

개발 조직을 민첩한 원팀으로 발전시킨 여정

FEAT. 실패해도 괜찮아



Introduce the speaker

김정우 @ AhnLabCloudMate Jungwoo (Jayden) Kim

現) 안랩클라우드메이트 개발1팀 리드
現) IT여성기업인협회 SW캠프 멘토
前) 모두투어 ERP개발팀 리드

웹, 서버, 클라이언트 앱을 개발하고 그 애플리케이션이 서비스 목표를 달성하도록 만드는 일을 하고 있습니다.

기술을 쉽게 풀어서 설명, 공유하는 것을 즐겨워 합니다.



Question!



Table of Contents

- 01** 새로운 개발 조직 구성
- 02** 프로젝트 시작하면서 겪은 다양한 시행착오
- 03** 협업 문화 만들기
- 04** 첫번째 프로덕트 출시
- 05** 새로운 프로덕트 개발 및 확장
- 06** 앞으로의 여정

사업 환경 이해하기

단기 goal과 함께 TF 먼저 수행

직무명세 만들고, 인원 섭외

정규 팀 신설

01

새로운 개발 조직 구성

일 벌리는거 좋아하는 편

QUESTION

우리 회사에는 어떤 개발 조직이 필요할까



Question!

01

사업 환경 이해하기



02

우리의 강점은 무엇인지?



QUESTION

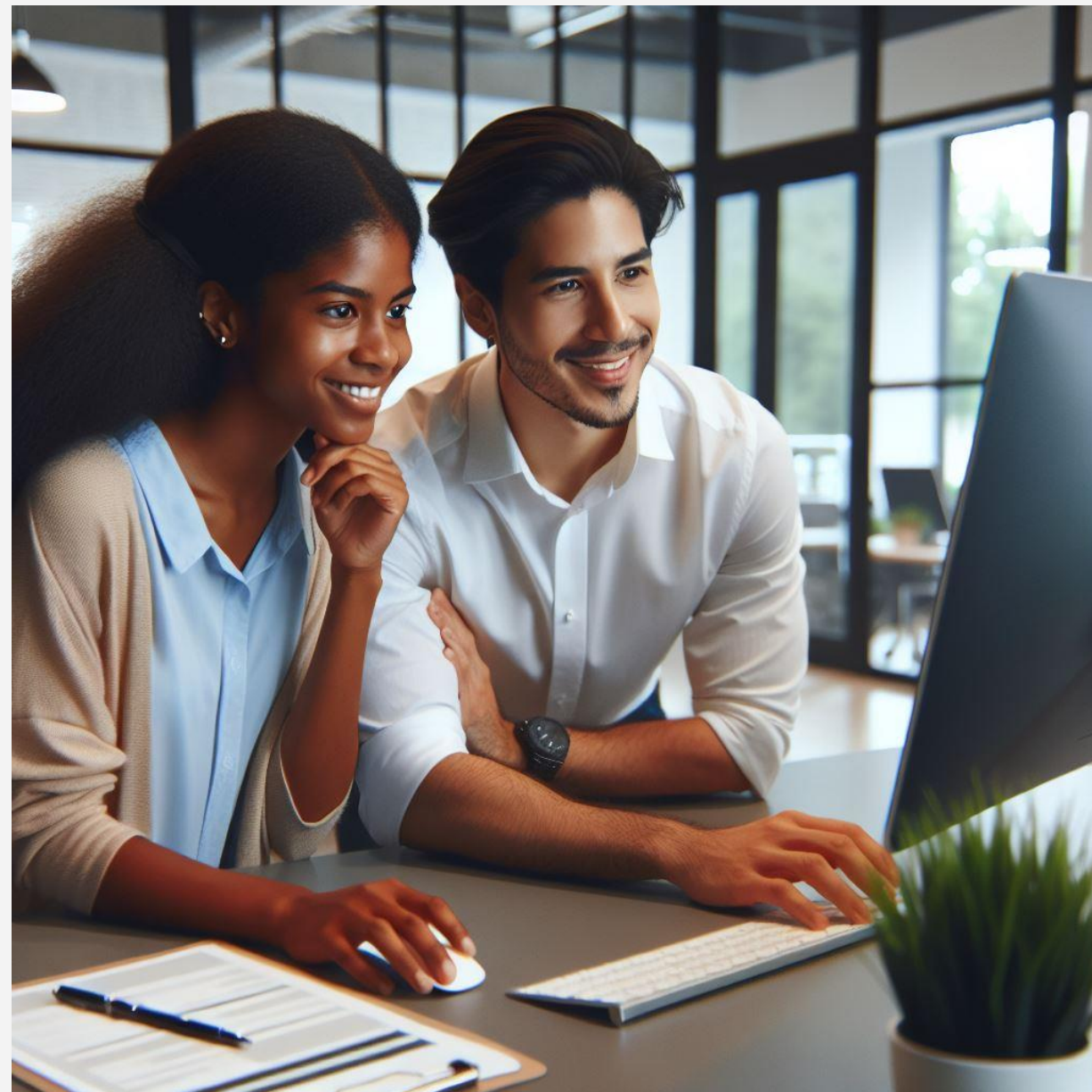


Question!

