

윤종혁

서울특별시 은평구 구산동
+82 10 4922 9653
ageldama@gmail.com
github.com/ageldama

생년, 병역, 자격증

- 출생 1982년 10월
병역 육군병장 2년 만기, 국군수송사령부
자격증 정보처리기사

교육

- 2016 학사 (4년제), 컴퓨터과학, 대한민국 교육부, 독학학위제
2014 교육생 리더, Software Developer Training Program, 대한민국 정보통신산업진흥원
정보통신산업진흥원 전액 지원 및 기획으로 1개월간 실리콘밸리 San Jose State University
체류하며 연수하였습니다. 당시 처음 시작하는 프로그램에 참여하였고 교육생 리더로서
자원하여 봉사했습니다.

실리콘밸리의 구글, 애플, VMWare와 같은 테크기업과 스타트업들을 방문하고 강연
참석하며 실리콘밸리의 문화와 기술, 방법론에 대해서 많은 것을 느끼고 배울 수 있었던
값진 경험이었습니다.

구사언어

- 한국어 모국어
English Intermediate

Knowledge Area

- Programming Languages ANSI C, Modern C++, Go, TypeScript, ES6, Java, Delphi, Ruby, Python 2/3, Perl 5, Common Lisp, Clojure, Solidity
Backend Node.js, Express.js, Nest.js, Java Spring, JPA/Hibernate, Express.js, Python Flask, Django, FastAPI, Ruby on Rails
Database PostgreSQL, Oracle DBMS, MySQL/MariaDB, Apache Cassandra, MongoDB, Redis, SQLite
Frontend jQuery, React, htmx, Hotwire, Webpack, esbuild
Etc ERD & UML Diagrams, Git, Jenkins CI-CD, Agile & SCRUM 교육 다수 참여

경력 2024–2019

2023.11 – 백엔드개발자, 아키텍트, VETCHING PTE. LTD., 한국 서울

2024.11 동물병원을 위한 클라우드 EMR(전자의무기록시스템)서비스 기업에 백엔드개발자, 아키텍트역할으로 일했습니다.

기존 아키텍쳐, 인프라를 확장성 있게 만들고, 개발팀 전반의 업무프로세스를 확장하였으며, Windows 애플리케이션/COM 디버깅/개발, 외부API 및 계측장비 RS-232/USB/TCP-IP 연동, DICOM 의료영상포맷처리, 애플리케이션 설치 및 업데이터 재개발 등 넓은 범위의 기술적인 내용의 탐색, 개발, 지원을 하였습니다.

기업비전과 전망이 기여도에 맞지 않아 정리하였습니다.

VETCHING: 『 api-server 』

• GraphQL Subscriptions와 Redis Lua 스크립팅을 활용한 실시간 참여자 목록 기능 구현.

• Apollo GraphQL + Prisma ORM 기술부채를 원만히 단계적 개선할 방안을 탐색:

 docker-compose testsuite와 DB마이그레이션 정리하여 각 조합케이스별 memory footprint 검토.

 - Nest.js, TypeORM, Kysely 등의 조합을 비교/검토.

• ‘미라벨소프트’ PACS 연동 개발.

• node.js용 Oracle DBMS 드라이버를 OCI FFI를 사용하는 버전을 Thin 버전으로 대체하여 프로젝트 빌드 및 의존성 개선.

VETCHING: 『 socket-server 』

• (리거시) 연결유지가 불안정한 문제의 분석을 착수: Apollo GraphQL Subscriptions + 웹소켓 스펙 및 구현이 keepalive 처리를 잘 지원하지 못함을 확인하여, 대체할 방안의 탐색을 시작.

• (v2) 연결유지 불안정 및 성능부하가 큰 문제를 해소하기 위해, Go언어 WebSocket/ProtoBuf기반으로 새롭게 설계/개발.

 - Google Wire, Uber Zap Logger, GORM, kafka-go, Gorilla WebSocket, go-chi

 - Uber Mock, Testify, Docker Compose을 활용하여 단위테스트, 통합테스트, 시스템테스트 모두 작성하여 검증.

 - TypeScript 클라이언트 모듈도 설계/개발.

