|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 박인수 | |  |  | | --- | --- | | 경기도 용인시 기흥구 구성로 90 |  | | 010-2376-8208 |  | | Insoo67.park@gmail.com |  | | [www.linkedin.com/in/insoo-park](http://www.linkedin.com/in/insoo-park)  <http://github.com/insoo67park>  http://github.com/wondermelon4ever  http://lucky-sugar-park.github.io |  | |

커리어의 대부분을 (SI 현장에서 아키텍트로서 그리고 개발자로서) 크고 작은 시스템의 아키텍처를 설계하고 구현했습니다. 그 과정에서 자연스럽게 어플리케이션부터 IT 인프라까지, UI부터 DBMS까지 폭넓은 경험과 지식을 얻을 수 있었습니다.

좋은 시스템을 구축하기 위해서는 좋은 아키텍처를 구현해야 하며, 좋은 아키텍처는 기업의 Business Agility 역량확보에 필수적이라는 철학을 가지고 있습니다. 다년간 (IT Infra, App 개발, Data 등 모든 분야의 아키텍처를 책임지는) Chief Architect로서 현장에서의 활동경험으로 얻은 결론입니다.

좋은 아키텍처를 구축하기 위해서는 구성원을 포함한 이해관계자간 긴밀한 소통이 필수적입니다. (특히 시스템이 커질 수록) 아키텍처는 결코 어느 한사람의 독단으로 결정되어서는 안되며, 그렇게 될 수도 없습니다. 아무리 뛰어난 개념을 가진 Big Picture를 그렸어도, 개발자가 (시간부족 등의 이유로) 따르지 않는다면 무용지물이 되기 때문입니다.

리더는 올바른 방향제시와 구성원들의 자발적인 참여와 역량을 끌어낼 수 있어야 한다는 생각을 가지고 있습니다. (수십명의 개발조직을 이끈 적이 있는) 제 경험상 가장 효과적인 리더쉽은 “솔선수범” 입니다. 소통을 기반으로 하되, 가장 어려운 기술적 난제들을 우선적으로 해결해 줌으로써 구성원들의 (할 수 있다는 자신감 상승과 더불어) 자발적인 참여를 이끌어 낼 수 있었습니다.

소프트웨어 엔지니어링 고전중에 “Divide and Conquer”라는 말이 있습니다. 아키텍처도 그렇지만 소프트웨어도 마찬가지입니다. “잘 쪼개고 잡 합치는 것” 이것만 잘하면 아무리 어려운 난제도 해결할 수 있다는 확신을 가지고 있습니다. 이 과정에서 프로그래밍 언어는 크게 중요하지 않습니다. 언어는 필요하면 학습하면 됩니다 (오히려 학습능력이 더 중요하겠군요)

우리 IT 업계는 지난 20여년 동안 몇 번의 큰 패러다임 변화를 겪었습니다. 웹 기술의 태동으로 인한 어플리케이션 개발 기술/방식 변화가 있었고, 지금 우리가 겪고 있는 (기업의 IT 환경을 통째로 바꾸고 있는) 클라우드로 인한 변화입니다. 이 과정에서 많은 엔지니어 들이 “나는 여기 까지”는 말과 함께 기술을 포기하고 다른 분야로 전향하는 것을 많이 봐왔습니다.

개인적으로 웹 기술로인한 변화와 그 이후의 다른 변화들도 어렵지 않게 적응했었는데, 지금의 클라우드로 인한 변화는 적응하는 데에 적지 않은 시간이 걸렸습니다. 그 동안 쌓아 두었던 경험/지식/기술을 거의 Zero Base에서 새로 시작해야 했기 때문입니다. (물론 이전의 지식 덕분에 남들보다는 훨씬 빨리 적응할 수는 있었습니다)

이러한 변화는 앞으로도 계속될 것입니다. IT 기술의 발전은 이런 변화를 더욱 가속화 할 것입니다. 하지만 IT 시스템의 목적은 기업의 Business 역량향상이고 아키텍처와 소스코드는 기업의 Business Agility능력을 증대 시키는 방향으로 진화해야 한다는 점은 변하지 않을 것입니다.