우리 회사에는 어떤 개발 조직이 필요할까

03

고객에게 제공해야 할 가치



04

단기, 중장기 목표



심지어 개발자의 장기적인 커리어까지 고민이 필요



우리 회사에 필요한 개발 조직은,

1. 기존 클라우드 제공업체와의 차별성이 필요했고
2. “이거 만들어줄래?” 했을 때 함께 논의하고 만들어줄 수 있는 팀이 필요했고
3. 큰 제품을 오랫동안 만드는 것 보다는 짧은 호흡으로 업데이트를 빈번히 수행하여 시장 반응에 대응해야 했으며
4. 여러번 새로운 시도를 통하여 고객이 만족할 수 있는 소프트웨어를 만들어 내야 했습니다.

ANSWER

단기 성과를 위해 TF 결성



Question!

GarlicBread IT Story

Home Categories Tags About

2달 만에 자사 솔루션 MVP 만들기 (5명이서...)

자사 솔루션 개발이라는 프로젝트가 주어졌습니다. MVP(Minimum viable product)를 2개월 동안 개발해야 하는 상황이었습니다. 이번 프로젝트를 스프린트 방식으로 개발해보고자 하는데, 스프린트 기간은 개발해야 하는 상품과 서비스, 상황에 따라 달라질 수 있습니다. 자사 솔루션을 MVP로 개발하는데 필요한 핵심 기능 구현을 목표로 세우고 각 스프린트를 2주씩 진행하기로 했습니다.

MVP(Minimum Viable Product) 개발이란? '최소한의 노력으로 최대한의 고객 검증이 가능한 버전의 제품'을 개발하는 것! 검증하고자 하는 핵심 기능을 짧은 시간 안에 만든 뒤 고객 피드백에 따라 다음 회차 개발에 추가할 기능들을 결정합니다.



PoC



Prototype



MVP

스프린트 설정하기

자사 솔루션의 핵심 기능 별로 스프린트 목표를 4개로 쪼개고 각 기간은 최대 2주(근무일 10일)로 설정하였습니다. 그리고 각 스프린트마다 구현해야 할 결과물에 대하여 팀원들과 협의를 통하여 결정하였습니다.



함께 개발했던 크루들이 모두 훌륭
했던 덕분에, 앞으로 다양한 프로덕
트를 출시할 수 있겠다는 공감대가
형성

<https://rokag3-gb.github.io/sprint-experience/>

그래서 정규 팀을 신설하기로 했어요.



Question!

1. TF를 통해 내부 HR에 대한 신뢰와 확신
2. 팀의 Mission, Vision, Core Value 구상
3. 직무 명세 작성
4. 인원 섭외 - 자네 혹시 개발해보지 않을텐가?
5. 정규 팀 신설을 위한 기안 작업
6. 중장기 목표 설정

Our Mission, Vision and Core Value



Question!

Mission - 차세대 Cloud 관리 SaaS 솔루션 개발

Vision - 우리를 통하여 Cloud를 공급받는 모든 사용자가 편리하게 이용할 수 있는 Cloud 관리 SaaS 솔루션을 만듭니다.

Core Value

정보의 투명성

적극적이고 효율적인 커뮤니케이션

선의 기반의 비판적인 사고

자율에 비례하는 책임

일하는 방식

좋은 개발자란 무엇인가?

잘하는 것에 집중하자

테크 스택 결정

개성 강한 개발자들

02

프로젝트 시작하면서 겪은 다양한 시행착오

복무신조! 하나! 우리는!

팀이 만들어지고 일단 맛있는걸 먹으러 갔습니다.



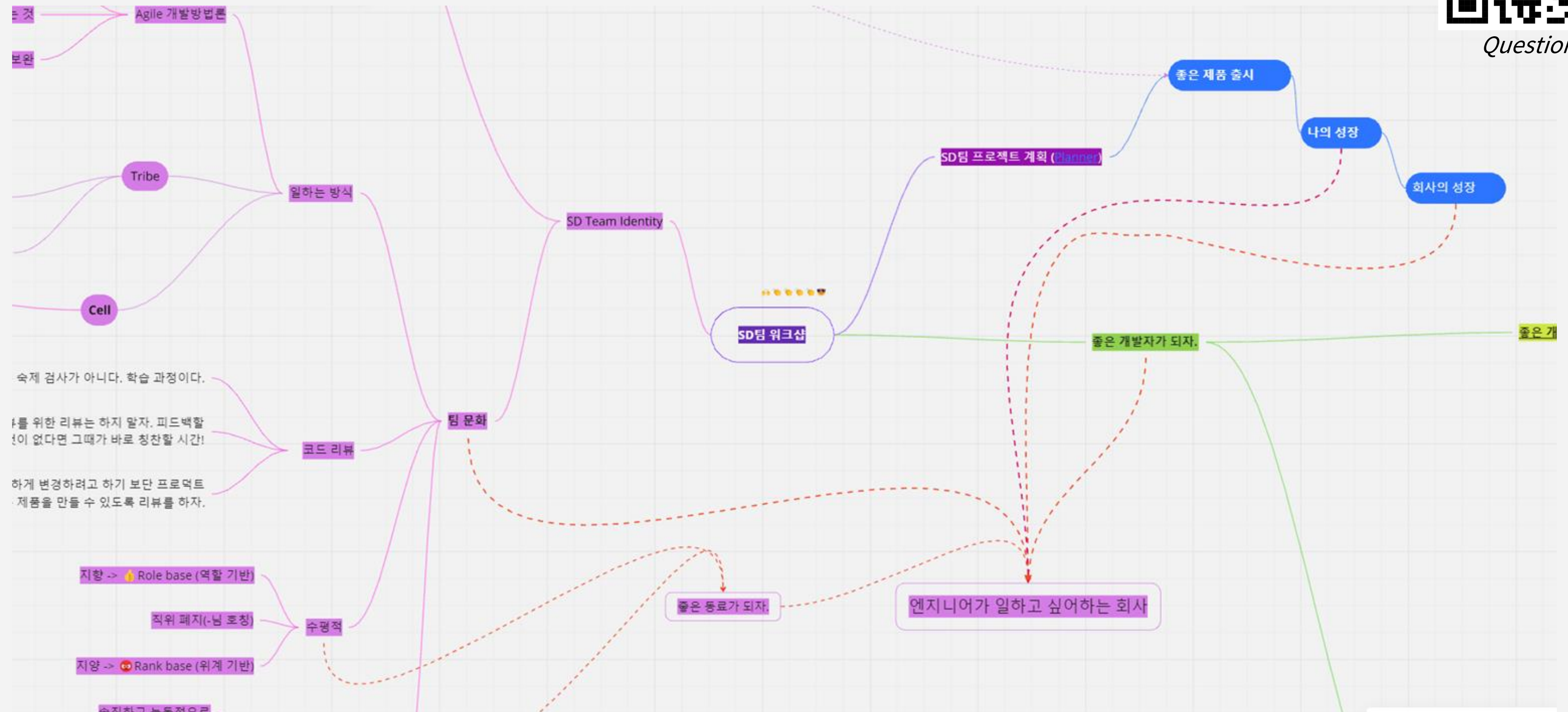
Question!