 - CTO 및 백엔드개발팀에 Go언어 및 개발전반 교육.

VETCHING: 『 인프라 』

- 시스템메트릭 수집을 위한 **Munin** 도입 및 주요 마이크로서비스 적용.
- **munin-anmeldung**: Munin에 자동적으로 등록될 수 있도록 munin-anmeldung (agent) 개발하여 적용.
- **locust, ansible + AWS ec2 CLI** 활용하여, 성능 테스팅 및 부하테스트 진행 (외부감사제출용)
- **docker-compose** 및 새로 작업한 db-migrations 적용, testable하도록 리팩토링, integration/system test suites 기반 작성.
- **OpenTelemetry Zipkin** 도입, 인프라 작업 및 api-server 등에 연동.

기존 Sentry연동을 대체하려 작업.

- **dmsfilter**: 기존 **AWS DMS** 방식을 대체하는, *PostgreSQL Logical Replication* 기능을 활용한 “dmsfilter”을 설계/개발/인프라 작업.

AWS DMS은 DB마이그레이션을 위한 서비스으로, 실시간 디비변경점을 수신하기 위한 도구로 사용하기엔 불편함이 많고, 다수의 멀티테넌트 스키마를 잘 지원하는데 한계에 다다르고 있었음.

Go, pgx, kafka-go, gRPC 사용.

- 새로운 스키마를 추가하더라도 기존 AWS DMS와 달리 서비스 다운타임도 없고, 매우 적은 ec2 인스턴스 CPU/메모리 footprint만으로 대체가 가능했음.

- **AWS ELB**을 통한 멀티테넌트 분리구조를 **haproxy**으로 개선하고, **haproxy-discovery** 설계와 개발, 인프라 작업.

기존 AWS ELB은 ELB Ruleset의 갯수 제약 등으로 테넌트별 분리가 한계에 다다랐었음. 또한, ”프리미엄” 테넌트들을 적용하여 쾌적한 인스턴스에 특정 고객병원을 할당하거나 할 수 있도록 지원. haproxy으로 대체하고, AWS API를 이용하여 자동으로 api-server 인스턴스를 감지하여 haproxy에 연결하도록 “haproxycfg” discovery을 설계/개발.

- ELB의 제약점을 해결하였고, 새로운 테넌트/인스턴스를 추가/삭제할 때에도 다운타임이 없도록 개선. 또한, 인프라 관리에도 같은 테넌트별 정보를 여러 군데에 분산하였던 것들을 통합관리할 수 있도록 하였음.

VETCHING: 『 victor 』

윈도애플리케이션, 계측장비, 결제, 영수증프린터 등 연동.

- 결제연동 DLL에서 발생하는 오류를 제대로 trap하고, debug용 메모리덤프를 남기도록 수정 및 윈도용 디버그덤프 분석방법을 교육.
- 자동업데이터 및 배포패키지를 NSIS 방식으로 재작성 및 NSIS교육. 기존 인스톨러/업데이터의 복잡성을 줄이고, 제대로 배포되지 않은 문제를 해소, NSIS을 통해 설치/업데이트 시 세밀한 제어가 가능해짐.
- シリ얼포트 통신 장비를 개발/디버깅하기 수월하도록 “com0com” 가상드라이버를 이용한シリ얼통신 sniffing 도구 개발 (Python, pyserial)
- SKB IMS 전화API 연동을 위한 “32비트 COM연동서버” 개발 및 .NET 클라이언트 개발.
 - 32비트 COM/DLL을 연결하기 위한 Python서버 프로세스, (pydantic, flask, pywin32)-와 .NET으로 작성된 Victor 애플리케이션에서 이를 호출하기 위한 클라이언트 라이브러리.
- “BC-30 VET” 장비 시뮬레이터 개발.
 - 통신 프로토콜 및 페이로드의 형태를 Victor 애플리케이션 개발자가 쉽게 이해할 수 있도록 개발. (Python, TCP Socket)
- “Skylar VB-1” 장비 시뮬레이터 개발.
 - (Python, PySerial)
- “Skylar PT-10V” 장비 시뮬레이터 개발.
 - (Python, TCP Socket)

VETCHING: 『 데이터컨버전 』

- 타사 델파이 기반 EMR에서 작성한 rich-text 형식인 rvf → html/css 변환기를 개발.