(거창하지는 않더라도) 한 분야에 일가를 이루고, 아키텍트로서 소프트웨어 개발자로서 족적을 남기고자 노력하고 있습니다. 저도 오픈소스를 통해서 많은 도움을 받았기 때문에, 기회가 될 때마다 오픈소스를 통해서 세상에 기여하고자 합니다.

# 주요 보유기술(역량)

### 학습능력

제가 가진 가장 뛰어난 기술입니다. 모든 개발지식, IT 지식을 오로지 구글링을 통해서만 습득하고, 제 것으로 만듭니다 회사에서의 IT 기술교육은 어지간하면 모두 거부합니다. 유일한 이유는 “구글링을 통하면 하루”면 기술을 습득할 수 있기 때문입니다 (사실 웹으로 어지간한 개발, 테스트 환경을 제공해 주는 곳이 많습니다-교육기관까지 갈 필요가 없는 이유입니다)

### 오픈소스

많은 오픈소스들이 있고, 필요하면 학습 후에 적용하면 된다는 생각을 가지고 있습니다. Springframework 관련, Kafka, Kafka connect, Rabit MQ, Elastic search 관련, Nifi, Kubernetes 관련, MySQL, PostgreSQL, Next, React, Material-UI 등을 잘 다룹니다.

### 알고리즘

알고리즘은 아주 잘 하는 편은 아닙니다만, 역량 향상을 위해서 꾸준히 노력하고 있습니다. 문제풀이 사이트 백준에서의Platinum-1 수준은 되는 것 같습니다. 그 동안 여러 사이트에 걸쳐서 다양한 난이도의 문제를 700 개 정도 풀어본 것 같습니다.

### 아키텍처

소프트웨어 아키텍처, 기술 아키텍처 ( IT 인프라 부분 ), 데이터 아키텍처에 대한 많은 경험이 있습니다. 주요 대상이 되는 품질속성으로는 “가용성”, “보안”, “사용용이성”, “테스트용이성”,”성능”,”유지보수용이성”이 있고,이런 주요 품질속성들을 대상으로 시나리오 기반의 아키텍처를 설계하고 구현하는 데에 탁월한 능력이 있습니다.

아키텍처를 수립할 때에는 이해관계자들과의 소통을 중요시 여기며, 특히 비즈니스 담당자들의 지식을 십분 활용합니다.

### FRONTEND

React.js를 잘 사용하며, Next.js를 적용해서 CSR/SSR 방식을 구현할 수 있습니다. 특히 오픈소스인 Material-UI를 능숙하게 다루며, 별도의 웹 디자이너 없이도 수준급의 UI를 개발할 수 있습니다 (vue.js는 Hands-on 정도 해본 수준입니다)

Webpack 5의 Module Federation 기술을 적용한 Micro frontend 아키텍처를 구현할 수 있습니다. 오픈소스인 Nginx와 Minio를 적용해서 Micro frontend 아키텍처 기반의 CDN 서비스를 구축할 수 있습니다.

React.js + Next.js + Webpack 5 Module Federation기반의 Frontend 디자인 패턴과 Coding Convention에 많은 연구를 하고 있습니다.

### backend

특별히 가리는 프로그래밍 언어는 없습니다만 자바, 파이선, C++, C# 언어를 사용할 수 있습니다. 최근에는 자바언어를 주로 사용하고 있습니다.

스프링 프레임워크와 스프링 부트를 잘 사용합니다. 자바 언어로 Kubernetes Cluster와 연계하여 Kubernetes Resource를 생성/삭제/모니터링 할 수 있습니다.

