**<프로젝트 수행 이력>**

* SKB G2 프로젝트
* 프로젝트 유형 : 사내프로젝트
* 소속회사명 : SK 주식회사 C&C
* 프로젝트 수행 기간 : 2017. 02 ~2018.03
* 주사용기술 : PaaS, MSA, Spring, Linux
* 담당업무 :

\* 프로젝트상세설명

-- SKB TV 메뉴 / 인증 / 즐겨찾기 / 시청내역 / 구매내역 MSA 전환

-- SKB Private Cloud 환경의 PaaS Infra 구축

-- SKB DevOps 적용을 통한 통합운영

\* 담당업무

-- 통합 Arch 파트에서의 Technical Arch

-- 타 파트에서의 이슈 트러블 슈팅 및 가이드라인 제시

-- Backing Service 구축 및 가이드

\* 담당상세업무

<Infra 개발 환경 구축>

- Backing Service 설치 및 관리

- IaaS 관리

- NAT 관리

- DNS 구축 및 관리

- VPN 신청 및 관리

- LDAP 구축 및 관리

- PaaS 시스템 연동

<아키텍처>

- 시스템 구성 설계

- PaaS 연동 개발 가이드

- 로그 표준 가이드

- Backing Service 연동 가이드

- 단위 테스트 가이드

- 통합 성능 테스트

<DNS 구축 및 관리><LDAP 구축 및 관리>

- DEV/STG/PRD DNS 가 하나의 Master 로 관리

- vLB/GSLB 를 통해 Domain 분기처리

- 외부 DNS / 내부 DNS 따로 관리

- HA 구성을 통해 서버 8중화로 관리

- 환경 : Ubuntu > CentOS 로 변경관리

- 시스템 통합 로그인 체계 구축

- openLDAP 을 통해 도메인 (환경/서비스) 구분값 관리

- DevOps (JIRA/Confluence/GIT…) 을 통한 PW 수정 관리

- DEV/STG/PRD LDAP 12중화로 관리

- 환경 : Ubuntu > CentOS 로 변경관리

<Backing Servie 구축 [REDIS]>

- DEV/STG/PRD 전부 다른 구성으로 진행 ( 자원 )

- PRD 국사간 동기화를 통해 관리

- 12중화로 서버의 부하시 IMDB단점을 최소화

- 서버 구축가이드를 통해 기본 IMAGE 제작

- REDIS Fail-over 테스트를 통해 서버스펙산정

- REDIS-STAT 페이지를 통해 모니터링 진행

<Backing Servie 구축 [Obejct Stroage]>

- Openstack SWIFT 사용

- KEYSTONE 을 통해 MSA 별 파트 계정 인증 시스템 활용

- VLB  HAProxy  NODE 구성을 통한 관리 체계 도입

- STORAGE 당 동기화는 진행하지 않게 관리

- API G/W 세션 관리를 Object Storage 에서 관리

- Fail-over 테스트를 통해 서버스펙 산정

<LDAP ADMIN PAGE>

- 개발개요

: LDAP 서버를 통하여 앱과 유저 그룹을

등록/관리할 수 있는 웹 애플리케이션 개발

- 개발언어 : NodeJS, HTML5, BootStrap, Javascript, LDAP

- 개발내용

 로그인 정보 관리를 위해 ADMIN 계정으로만 로그인 가능

 선택한 APPLICATION 그룹으로 USER 등록

 USER 기본 정보 수정 기능

 USER 패스워드 초기화

 USER 등록 정보 속성 관리

 APP 및 등록사용자 리스트 조회

 USER 일괄 IMPORT / EXPORT 기능

 도움말 페이지 제공

**<프로젝트 수행 이력>**

* 11번가 SNAPSHOT
* 프로젝트 유형 : 사내프로젝트
* 소속회사명 : SK 주식회사 C&C
* 프로젝트 수행 기간 : 2018. 02 ~2019.03
* 주사용기술 : SaaS, Spring, API G/W, Redis, ELK, HTML5, javascript, cssm Python, HAProxy, Object Storage
* 담당업무 :

\* 프로젝트상세설명

-- 자유 형식으로 업로드한 이미지 및 html 페이지를 규격에 맞게 가상 브라우저에서 html로 load하여 스크린샷을 저장하는 일련의 과정을 ‘snapshot’ 이라 통칭