리거시 도구지만, 소스코드 유실 및 하위디렉토리 검색하지 못하는 등등의 문제 가 있어서 그냥 다시 개발.

- “테레사”-PACS의 DICOM 데이터를 “미라벨소프트”으로 이전하는 프로그램 작성.

(이전에 작성해놓은 DICOM → 이미지 변환프로그램 등 연결하여 작업) (Python, Requests)

- 타사(A) EMR 바이너리 동적분석하여 zip 백업파일의 암호 확인.

컨버전을 위한 데이터를 열기 위해서. ‘d7zip’ 컴포넌트를 사용해 zip 압축을 수행하는 것을 분석하여 착안, 해당 exporting API에 전달되는 문자열을 포착하는 동적분석. (x86dbg 사용)

- 타사(B) EMR 바이너리 정적분석하여 암호화된 DB컬럼의 복호화 방법을 확인.

암호화 알고리즘 관련 코드를 실행파일에서 찾아내어 분석+재구성하고, 이를 위한 암호화 Key/Salt 값을 찾아내어 재구성한 복호화/암호화 코드으로 테스트케이스 작성하여 확인. (IDR, Ghidra 사용)

- 타사(A) EMR/PACS의 비표준 DICOM 이미지의 인코딩을 분석하여, 이미지 변환도구 작성.

비표준의 이미지 압축코덱을 사용함을 정적분석을 통해 확인. (IDR, Ghidra 사용)

- FFI를 통해 C++ / x86 “thiscall”-방식 ABI DLL을 호출하여 raw bitmap 분리. (Python, VC++, PyDicom)

VETCHING: 『 back-office 』

- (리거시) Infinit PACS 연동 기능, JWT/Passport 이용한 백오피스 사용자관리 (기존에는 ID/PW 하드코딩), AWS Beanstalk 인스턴스 관리 기능, AWS ELBv2 관리 기능, AWS DMS 관리 기능 등을 추가.

- (리거시) 기능추가/유지보수 방식이 GraphQL/React.js 등을 통한 방식이어서 오버엔지니어링이 커진 것을 대체할 방안들을 검토.

nest.js/typeorm, reflex/python, pysimplegui/python, streamlit/python, nicegui/python 등 검토.

- nest.js, kysely + htmx 방식으로 진행하기로 계획.

- (v2) nest.js + htmx으로 백오피스 재작성.

기존 GQL/React.js 방식은 백엔드와 프론트엔드 서버가 별도로 필요하고, bundler 실행 및 gql 빌드 등을 위해 한 가지 API를 추가하려면 2 3번은 서버를 재시작해야 했었으므로 백오피스 작업이 불편하던 것을 효율화.

VETCHING: 『 etc 』

- 데이터베이스 스키마 마이그레이션 도구를 설계/개발.

- 기존에는 *Prisma ORM*의 마이그레이션 기능을 활용했지만, 실제 SQL DDL을 표현하는데 제약이 많고, 형상관리가 제대로 되기 어려운 형태여서 새로 설계.

- node.js, nunjucks template engine 등을 활용하여, multi-tenency에 잘 대응될 수 있도록 설계/개발.

- socket-v2, booking-server, api-server 등의 통합테스팅, 시스템테스팅을 제대로 하기 위해 디비 스키마를 쉽게 만들고 공유할 필요가 있으므로 또한 필요했음.

- batch-server 설계/개발.

- PDF to 이미지 변환 등의 비동기작업을 위한 신규 마이크로서비스를 설계/개발 및 인프라 작업. (Python, FastAPI, Celery)

- ‘네이버예약’ 및 LG U+ AIICC 예약 연동을 위한 booking서버 기반을 개발 및 AWS 인프라 작업.

- 네이버예약 API 클라이언트 및 대규모 예약내용을 저장/조회용 DB 개발.

- gRPC으로 기존 api-server와 연동가능하도록 개발.

- 네이버예약측의 콜백 노출부분 등 개발.