파이선은 FastAPI를 기반으로 하는 REST API 개발을 할 수 있습니다 (Django 경험은 없음) Uvicorn, Gunicorn과 조합하여 멀티 쓰레딩 환경을 구현할 수 있습니다. AI 모델은 구글의 코렙 (<https://colabs.research.google.com>) 환경에서, 이미 만들어진 여러 모델의 분석, 구동할 수 있는 정도의 수준입니다.

파이선 언어는 “정말 사랑스러운데” 특히 Decorator (@) 개념을 너무 좋아합니다. 자바에서는 스프링 프레임워크가 없었으면 정말 힘들었을 IoC (Inversion of Control), Injection 문제들을 Decorator를 활용해서 너무 쉽게 해결할 수 있었습니다. 저는 로그와 DB Transaction 처리 부분에서 Decorator를 적용한 경험이 있습니다.

C++언어로는 미국의 스피드게이트 업체인 “르넬”사의 ACS 시스템과 연계해서 얼굴인식 기반 “출입통제시스템”을 개발한 경험이 있습니다. 몇 년 지나긴 했지만, 기억을 되살려보면 일반적인 노트북 1대로 CCTV 4대까지 운영이 가능했었던 것 같습니다. 주요 활용했었던 라이브러리로 OpenCV와 자체 개발된 얼굴인식 모듈이 있었습니다.

언어와 무관하게, 백엔드 부분은 특히 소프트웨어 설계가 매우 중요한 영역이라는 판단입니다. 이해관계자들이 시스템을 바라볼 때에 요구하는 대부분의 품질속성들이 백엔드 부분에 해당되기 때문입니다.

관련된 많은 디자인 패턴들이 있는데, 2,000년대 초반부터 유행했었던 대부분의 디자인 패턴들을 매우 능숙하게 실무에 적용할 수 있습니다.

### IT Infra

Kubernetes Cluster와 오픈소스를 활용해서 CI/CD (Continuous Integration/Continuous Deployment)를 구축할 수 있습니다. Kubernetes상에서 Jenkins와 ArgoCD를 설치 및 사용할 수 있습니다.

Kubernetes의 주요 서브 프로젝트인 Istio를 설치하고 운영할 수 있습니다. 특히 Istio + Oauth-proxy + Keycloak을 활용해서 서비스 매쉬의 중요 기능인 인증관리를 수행할 수 있습니다.

데이터베이스 (MySQL, Maria DB, PostgreSQL)을 (관리자 수준에서) 잘 다룹니다 (물론 시간이 지나면서 자세한 커맨드가 잊혀진 것도 있는데, 우리에게는 구글이 있으니까요)

리눅스 서버를 잘 다룰 수 있습니다 (관리자 수준까지는 아닙니다만, 개발 또는 아키텍처에 필요한 정도의 지식과 경험은 있습니다)

네트워크에 대한 평균이상의 지식과 경험이 있습니다 (네트워크에 대한 기본지식은 클라우드 내에서 시스템을 구축할 때에 꼭 필요한 것입니다)

# 경력

### 2023년 6월 ~ 현재

##### 통합공사관리시스템 아키텍처 수립 – 삼성SDS

삼성SDS 고객사의 통합공사관리시스템 아키텍처를 수립하고 구현함. 삼성SDS의 클라우드인 SCP (Samsung Cloud Platform)에 탑재되어 서비스될 수 있도록 네트워크, 서버, CDN, 통합SSO 인증, 마이크로 서비스, 마이크로 프론트엔드 등을 설계했습니다

특히 K8S + Istio + Oauth2-Proxy + Keycoak 방식으로 사용자 인증 시스템을 파일럿 구축 및 검증하였습니다

또한 (일부 업무를 대상으로) React + Next + Nginx + Minio 조합으로 Micro frontend를 구현 및 검증하였고, Frontend 소스코드의 골격을 제공하였습니다

### 2022년 11월 ~ 2023년 5월

##### 3dx 도면변환 서비스 개발 – 삼성SDS

다양한 유형의 CAD 도면을 3d Streaming 형태의 서비스가 가능하도록 변환해 주는 시스템입니다. Kubernetes Cluster 내에서 구동되어 SaaS 방식의 서비스를 지원할 수 있도록 구현되었습니다. 실제 변환을 담당하는 Worker 프로세스와 Worker를 K8S의 POD 형태로 생성/삭제/관리하면서 변환 요청을 관리해 주는 Orchestrator로 구성되었습니다.