워크샵도 하고,



Question!



워크샵도 하고, 실내 야구 게임도 했습니다.



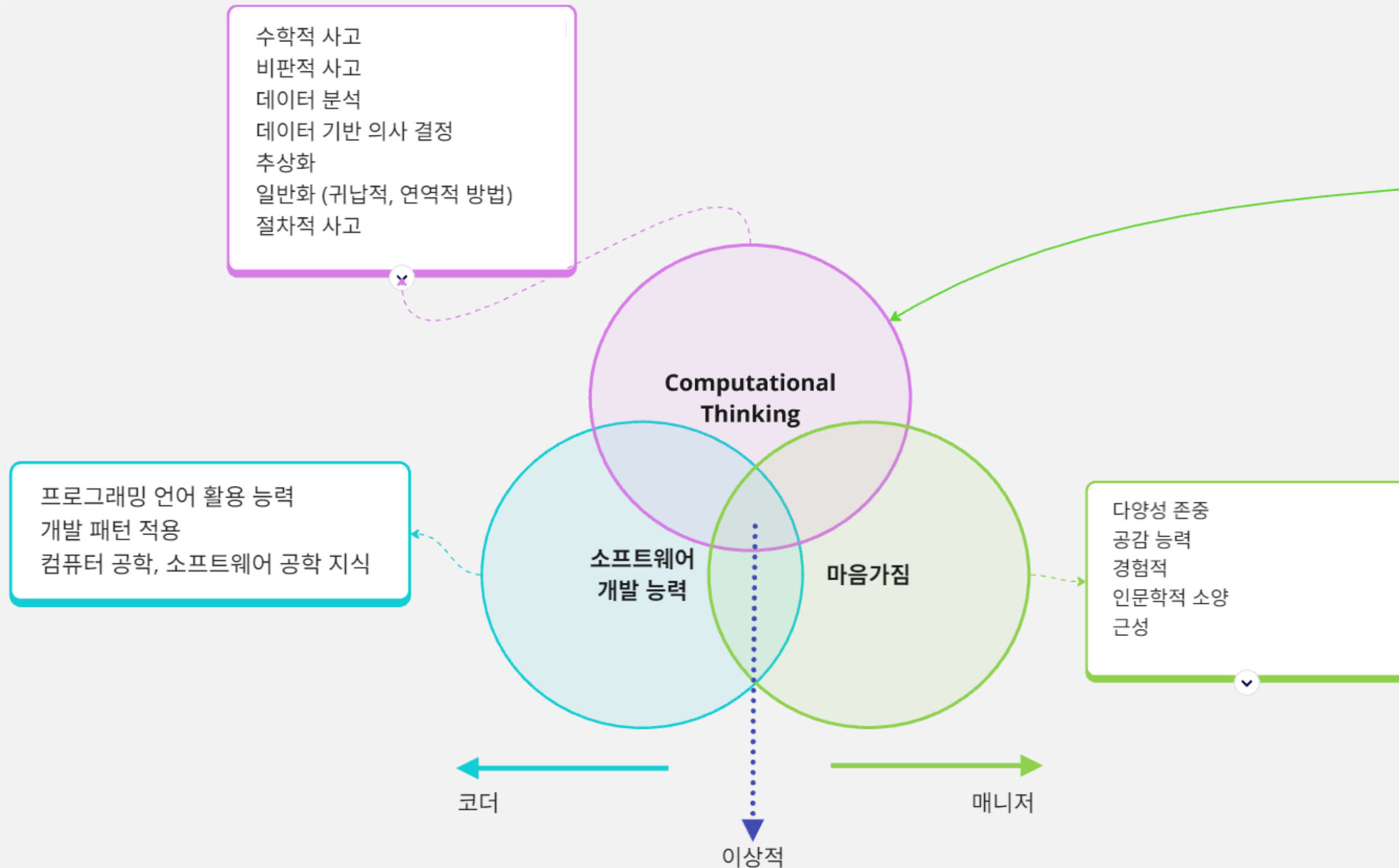
Question!



워크샵을 통해서 “좋은 개발자”를 정의했어요.



Question!



업무 방식



Question!