(Nest.js, Kysely, Mongoose, gRPC)

- plusvet-booster 설계/개발.

(Visual C++, Windows API)

- Electron 앱인 PlusVet이 느리고 매우 불안정하여, OS 프로세스 Priority, CPU Affinity 등을 우선적으로 조정해주는 추가 애플리케이션 개발. (주로 Delphi으로 개발된 타사 EMR들도 같은 조정기능을 포함하고 있음)

- 실제 불안정성의 원인은 React.js, 상태관리, Electron SPA의 조합에서 오류가 있었었음. (예: *unhandled-rejection* 등)

2020.04 – **창업자**, 고래모비, 한국 서울

2022.07 경매식 농수산물 도매거래소.

1인 개인기업으로 기획, 설계, 프론트엔드, 백엔드 및 운영을 하였습니다.

Go언어와 자체적으로 만든 바닐라JS으로 백엔드, 프론트엔드를 작성하였습니다.

농산물 생산자가 직접 판매하는데에 도움이 되고 싶어 기획하여 론칭하였습니다.

최초 버전은 Web3을 이용하여 입출금을 만들었고, 이후 PG연동으로 원화 입출금으로 변경하려 하였습니다.

백엔드는 모두 Go으로 작성하였고, Node.js/Jest으로 시스템테스트suites 작성하여 검증하였습니다.

스타트업의 농수산물 분야 스타트업에서 성공적으로 잘 진행하고 있고, 더는 진행하지 않아도 배우고 얻은 것이 많다고 판단하였습니다.

- Go, Ethereum, Web3, Node.js, PostgreSQL

- Vanilla JS, Webpack, Bootstrap

2021.01 – **SW Engineer**, 플랫폼팀, 코빗, 한국 서울

2021.04 헤드헌터를 통해 제안.

진행하고 있던 개인사업을 계속하는게 맞다는 판단이 들어, 헤드헌터에게 수익이 갈 수 있도록 프로베이션 기간만 마치고 퇴사하였습니다.

프로베이션 기간 동안, 프로토타이핑 단계에 있던 JWT인증 서버를 내부 API게이트웨이에 연결하여 실적용이 가능하도록 다시 Spring Data R2DBC, Spring WebFlux으로 비동기입출력 고성능 서버로 재개발하였습니다.

Spring Test, JUnit, Python 3, PyTest을 이용한 기능테스팅을 통하여 기존spec과 동일한지를 검증하였고, 동일한 방식으로 성능측정 테스팅 등도 작성하여 비동기식 입출력으로 성능이 향상되어 내부 마이크로서비스들간의 API호출이 원활히 인증될 수 있음을 확인하였습니다.

2020.01 – **Lead SW Engineer, 개발팀, promedius.ai, 한국 서울**

2020.03 초기 단계의 의료영상 딥러닝 스타트업에 지인의 소개로 유일한 개발자로서 합류하였습니다.

딥러닝/영상처리 연구에 집중되어 있었었고, 제품개발방향이 아직은 모호한 단계에 합류하여 제가 도움이 되기 어렵다는 판단이 들어, 정리하고 제 사업을 시작였습니다.

제 사업을 착수하기 이전에 코로나바이러스 유행이 시작될 즈음이어서, 대표의 아이디어로, 원격진료를 위한 병원, 약국, 고객을 위한 서비스를 기획-설계-개발을 주도하며, 프론트엔드, 관리자 시스템, API서버 모두 단독 개발하여 론칭하였습니다.

또한, 의료 영상인식 서비스의 의료기기 심사를 위한 AI모델과 연동하며 의료 영상정보 DICOM 및 의료장비 간 네트워크 프로토콜인 SCP/SCU으로 통신하는 시스템을 Java / Apache Camel으로 단독으로 설계-개발하였습니다.

- Java 8+, Apache Camel, dcm4che, Spring WebMVC, Spring Data/JPA, PostgreSQL, JUnit 5 Jupiter, Mockito
- C++, DCMTK, CMake
- Kotlin, Vue.js, Vuex, Vue-Router, Ruby on Rails, PostgreSQL, Bootstrap

경력 2019–2018

2018–2019 **Lead Architect, Architecture Team, Streami.co, 한국 서울**

독일에서 귀국하여, 흥미로운 한국스타트업에서 경험을 하고 싶어 입사하였습니다.