Docker, Kubernetes, Kafka, Kafka connect, Springboot, Springframework, jenkins, ArgoCD, Nexus Repository 등을 기술을 적용하였습니다.

### 2021년 2월 ~ 2021년 12월

##### 솔루션경쟁력 강화/삼성SDS

Chief architect로서 사내 다양한 SW 프로덕트의 품질향상을 위해 개선점을 도출하고 이에 맞는 아키텍처를 제시 및 설계 하였음.

제조 현장에서 앓고 있는 고질적 문제점을들 도출하고 IT 기술을 활용해서 해결할 수 있는 방안을 제시하고 적용했습니다 (Big data web chart, Big data web grid 등)

특히 웹 차트, 그리드의 경우, 제조업의 특성 (수백만 이상의 데이터를 클라이언트에서 렌더링) 하는 특성 상, 기존 상용 제품으로도 만족하지 못하는 상태 였으나 SSR 방식의 차트방식으로 전환, Indexed DB 활용 등의 아키텍처를 제시하여 문제를 해결하였습니다.

### 2018년 1월 ~ 2019년 1월

##### WCS시스템 개발/아이마켓코리아

아이마켓코리아 신규창고 (동탄)에 창고제어시스템을 구축하였습니다. 주요 Picking 시스템으로 독일의 쉐퍼 시스템, 국내 DPS 시스템 (Digital Picking System) 일본의 Sorter 업체인 호쿠쇼 시스템과 연계해서 배송라인까지의 물품을 제어하는 시스템입니다.

대부분의 현장 물류 시스템들은 PLC 기반으로 제어가 이루어지기 때문에 WCS는 PLC와의 통신은 필수적이었습니다. 이 때에 PLC와의 통신을 처음으로 해 보았습니다. 시간과 자원이 부족해서 (아키텍트로서의 자존심을 버리고) 일단 동작하는 코드로 만족했었던 아픈 기억이 있습니다.

React.js, Three.js, Netty, Springboot, MySQL, 자체개발 application framework등의 기술을 적용했습니다.

### 2013년 1월 ~ 2015년 1월

##### VA4S (Vidio Analysis for Samsung SDS) 개발/삼성SDS

CCTV를 분석해서 보안관련 이벤트를 감지하고 시각적으로 위치를 알려주며 알람을 발생시키는 시스템 입니다. 개발을 총괄하는 PM + 아키텍트 + Core 개발자로 참여했으며 개발자만 40명 가까이 되었던 것으로기억합니다.

주요 기능으로는 단위이벤트 탐지 (쓰러짐, 뛰어감, 경계침범, 특정차량번호 등)와 복합이벤트 탐지 (단위 이벤트를 플로우 형태로 엮은 복합적인 이벤트), 객체/이벤트 검색 등의 기능을 제공했습니다.

객체는 사람/차량/미확인 3가지 부류로 구분하였고 사람의 경우에는 얼굴인식과 옷의 색상에 의한 검색을 지원했고 모든 UI는 지도를 기반으로 했습니다.

주요 사용기술로는 얼굴인식 라이브러리, Google Map API, (NoSQL DB인) Hbase, Hadoop의 MapReduce, MariaDB 등이 있었습니다.

특히 1,000개 이상의 CCTV로부터 수억개의 사람얼굴을 등록하고 특정 얼굴을 검색할 때에 수초 이내에 검색결과를 보여준 경험은 매우 짜릿했습니다. NoSQL DB를 어떻게 사용해야 효율적인지에 대한 통찰을 얻는 계기가 되었습니다.