-- 11번가의 legacy 환경에서 운영되고 있던 snapshot을 SK C&C의 cloudZ로 이전, SaaS의 형태로 제공

\* 담당업무

-- API G/W 개발자

-- 모니터링 시스템 개발자

\* 프로젝트 개요

-- SNAPSHOT을 통해 빠른 속도로 로드하며, 일정한 크기 이미지를 보기 위해 IMAGE RESIZE / ENCODING 처리하여 고객에게 노출시킴

\* 담당상세업무

<Infra 설계>

- G/W

KeepAlive : Active – Standby 구성을 통해 Main VM 활용도 높임

HAProxy : G/W health-check를 통해 TR 분배

JAVA (Spring) : 인증/redis 적재

HAProxy : REDIS health-check를 통해 TR 분배

REDIS : Master/Slave 구성

- Snapshot

python : Snapshot 처리후 ICOS 적재, REDIS 응답 적재, VM 1개당 프로세스 여러 개 수행

- 모니터링

ELK 스택 / NGINX / Spring

REDIS Sentinal : REDIS M-S 구성을 위해 함께 둠.

<G/W 개발>

암호화 KEY I/F 개발

 SaaS 요청 시, KEY TOKEN 방식을 통해 I/F

생성 요청 I/F

 생성요청 시, 암복호화/유효성검증 이후 Redis 에 요청값 적재

조회 요청 I/F

 조회요청 시, 암복호화/유효성검증 이후 Redis 에 응답값 조회

HEALTH-CHECK API

 Health-check 을 통해 호출 여부 판단 로직

REDIS Queue Count API

 Redis Queue 에 대한 정보 API 개발

<모니터링 개발>

Object Stroage 사용량 화면 개발

 IBM API 시, Class A/B/C 등의 HTTP 콜에 따라 사용량 계산

KIBANA 화면 조정

 각 이슈정리를 위해 KIBANA 화면 개선

<기타 운영개선>

오류코드 추가 개선

 네트워크의 에러로 이미지 로딩시, 엑박이 도래될 가능성을 예외처리 되도록 변경

REDIS Queue NAME 개선

 기존 스태틱하게 관리한 Queue name 유연하게 DB로 관리

WEBP 개선

 Cloud 처리시간을 단축 위해 jpg 이미지를 , google image format인 webP 사용하도록 변경

**<프로젝트 수행 이력>**

* OK Cashbag 시스템 운영 프로젝트
* 프로젝트 유형 : 사내프로젝트
* 소속회사명 : SK 주식회사 C&C
* 프로젝트 수행 기간 : 2018. 02 ~2020.02
* 주사용기술 : ProC, Oracle, JEUS, AIX, JAVA(1.4/1.8)
* 담당업무 :

\* 프로젝트상세설명

-- 오픈형 마일리지 시스템 OK Cashbag 운영 및 유지/보수

\* 담당업무

-- 정 가맹 / 부 회원 업무 담당자

\* 규모

-- 약 3,000 만명 회원

-- 최대 TPS 1,1000 TR

\* 담당상세업무

<FDS 시스템 개발>

- 개발 언어 : Spring (JAVA 1.8) , JEUS (JAVA 1.4)

- 개발 내용

가맹점등록시, FDS정책을 활용할 수 있도록

기존 SKP FDS 시스템과의 연동

- 이슈 상황

OK Cashbag 의 기존 시스템의 JAVA 버전이 낮은 이유로 외부와의 JSON I/F 시 많은 예외처리 필요하였음.

따라서 해당 이슈해결하기 위해 GW형식의 Spring 을 통해 I/F 함.