시니어 백엔드 엔지니어로 입사하여, 이후 아키텍쳐팀의 팀장이 되어, 자사 암호화폐 거래소 GOPAX의 성능 및 안정성 개선을 위한 재설계와 리팩토링 업무를 주로 담당하였습니다.

기존에 Node.js으로 작성된 거래 API 서버의 성능 및 안정성 개선과 Java으로 작성된 주문 매칭 서버 유지보수 및 개선도 수행하였습니다.

거래 API 서버를 Go언어으로 재작성하여 동시성 등 개선 시도하였습니다. 또한 Node.js 코드베이스를 테스트 가능한 형태로 만들기 위한 테스팅 지원 프레임워크를 설계-개발하여 적용하였습니다. Amazon AWS 환경을 경험을 할 수 있었습니다.

‘특금법’과 같은 한국내 규제와 맞물려, 고팍스에는 은행실명계좌 발급이 어려울 것으로 예상되어 퇴사하였습니다. (실제로 고팍스는 이후에 3, 4년간 실명계좌를 받지 못하였습니다)

2017–2018 **백엔드SW엔지니어, Architecture Team, Smava GmbH, 독일 베를린**

베를린의 Loan/Credit Comparison 스타트업. eBay의 자동차 판매서비스인 mobile.de와 연계하여 mobile.de에서 차를 구매하려는 사용자에게 가능한 대출상품을 제안하는 서비스를 프로젝트를 주로 담당하였습니다.

Spring Cloud 프레임워크를 활용한 Smava의 서비스 Configuration, Discovery, Security 인프라에 맞춘 마이크로서비스들으로 구성해 만들었습니다. 마이크로서비스 아키텍처와 Spring Cloud 프레임워크, Netflix Eureka/Hystrix와 같은 MSA를 위한 프레임워크에 많은 경험을 했습니다.

또한 사용자의 희망금액-이자율 등의 입력 및 신용상태(SCHUFA)에 따른 상품 및 한도를 예측하기 위해 Smava의 다른 Prediction Team등과 마이크로서비스 간 연동과 계속적인 e2e테스팅이 필요하여 초기 개발단계에서 간단히 Perl, LWP 모듈을 활용하여 계속적인 테스트를 진행할 수 있도록 제안하여 만들었습니다.

MSA구조와 코드베이스의 단위테스팅, 그리고 나아가서 마이크로서비스 간의 협업, 연동 테스팅에서 많은 경험을 했고, 이후 서비스 초기 릴리즈 이후 테스팅엔지니어가 더욱 전문적인 Test Suite을 작성하도록 기준에 만들었던 테스트 시나리오와 도구들을 인수인계해주었습니다.

소득세율, 송금비 등을 고려하여 한국 스타트업에서 일하는 것이 더 수익성이 좋을 것으로 판단하여 프로비전과 담당하던 프로젝트를 모두 성공적으로 완수하고 귀국하였습니다.

경력 2017-2016

2016-2017 백엔드SW엔지니어, Discount Service Team, Zalando SE, 독일 도르트문트

유럽 전체에서 가장 성공적인 패션IT기업인 Zalando SE에 직접 지원하여 취업하였습니다. 1개월간 베를린에서 On boarding 교육 수료 후 원하는 팀을 지원하여 도르트문트의 Discount Service Team에서 일하였습니다. 선택한 상품의 할인, 쿠폰, 기프트카드 등의 준실시간 처리를 담당하는 팀에서, 기존 리거시 엔진을 다시 설계하고 만드는 작업을 했습니다.

기존 리거시 시스템은 할인, 사은품 등의 규칙이 DB 스키마에 내장된 형태로 규칙을 설계하는데 유연성이 부족하고 성능과 저장공간 확장성이 매우 떨어지고 소프트웨어엔지니어가 아닌 상품진열 및 마케팅 인력이 데이터를 직접 입력하고 확인하기도 어려움이 있었습니다.