### 2010년 1월 ~ 2011년 1월

##### 차세대 MES 시스템 설계 컨설팅/삼성전자

아키텍처 부분 PL로 참여 하였으며 SA/TA/DA 부분에 대한 설계 컨설팅을 수행했습니다. 당시 화두가 300mm 웨이퍼 도입과 이로 인한 빅데이터 실시간 분석/처리 방안 이었는데, 이 문제를 어떻게 해결할 것인가에 대한 토론을 정말 많이 했었던 것 같습니다.

심지어 소프트웨어 아키텍처의 대가이신 카네기 멜론의 데이비드 갈랜 교수님을 모시고 1박 2일 워크숍을 했던 기억도 있습니다.

당시 데이터베이스로는 상용기반 메모리 DBMS, 실시간 처리로는 ESPER라는 CEP (Complex Event Processing) 오픈소스를 집중적으로 고민했었습니다.

시스템이 너무 크고 구성요소들이 너무 많아서 미래모형을 설계하는 데에 정말 시간도 많이 걸리고 머리가 지끈지끈 거렸던 경험도 있습니다.

### 2009년 1월 ~ 2010년 1월

##### 차세대 NEIS 시스템 컨설팅/KERIS (한국교육학술정보원)

약 2,000억원에 가까운 시스템의 아키텍처를 설계한 (제 인생에서) 가장 큰 규모의 작업이었습니다. 프로젝트 기간이 6개월, 사전준비 2개월, 사후 서비스 4개월 등 정말 오랜 시간동안 밤을 불태웠던 기억이 있습니다.

시간이 많이 걸려었던 이유는 매번 수정이 발생했으며 거의 무한 (작업후 검토 수정)이 이루어졌기 때문입니다.

아키텍처 부분 PL로서 참여했었으며 AA/SA/TA/DA 부분의 아키텍처를 팀원들과 머리를 맞대로 고민했었습니다.

한가지 기억에 남는 것은 당시 화두가 Active-X의 제거와 RESTful API 였었는데, RESTful API가 기술표준이 되리라고는 생각하지 못했었던 것 같습니다. React.js나 Angular.js가 지금처럼 자바스크립트 활용 SPA를 대세로 만들것이라고는 더더욱 생각을 못했었던 것 같습니다.

주요 사용기술로는 QAW (Quality Attribute Worksop), 그리고 어플리케이션 개발에 대한 전반적인 지식, IT에 대한 전반적인 지식 등이 총 동원 되었었습니다.

### ~ 2008년 12월

##### 기타/기타

기타 공공분야, 국방분야, 제조분야 등에서 크고 작은 프로젝트를 수행했었습니다. 주로 소프트웨어 아키텍트로서 참여하였으며 코어모듈 코딩, 개발표준수립, 개발환경수립, 아키텍처 수립/구현 등의 업무를 수행했었던 것 같습니다.

# 자격

##### CSDP (Certified Software Development Proficiency)

2011년 11월 취득 (IEEE Computer Society)

##### Chief Architect

2010 (Samsung SDS)

##### Senior Software Architect

2011 (KOSTA: Korea Software Technology Association)

##### SW Proficiency (Professional)

2015 (Samsung SDS)

##### Google Cloud Associate Cloud Engineer

2022 (Google)

# 학력

##### 수도전기공업고등학교

1983 ~ 1985

##### 숭실대학교

학사, 컴퓨터공학과, 1996~1999

# 수상

##### 포스코ICT 사장상

2021. 07, ERP 시스템 성공적 구축 공로 (전자구매 부분)

# 활동

##### 번역서 (소프트웨어 아키텍처 평가-공동번역)

Evaluating Software Architectures (폴클레멘츠, 릭캐즈먼, 마크 클라인)

2011년 (공통번역: 이석준, 백창현, 박인수)