- 해결 방안

JAVA 1.4 에서의 I/F 하여야 될 것들에 대해 JAVA 1.8 FDS G/W 를 통해 I/F 되도록 변경

<운영 개발/개선 건>

- 개발 언어 : Pro\*C ( Proframe – Tmaxsoft )

- 개발 내용

SKT 가맹점 등록배치 개발 ( 2019.01 ~ 2019.03 )

가맹점 계약등록 1년제어 ( 2019.03 ~ 2019.05)

무실적 가맹점 계약 종료 처리 ( 2019.05 ~ 2019.06 )

제휴사 FDS 시스템 연동 ( 2019.07 ~ 2019.08 )

유종별 고객 등급관리 개발 ( 2019.08 ~ 2019.09)

이메일 전자결재 변경 ( 2019.10 ~ 2019.11 )

운영성 모니터링 메일 정리 ( 2019.11 ~ 2019.12 )

<운영 체계화>

- 운영 업무 프로세스 내용 정리

10년이상 동안 운영되는 시스템이나, 구전으로 운영되는 인수인계 내용 정리 및 체계화

<G/W 개발>

암호화 KEY I/F 개발

 SaaS 요청 시, KEY TOKEN 방식을 통해 I/F

생성 요청 I/F

 생성요청 시, 암복호화/유효성검증 이후 Redis 에 요청값 적재

조회 요청 I/F

 조회요청 시, 암복호화/유효성검증 이후 Redis 에 응답값 조회

HEALTH-CHECK API

 Health-check 을 통해 호출 여부 판단 로직

REDIS Queue Count API

 Redis Queue 에 대한 정보 API 개발

<모니터링 개발>

Object Stroage 사용량 화면 개발

 IBM API 시, Class A/B/C 등의 HTTP 콜에 따라 사용량 계산

KIBANA 화면 조정

 각 이슈정리를 위해 KIBANA 화면 개선

<기타 운영개선>

오류코드 추가 개선

 네트워크의 에러로 이미지 로딩시, 엑박이 도래될 가능성을 예외처리 되도록 변경

REDIS Queue NAME 개선

 기존 스태틱하게 관리한 Queue name 유연하게 DB로 관리

WEBP 개선

 Cloud 처리시간을 단축 위해 jpg 이미지를 , google image format인 webP 사용하도록 변경

**<자기소개>**

[Effort]

휴학을 하여 웹 프로그래밍에 관한 것을 강의와 책을 통해 학습하였습니다. 하지만 강의와 책에 내용을 똑같이 입력하였는데 제가 한 것은 작동을 하지 않았습니다. 그럴 때 마다 학원을 갔으면 조금 더 나았을까? 라는 유혹이 다가왔었지만 혼자 해결하려는 방법을 터득하려 노력하였습니다.

가장 믿을만한 것은 구글링을 하는 것이었으며 한국어 검색이 아닌 관련 에러를 영어 검색을 하는 방법을 시도하였습니다. 많은 사이트 중 적어도 하나는 저와 같은 에러를 가지고 있었고 해결을 할 수가 있었습니다. 하지만 내용이 조금씩 어려워지는 부분을 대비하여 저만의 책을 제작하여 에러에 대한 경험들을 함께 작성했습니다. 혼자 이런저런 에러를 겪다보니 에러에 대해 익숙해졌습니다. 독학의 힘은 무서울 만큼 저에게 이해도를 높여나갔다고 자부합니다. 쉽게 안 지식은 쉽게 사라진다는 믿음을 가지고 기초에 대해 꾸준한 노력하고 있습니다.

이후, 분석 Tool 제작하여 IEEE 에서 발표를 하게 되었고, 현재 재직중인 회사로 취업하게 되었습니다.

[DevelopMent]

게임을 만드는 학과를 다녔던 덕에 그래픽, 사운드, 미디어, 프로그래밍 등에 대한 다양한 것에 대해서 배울 수 있게 되었습니다. 저에게 완성의 가장 큰 성취감을 주었던 것은 개발이었습니다. 수업 이외의 Google App Engine 을 Cloud 개념이 나온지도 얼마 안된 시점에서 사용을 하고 MS 의 BOT Framework 로 Chatbot 프로그램을 만들기도 하였습니다. 꾸준히 새로운 기술에 대해서 배워야하는 직업이지만, 다르게 생각하면 꾸준히 다양한 접근으로 풀어나갈 수 있는 길들이 열려있는 것 같습니다. SaaS 시스템을 만들어야 했던 적이 있는데 그때의 핵심은 많은 비용이 문제였었습니다. 실수와 실패는 많았었지만 조금 더 비용에 대해서 SAVE 할 수 있는 구성을 고민을 하였습니다.