이런 문제를 해결하기 위해 재귀적 트리-리스트 표현법을 이용한 작은 프로그래밍언어를 만들고, 이 프로그래밍언어로 할인규칙을 표현할 수 있도록 설계했습니다. 또, 문법구조가 일관적인 트리-리스트 표현법이어서 Block Programming 언어와 같이 쉽게 GUI으로 작성 및 수정이 용이하도록 설계했습니다. 이 언어와 구현체를 직접 고안, 설계, Proof-of-Concept 개발 및 JVM 타깃의 컴파일러를 만들었습니다. PoC 프로토타입은 오픈소스로 <https://github.com/ageldama/thompson-py-poc> 공개하였습니다.

많은 새로운 과감한 새로운 시도를 해보는데 저 혼자서는 간과할 현실적인 문제와 조언들을 같은 팀 동료들로부터 듣고 설득하며 더 나은 결과를 이끌어냈습니다. 어쩌면 비판이 아닌 비난과 비방만 하고 싶을지도 모르는 새로운 시도에 현실적이고 건설적인 논의를 함께 할 수 있는 동료들과 함께 일할 수 있어서 기쁘고 건설적으로 진행할 수 있어서 감사한 경험이었습니다.

Zalando이 유럽내 수많은 국가에 서비스를 하고 있고 가장 유명한 온라인 패션브랜드이고, 지역마다 테크오피스가 있고 회사내부에 수많은 인프라, 공통 마이크로서비스팀들이 있어 MSA을 극한까지 활용하는 기업입니다. Nakadi Message Stream와 이에 기반한 MSA연동, Coordination, Discovery 등등과 자체 Docker에 기반한 테스팅과 배포 솔루션 등등 다양한 MSA 아키텍쳐는 물론 팀의 구성, 문화, 방법론에 대한 깊은 경험을 할 수 있었고, 문화와 방법론과 같은 교육과 이벤트가 사내에 계속적으로 있어 많은걸 경험하고 배울 수 있었던 유익한 경험이었습니다.

종신직으로, 영주권 취득 후 계속 근무해도 좋은 상황이었지만, 독일내 소득세와 한국으로 송금비를 고려하여, 프로베이션과 맡고 있던 프로젝트를 모두 성공적으로 마친 다음, 다른 직장을 더 찾아 보기로 하였습니다.

경력 2015–2014

2015–2016 백엔드SW엔지니어, LINE Plus, 한국 분당

피키캐스트 재직 중 이직제안이 있어 입사하였습니다.

세계적으로 수많은 사용자와 트래픽을 가진 메신저 서비스인 라인의 한국 조직에서 일했습니다.

입사 후 인도네시아의 많은 인구가 라인을 쓸 수 있도록 유도하고 싶어서 인도네시아향으로 출시한 LINE Alumni 서비스에 합류하였습니다. 이미 초기 릴리즈를 한 서비스였는데, 이 서비스를 위한 어드민 개발과 친구추천 초기 버전을 MySQL NDB Cluster, Redis, Python, Spring Integration, Apache Hive으로 직접 설계하여 개발하였습니다.

다음으로 라인 앱버, 타임라인 서비스 유지보수를 담당하며 Kafka, Redis, Cassandra, MySQL, Java, Spring Frameworks을 계속해서 사용하고, JVM 메모리 누수 문제나 튜닝, 분석을 작업하여 매우 성공적인 결과를 얻었습니다.

이후 내부 데이터 마이그레이션 마이크로서비스를 담당하여 설계하고 직접 개발하였습니다. 수많은 사용자의 프로필 데이터를 무리 없이 안정적으로 Hive, Cassandra, Kafka을 활용하여 마이그레이션을 On demand으로 처리할 수 있도록 개발했습니다.

기술적으로나 기업문화 모든면에서 라인플러스 외부에서 경험하기 어려웠을 당시 매우 혁신적인 경험을 많이 할 수 있었습니다. 깔끔하고 좋은 technical writing 문화와 직원들이 참여하고 공유하는 지식베이스, 팀별로 자치적으로 마이크로서비스를 만들고 관리할 수 있는 고속 대용량 메시징버스와 서비스레지스트리 등등 대규모, 대용량 서비스와 조직을 위한 것들이 어떤지 배울 수 있어서 값진 경험이었습니다.