1. 팀 내에서는 모두 완벽한 수평 조직
2. 내가 작성한 코드가 비즈니스에 어떤 도움이 되는지 고민
3. 메이커 조직으로서 대중음악 vs. 예술가요 에 대한 고민 항상 필수
4. 잘하는 것에 집중
5. 몇가지 테크 스택 선정 (Python, Golang, C#, JavaScript)
6. 용어 사전
7. 바퀴 재발명 금지 (이미 있는 것 잘 활용하기)
8. 우리의 문화, 대화하는 방식

7. 우리의 문화, 대화하는 방식

너무나 중요함!!



Question!

1. 모든 발언 속에 타인에 대한 배려, 역지사지, 자기 객관화가 필요
2. 스페셜리스트가 되기 위해 노력하는 동료들을 인정하는 자세 필요
3. 스페셜리스트, 제네럴리스트 차이 존중
4. 본인의 경험담이 전부라고 생각해서도 안되며, 동료의 기술과 경험을 경청할 수 있는 자세 필요
5. 모든 기술적 주장, 견해에는 증명 가능한 근거가 필요
6. 과거의 누군가가 작성해놓은 코드에 대한 비난 자제 (그게 나 일 수도 있음)

7. 우리의 문화, 대화하는 방식

너무나 중요함!!



Question!

7. 일을 시작하기 전에 **목적**, **기간**, **목표 결과**를 미리 생각
8. 트렌디한 최신 기술을 맹목적으로 신뢰하는 것 경계
9. 전통적 라이브러리만 고집하는 것도 경계
10. 어차피 남이 만들어놓은 프레임워크, 라이브러리 사용하는거면서 기술적 우위 논하면서 에너지 낭비하는거 금지
11. 자율과 비례하는 책임. (모든 자유에는 댓가가 따름)
12. 연차계는 자기 마음대로. 선보고 없음.

K-애자일

우리 조직에 딱 맞는 스프린트

빛나는 실패

코드 리뷰

03

협업 문화 만들기

실패해도 괜찮다.

협업 문화 거버넌스



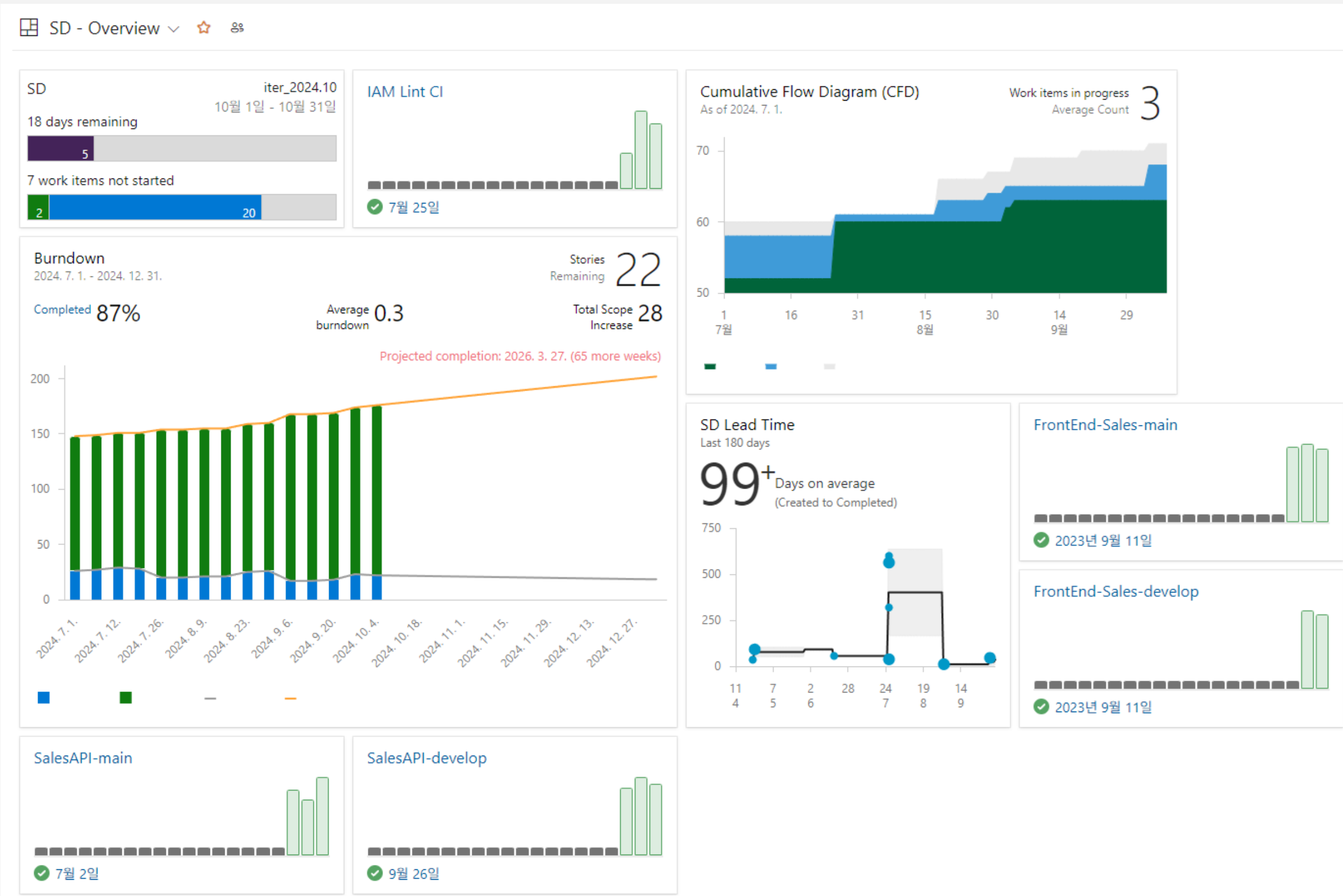
Question!