[Harmory]

나무를 바라보는 단위개발자로 시작한 것이 아닌, 숲을 보는 아키텍처부터 회사 업무를 담당하였습니다. 그러다 보니 많은 파트들과 협업을 통해 많은 논쟁이 발생하였지만 그 토론을 통해 배운 것이 있었습니다. 다양한 파트들이 자신의 주장은 발휘하지만 결국 대화를 통해 협의점을 찾는 과정들이 아름다웠습니다. 거친 돌들을 한 통에 넣고 계속 굴리면 다음 날 매끈한 돌이 된다는 말이 있었는데, 계속 무언가를 다투고 투쟁하고 논의하는 과정에서 좋은 결과물이 나온다고 몸소 경험하였습니다. . 앞으로의 협업 시, 해당 사항은 카카오 입사 후에도 조직생활에 큰 도움이 될 거라 생각합니다.

[Experience]

초등학교 시절 베이직(BASIC) 언어를 접한 후 게임을 직접 하는 것 보다는 만드는것을 더 좋아하게 되었습니다. 그 시절 내장 도움말 만으로 베이직을 공부하여 텍스트 기반의 게임을 만들었고 친구들에게 자랑하곤 하였습니다.

군 복무 시절 부대 급식 인원 관리 데이터베이스를 구축한 경험이 있습니다. 당시 관련된 지식이 전무했지만 MS Access와 VB 스크립트로 도움말을 참고하여 데이터베이스 및 데이터 시스템을 구축하였습니다. 이렇게 수집된 데이터는 각종 훈련 및 이벤트 시 급식 인원 산정에 중요한 자료로 쓰이게 되었습니다.

대학교 학부 졸업 후 더 깊이 있는 공부를 위하여 대학원에 진학하여 컴퓨터 네트워크를 전공하였습니다. 당시의 트렌드였던 SDN/NFV를 연구하여 "Load balancing strategy of SDN controller based on genetic algorithm"의 논문을 쓰고 졸업하였고, 현재 재직중인 회사로 취업하게 되었습니다.

[Vision]

IT 업종에 종사하면서 하루가 다르게 변화하는 기술에 대한 끊임 없는 자기 계발은 필수일 것입니다. 다만 이러한 기술에 지나치게 매몰되는 것도 경계해야 한다고 생각합니다. 그 기술이 어떤 문제를 어떠한 방식으로 해결하고자 하는지에 대한 고찰이 반드시 필요하며, 나아가 그 문제를 해결했을 때 어떤 가치를 가져다 줄지에 대한 넓은 시각을 가져야 한다고 생각합니다.

현 재직중인 직장에서는 주기적, 반복적으로 수행하는 모니터링 업무가 있었습니다. 이러한 업무는 다소 소모적이고, 담당자 부재 시 인수인계가 필요했기 때문에 직원들은 큰 불편을 갖고 있었습니다. 이러한 문제점에 착안하여 모니터링 에이전트를 제작하였고, 이는 담당자들의 단순 업무를 줄여주어 더욱 생산적인 일에 집중할 수 있게 되었으며, 인수인계로 인한 부담 또한 줄어들었습니다.

카카오에 입사 후에도 이러한 '가치'를 가져다 주는 업무를 하고 싶습니다. 사내 업무 효율화를 넘어 다양한 분야에서 전국민을 대상으로 삶을 더욱 풍요롭게 해주는 가치를 실현하고 싶습니다.

[Harmony]

24살 때부터 기타 주자로 밴드 동아리 활동을 시작하였습니다. 하나의 곡을 합주하는 데 있어 베이스는 곡의 무게감을 더하고 모든 소리를 모아주는 역할을 합니다. 드럼은 곡 전체의 박자를 이끌어나가며 기타는 곡에 색깔을 입히고 보컬은 사람이 알아들을 수 있는 언어로 음악을 표현합니다. 각자의 위치에서 각자의 역할을 이해하고 맡은 임무를 충실히 수행하며 서로의 영역을 침범하지 않을 때 하나의 멋진 음악이 완성됩니다. 밴드 활동을 하며 단순한 취미생활을 넘어 '조화'를 배웠으며, 이는 직장 내의 제 위치에서 해야 할 일이 무엇인지 알고 맡은바 업무를 최선을 다 하여 공동의 목표에 이바지 해왔습니다. 이러한 면모는 카카오 입사 후에도 조직 생활에 큰 도움이 될 것이라 생각합니다.