分布式配置中心（Apollo、Nacos）、限流（Sentinel）等中间件；搭建运维工具（Rancher、JumpServer、Octant 等）；在离线数据工具体系（MySQL、Redis、Clickhouse、Superset、Grafana 等）；日志及应用性能监控工具（Loki、DataDog、腾讯云 CLS、Skywalking 等）。提供技术研发需要的全部工具。
      · **云资源管理：** 负责公司 IaaS、SaaS 资源生命周期管理，用户权限管理，精细化管控用云、用三方服务的成本。设计、制作及维护成本大盘。
    * **Data&BI ——管理数据资产、提供实验工具、数据驱动决策**
      · **在离线数据体系：** 设计、构建并维护基于 OLTP（MySQL）->CDC（外部供应商）->OLAP（Clickhouse）->DataVisualization（Superset+Grafana）的在离线镜像数据流用于离线数据开发，以及基于消息队列（RocketMQ）-> 基础服务（Springboot）的异步数据流用于用户行为数据分析等场景。作为产品、研发、运营同事所关心的各类指标的大盘、报表工具。
      · **A/BTest 工具体系：** 选型、设计、构建并维护包含实验配置/结果分析平台（GrowthBook）+ 前端埋点（GoogleAnalytics）+ 后端埋点（Springboot 实现，包含工程和算法）的一整套实验工具体系。支撑前端、页面、算法等开关和实验。
      · **产品隐私合规：** 基于 Vanta 提供的一系列隐私合规标准，包括但不限于数据脱敏、内部培训等，让公司产品 MindOS 达到 USDP 和 GDPR 标准。
    * **业务层研发——提供业务开发依赖的基础服务**

- **中台基础服务：**设计并构建中台基础微服务，为业务系统提供后端加密存储、用户隐私数据库、Web 爬虫、离线数据舒仓管理等通用基础服务，并以 RestfulAPI、二方库 +RPC 接口等形式输出。

- **支撑业务**
  - 主导移动端小程序万物总动员、Web 端产品 MindOS 的子系统和全链路压测、发布过程的重保；准备并制定 toB 公有云/专有云/私有化产品交付 SOP；服务架构随业务跨国/跨云地域迁移等。
  - 在全公司横向推进 blameless postmortem 文化，组织安全生产周会。

- **阿里云** 中国，杭州
  高级工程师、后端、Java、Springboot、阿里云 *2020 年 8 月 - 2022 年 5 月*
  - **流量调度中间件**: 在集团高可用团队负责阿里云内部流量调度中间件。基于阿里内部的 PandoraBoot（Springboot 加强版）提供集群上容器粒度秒级健康指标监控、3σ 离群点检测，依托阿里内部 RPC 框架 HSF 的权重表机制实现微服务的分钟级流量调度与恢复链路。防单点故障引起的集群雪崩，也可用于 JIT 预热。任职期间推广产品纳管了阿里集团全部核心应用，支撑 40w+ 容器，作为核心安全组件护航 2020 双十一 60wQPS 流量洪峰。
  - **私有云管理平台**: 时年 P8 团队目标建设政务场景下的专有云管理系统。基于 Java 框架 Springboot 开发，该系统分应用运维、资源运营、总集管理三个子系统。我负责其中应用运维中心子系统的后端开发。应用运维中心底层构建云资源列表、监控巡检、用量统计、自定义接口巡检、SQL 巡检等原子能力，并在上层此构建了慢 SQL 统计分析、业务大盘、体检报告等专家报表。该系统目前在全国超过 6 个局点商业化输出，重保护航 2021 某省会城市小学入学等项目、2022 春节核酸检测等项目。
  - **政务云管轻量化交付**: 时年新 P9 团队目标交付 7 个地方政务私有云，早期由开发同学使用 Rainbond 界面化地将来自原不同团队的微服务组合部署在客户 K8S 集群上，配置繁琐导致交付周期长。为解决该问题我基于 docker-compose 搭建单机版轻量化部署交付方案，基于通用的环境初始化脚本与交付步骤文档，仅通过维护不同客户现场的环境变量以中心化管控，纳管来自各个老团队的 20+ 个微服务。编辑 WBS 和 SOP，组织外包同学培训交付流程，单个经过培训的外包可以在 2 日内交付/升级一个现场。基于该方案，团队技术产品簇的单月商业化输出能力从个位数据点上升到两位数。

- **小马智行** 中国，北京
  工程师、Infra、C++、Python、Linux、Bazel、严苛的代码质量标准 *2019 年 2 月 - 2020 年 3 月*
  - **行车录音工具链**: 无人车路测需要一位跟车工程师通过外接键盘描述记录路测 issue，为去掉跟车工程师构建行车录音工具链，以麦克风和物理按钮作为硬件方案。首先在 Bazel 项目中引入基于 BSD 软件许可的输入设备接口库 evdev，并基于该库和 Linux 的 ALSA 音频驱动编写守护进程，负责硬件检测、监听来自外部按钮的信号和车载系统进程的管道信息、触发与停止麦克风录音，在行车过程中将 issue 信息存储为工控机上的音频文件。在数据处理阶段，首先在 GCP 上搭建语音转文字服务，并用 Python 脚本将音频文件的内容提取出来，然后使用 Google 的 Protobuf 工具将原信息序列化汇入 QA 数据流。
  - **车载语音系统**: 原车载语音模块使用 Google 的 Pico TTS 库在行车过程中将硬编码的语音内容文本实时播放，从而导致语言单一和 TTS 过程重复消耗计算资源的问题。新的车载语音系统使用基于音频文件的工作流程，旨在减少车载工控机的计算量并支持语音 I18N。为此 1. 研发环节用 Python 编写内部 CLI 工具 `car_sound_utils`(以下简称 CLI)，供内部工程师创建和变更现有语料库、管理语音包版本：首先在 CLI 中基于 AWS 上用 CloudFormation、Lambda 函数计算搭建的整套 TTS 服务封装音频创建命令，然后在 CLI 中添加上传、下载语音文件和上传语音包到内部 storage server 的命令，以及缓存机制加速上传下载。2. 车载系统环节，主进程初始化语音模块时使用 Linux 的 ALSA 驱动将音频流加载进内存，并在语音模块接到播放请求消息时播放。

实习经历

- **爱彼迎**: 2018 年夏，爱彼迎发力中国市场，作为大陆首批 8 个实习生，全栈开发了当年国内版房东年度回顾页。

- **旷视科技**: 2018 年初，旷视和 VIVO 合作期间，参与 X21 机型人脸识别模组的研发，做基于 InceptionV3（一种 CNN）移动端模型搜索优化。

行业经历与技术栈

- **系统经验**：大模型 Agent Platform、云计算、集群高可用、DevOps、自动驾驶等

- **技术栈**：Java、Python、Kubernetes、C++ 等