1. Agile 개발방법론 (빠른 Waterfall 아님)
2. 하나의 feature를 작게 구현하고 점진적으로 업그레이드
3. Agile process를 조직 성과 관리 측면으로만 바라보면 구성원들이 피폐해짐
4. 실패했더라도 과정 속에서 배운 것들 회고하고 보완 - **좋은 실패** 🤔
5. 각 프로덕트에 맞는 sprint iteration 설정 (2weeks / 4weeks)
6. 주기적인 Scrum meeting을 짧게 진행 (standup meeting)
7. R&R을 너무 타이트하게 따지지 않고 내 주변 업무도 내 업무 처럼

Agile Process 가속화 도구



Question!



Agile Process 가속화 도구



SD

Taskboard Backlog Capacity Analytics

iter_2024.10 Person: All

+ New Work Item

10월 1일 - 10월 31일
18 work days remaining

Collapse all

New Active Resolved Closed

3104 SmartReport 백엔드 수정 사항

Active

Kwangseok Moon

2928 UI 컴포넌트 구성

2929 테이블 구성

3034 서비스계정 관리 page 신규 개발

3035 서비스조직 관리

3105 MonthlyReport 백엔드 수정 사항 기능 개발

Kwangseok Mo...

3065 UI 컴포넌트 수정

Jinho Kim

3079 columnChooser위치 Button 누르면 나오도록 수정

Haeun Cho

3080 ServiceAccount 페이지 개발

Yeawon Lee

3066 TextBox, NumberBox, TextArea 등 컴포넌트 수정

Yeawon Lee

3078 페이지네이션 Button, Button Group

Haeun Cho

3081 multiSelect시 모든 선택된 옵션 하이라이트

Haeun Cho

3095 DataGrid UI 일부 수정 작업

Yeawon Lee

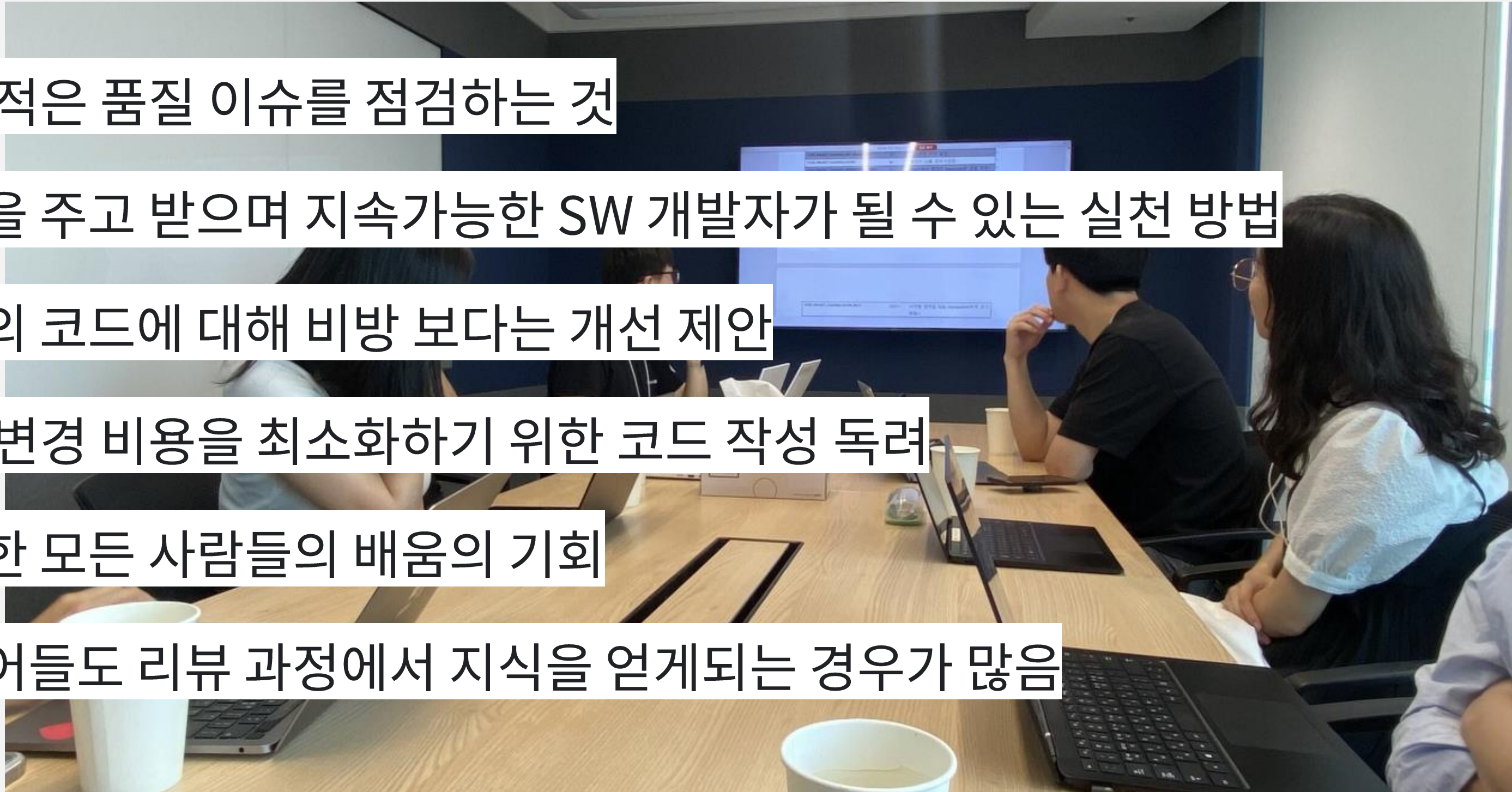
Question!