프로젝트를 마치고 몇 개월을 더 근무하며 지냈습니다. 이후 해외의 대학원 진학 혹은 해외의 테크기업을 경험하고 싶어 퇴사하였습니다.

경력 2015–2012

2014–2015 **백엔드SW엔지니어, 피카캐스트, 한국 서울**

처음 페이스북 페이지에서 자체 서비스로 독립한 피카캐스트에 합류하였습니다. 백엔드 서비스는 있었지만 모노리티 서버에 기반하고 있었고 많은 방문자를 감당하지 못하고 있었으며, 백엔드 엔지니어는 저 혼자뿐인 상황이었습니다.

계속되는 서비스 부하를 감당하지 못하는 서버의 구조를 부하분산을 할 수 있는 구조로 재설계, 리팩토링을 혼자 담당하였고, 서비스 모니터링, 서버 인스턴스의 생성 및 설치, 유지보수를 흘로 하였습니다.

API서버의 분산이 가능하도록 분리-리팩토링, 데이터베이스를 클러스터화하고, CDN 적용 및 Varnish/HAProxy/Redis을 이용하여 부하분산 및 캐싱을 개발하였습니다.

또한 각 서버들의 건강상태를 확인하고 알림을 받기 위해 Monit, Munin등의 모니터링 시스템을 설정하였습니다.

개발조직 내부에 인프라 개선에도 노력하였습니다. GitLab을 도입하여 소스코드를 관리하고, Maven 3와 Jenkins을 도입하여 빌드 시스템을 개선하고, 테스팅 가능한 환경, 변경에 대해서 자동적으로 검증이 가능하도록 했습니다.

컨텐츠 기획을 위해 사용자의 행위트래킹 시스템의 백엔드 부분을 설계하고 개발하였고, 이후 수집된 데이터를 분석하는 작업에도 참여하였습니다. (MongoDB, PostgreSQL, Hive, Python)

사용자에게 알림을 보내는 Notification 서비스도 설계 개발하여 안드로이드 GCM, iOS APNS, 바이두 Push등과 연동하였습니다.

급격히 성장하는 서비스에서 어떻게 하면 효과적인 방법으로 안정적이고 빠른 서비스가 가능한지를 많이 터득할 수 있었던 값진 경험이었습니다.

2012–2014 **프리랜서, 한국 서울**

프리랜서로 일하며 정부과제로 정부부처별 Facebook 페이지의 빅데이터 분석을 하여 보고서를 제출한 경험이 있습니다.

또, 약 1년간 프리랜서 유지보수 담당자로 팬텍CNI에서 근무하며 내부 ERP, 라츠온 라인쇼핑몰 유지보수, 행사 및 이벤트 개발을 담당했습니다. (Spring WebMVC, MyBatis, Oracle 10g RAC, jQuery)

팬텍CNI에서 일하며 CVS에서 Subversion을 도입하고, Hudson CI, Nexus저장소를 도입하여 팀 내부의 인프라 개선에도 기여했습니다.

그리고 유실된 Visual Basic 6으로 작성된 ActiveX을 수정하기 위해 VB6 P-Code을 디스어셈블하여 분석하고 다시 작성하여 교체하는 등 흥미로운 경험이었습니다.

경력 2011–2007

2007–2011 테크니컬리더, ExInno (ValueInfra), 한국 서울

병역을 마치고, 병역이전에 함께 일하던 동료가 창업한 회사에, 창립멤버로 입사했습니다.

처음 귀금속 도매상과 공장을 위한 ERP솔루션 서비스를 개발하고 유지보수하였습니다. (Oracle, Java, Cobra Framework)

유지보수하던 귀금속ERP 솔루션이 지식경제부 주관, 삼성SDS코소시엄의 시범사업에 포함하여 1차 시범사업과 2차 사업에 모두 참여하였습니다.

회사 자체 서비스, 주식가치투자 분석서비스인 quantv.com의 서버측 개발을 부분 담당하여, Django, Python 2, jQuery을 이용하였습니다.