코드 리뷰 거버넌스



Question!

1. 주 목적은 품질 이슈를 점검하는 것
2. 배움을 주고 받으며 지속가능한 SW 개발자가 될 수 있는 실천 방법
3. 타인의 코드에 대해 비방 보다는 개선 제안
4. 향후 변경 비용을 최소화하기 위한 코드 작성 독려
5. 참여한 모든 사람들의 배움의 기회
6. 리뷰어들도 리뷰 과정에서 지식을 얻게되는 경우가 많음



코드 리뷰 거버넌스



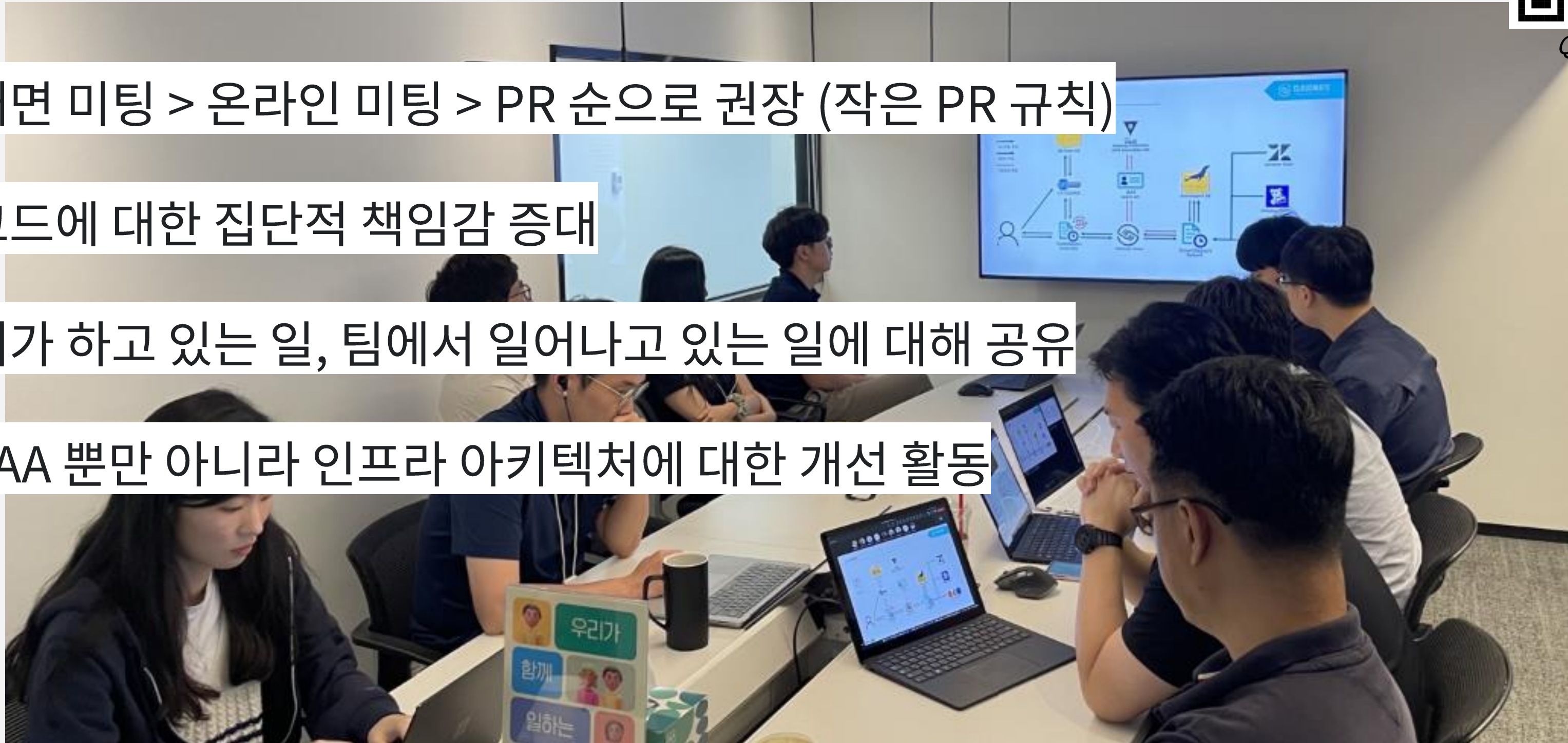
Question!

7. 대면 미팅 > 온라인 미팅 > PR 순으로 권장 (작은 PR 규칙)

8. 코드에 대한 집단적 책임감 증대

9. 내가 하고 있는 일, 팀에서 일어나고 있는 일에 대해 공유

10. AA 뿐만 아니라 인프라 아키텍처에 대한 개선 활동



스프린트 회고, 프로젝트 회고 (3L, 4L)



Question!

1. 스프린트 리뷰, 코드 리뷰와 프로젝트 회고는 성격이 다름.
2. 프로젝트 회고는 개발 조직만 참여하고, 개발한 기능 또는 코드에 대한 리뷰 뿐만 아니라 프로젝트 전체 과정에 대해 돌아보는 것.
3. 프로젝트 회고 시 PM/PO 는 facilitator [퍼실리테이터] 역할이 되어 서로를 탓하거나 변명하는 시간이 되지 않도록 회고 시간을 이 끌어야 하며, 가장 영향력이 큰 개선책을 세워나갈 수 있도록 독려해야 함.

👉 여기서 결정된 개선 방향을 다음번 스프린트 때 반영.

3Ls 회고법 (Liked - Learned - Lacked)

비교적 간단하게 시행할 수 있는 회고법. 스프린트를 진행하면서 느낀 점을 되돌아보면서 구성원들과 함께 부족한 점은 메울 수 있고, 장점은 나눌 수 있는 기회를 마련할 수 있음. 프로젝트 중반에 활용하면 특히 좋음.

1. **Liked:** 만족스러웠던 점
2. **Learned:** 깨달았거나 배웠던 점
3. **Lacked:** 놓쳤거나 부족했던 점

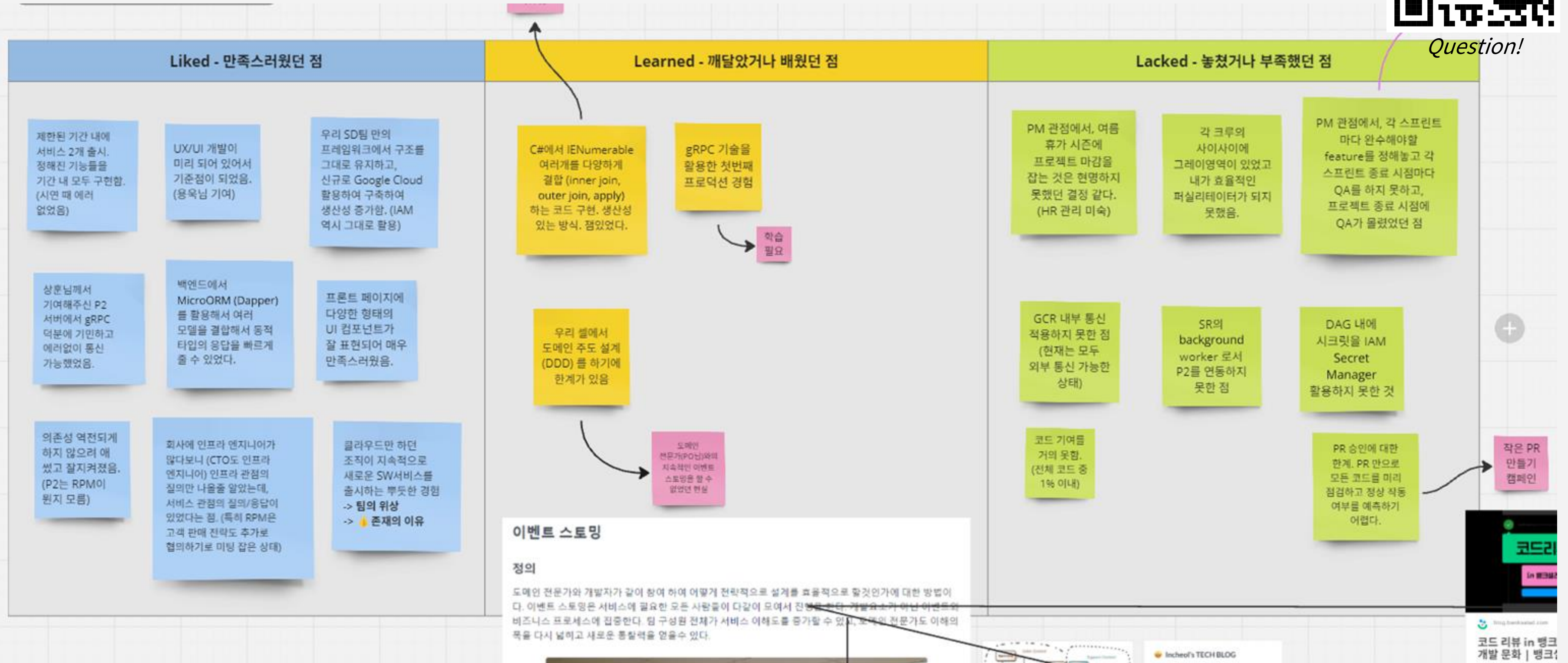
💡 **Facilitator** [퍼실리테이터]

퍼실리테이터란 목적을 달성하기 위하여 구성원 간 상호작용을 촉진 시킴. 구성원들의 모습을 관찰, 경청하여 목적을 달성할 수 있도록 격려하고 돕는 역할.

프로젝트 회고 예시 - 1 / 3



Question!



프로젝트 회고 예시 - 2 / 3



Question!



Liked - 만족스러웠던 점	Learned - 깨달았거나 배웠던 점	Lacked - 놓쳤거나 부족했던 점
개인적으로 프로젝트 완성도가 만족스러웠고 원하던 기능들이 전부 구현됨 프론트엔드쪽에 이런거 만들어 주세요 하면 다 만들어줌	DB 구조를 처음부터 만들어본게 처음이라 많이 배웠음 IAM 시크릿 기능을 외부에서 사용할 수 있다는것을 알고 있었지만 실제로 사용한것은 처음이었고 내가 한번 사용해봐서 다른 사람들에게도 사용해 보라고 해볼 수 있을것 같음	프로젝트 계획을 짜고 시작한게 아니라 일단 만들면서 맞춰보다보니 나중에 수정을 많이 하게 된 점 급하게 만들다 보니 테스트를 많이 진행하지 못하였음 프로젝트 협의를 완벽하게 하지 않고 진행하다보니 만들고나서 요구사항이 많이 들어옴

프로젝트 회고 예시 - 3 / 3



Question!