SI으로 국민은행, 하나은행의 자금세탁방지(Anti Money Laundering, AML) 시스템 구축에 참여하였고, SAS Korea의 파트너사로 일하였습니다. SI으로 일하였지만, AML 시스템을 위한 시각화 솔루션 ‘Fund Tracker’을 직접 개발하여 국민은행과 하나은행 등 다수의 은행에 납품하였습니다. (Oracle DBMS, Java, Adobe Flex)

포스데이터(現 포스코ICT)의 SmartECM 연구개발 프로젝트에 참여하였고, 이때 C++, Lua, Windows API/COM을 활용한 ECM Agent ('Agent8') 제품을 직접 설계하고 개발하였습니다. SmartECM 프로젝트는 성공적으로 완수하여 다음해 포스코그룹의 핵심 3대 키워드로 전사 및 협력사들에 확대 적용되었습니다.

그 이후 ECM Agent 납품과 함께, ExInno이 삼성반도체, 대한민국 공군, CJ E&M 등에 ECM 솔루션 구축에 계속적으로 참여할 수 있었습니다.

직접 기획, 설계, 개발한 ECM Agent 상품 등으로 인해, 처음 총3명으로 시작한 기업에서 몇십명 단위의 기업으로 성장시켜 본 흥미로운 경험이었습니다.

오픈소스, 공개활동

moonshot.el, *Founder*, 2020년

<https://github.com/ageldama/moonshot>

EmacsLisp, 현재 프로젝트/하위디렉토리에서 Levenshtein 문자열 매칭하여 현재 편집중인 파일과 가장 유사한 실행파일을 추천하여 실행, 디버깅.

그외에도 실행할 커맨드 패턴들을 등록하여 편리하게 실행기능 지원.

Cl-State-Machine, *Founder*, 2020년

Simple state-machine DSL library for CommonLisp.

<https://github.com/ageldama/cl-state-machine>

FiveAM, Graphviz등을 활용.

otp-sh, *Founder*, 2021년

<https://github.com/ageldama/otp-sh>

Node.js으로 간단히 T-OTP Secret이나 Google Authenticator URI을 인식하여 개발/테스트시에 활용하도록 개발.

Pazuzu, Docker Image Builder, *Contribute*, 2016년

Zalando SE의 내부 Incubating 프로젝트인 Pazuzu에 Configurations and Command-line Options Support 부분 코드를 작성하여 기여하였습니다.

Go Language: <https://github.com/pazuzu-io/pazuzu/issues/105>

Thompson-Py-PoC, *Founder*, 2017년

Zalando SE의 Discount Team에서 진행한 프로젝트의 일부 PoC을 회사와 협의하여 오픈소스으로 공개하였습니다. (실 서비스는 Java/JVM으로 동작하는 컴파일러.)

<https://github.com/ageldama/thompson-py-poc>

Proof-of-Concept implementation of Tiny Lisp-like language interpreter in Python이며, Supports closures and higher order functions을 지원하도록 설계-구현하였습니다.

Python 3.5+, PyTest, MyPy 정적타입체커 활용하여 검증하였습니다.

Cl-Tmcl, *Founder*, 2014년

<https://github.com/ageldama/cl-tmcl>

Simple String Templating engine for CommonLisp - like <https://ruby-doc.org/stdlib-2.6.5/libdoc/erb/rdoc/ERB.html>

SDLHan, *Founder*, 2002년
<https://sourceforge.net/projects/sdlhan/>

ANSI C, LibSDL 1.2용.

비트맵 폰트 사용, 조합형 한국어 문자열 출력 라이브러리.

FreeType2와 같은 트루타입 렌더링보다 가볍고 빠르게 사용하려고 개발.

OpenAL Binding for Ruby, *Founder*, 2009년
<https://rubygems.org/gems/ruby-openal/versions/0.2>

루비 언어에서 OpenAL 사운드 라이브러리를 사용하기 위해 바인딩 개발.

EUC-KR Encoding/Decoding 지원 모듈, *Contribute*, 2009년
<https://github.com/factor/factor/tree/master/basis/io/encodings/euc-kr>

EUC-KR Korean encoding support for Factor Programming Language. (<https://factorcode.org/>)