김 [redacted] ?



AhnLab Cloud
가즈아!!!!





안철수가 변했다!

존경하는 대주주님

김 [redacted]

Liked - 만족스러웠던 점	Learned - 깨달았거나 배웠던 점	Lacked - 놓쳤거나 부족했던 점
직접 경험하는 기술 스택으로 고매사가 원하는 기능을 구현하였다. SAM (segment-anything-model) 라이브러리, fabric.js, vue 등... 기존에 하지않은 SM 업무를 시작부터 인수인계를 받고 현재까지 문제없이 잘 진행 된 점 회사가 안랩에게 인수	검증된 오픈소스더라도 치명적인 버그가 존재할 수 있다는 것을 배움 (fabric.js의 브러시) - 너무 라이브러리에 의존하지말자 무차별적인 프레임워크 도입은 오히려 독이되는 듯 (lge-pri 프로젝트들) 개발환경의 중요성 Wasm 최고	Git branch 전략 멘탈 관리 이해하지 못하고 지나간 기술 billing3.0 프로젝트 준비를 하지 못함 canvas 에 대한 지식 부족, vue 생명주기, 렌더링과 같은 기본적인 지식 부족으로 인한 이슈 상대적으로 기술적인 성장이 없어 아쉬움

클라우드 그런거 잘 모름

크런치 모드

첫번째 프로덕트 출시!

프로젝트 회고

04

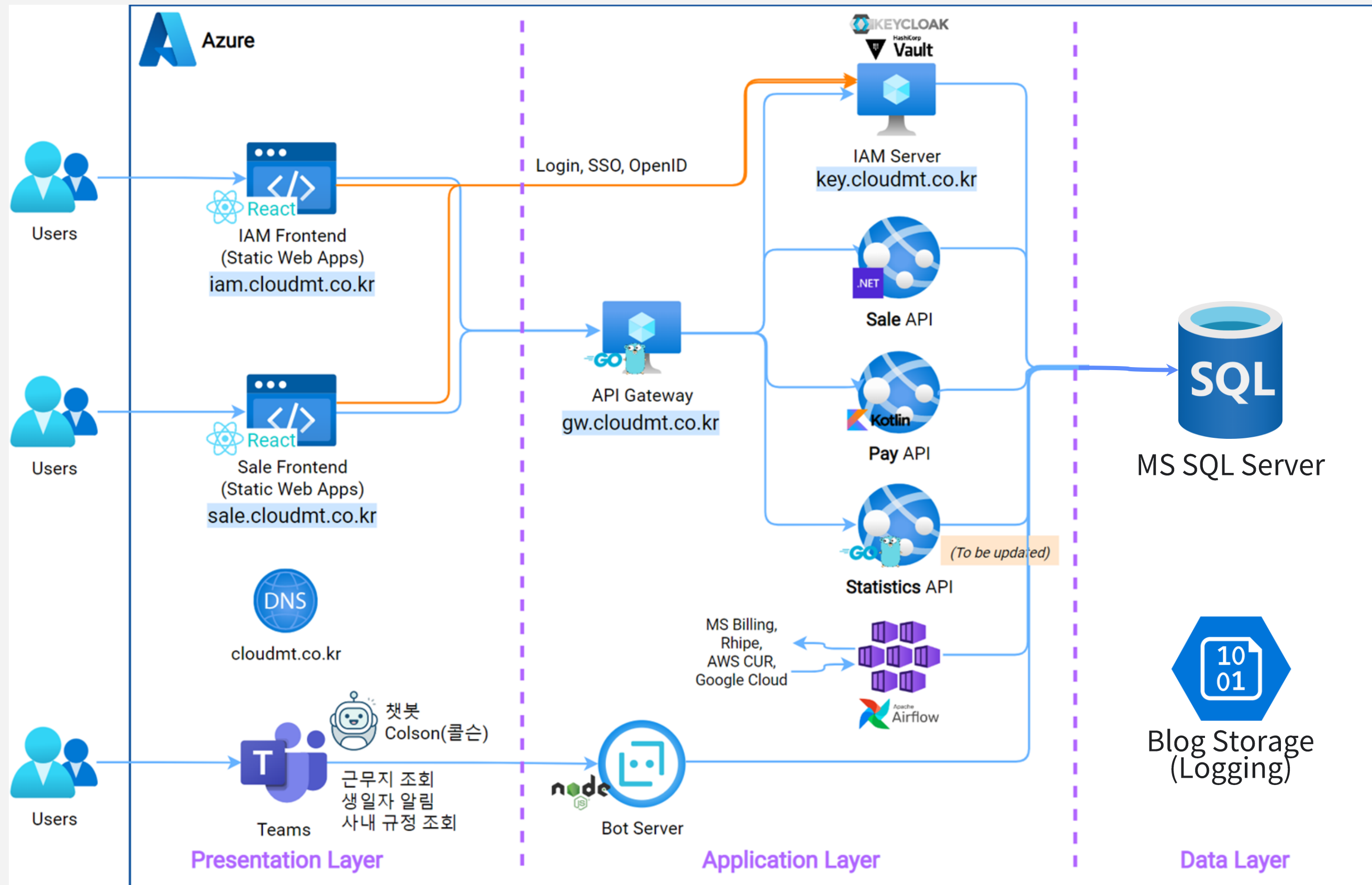
첫번째 프로덕트 출시

폭풍야근

기초 Infra Structure 구축



Question!

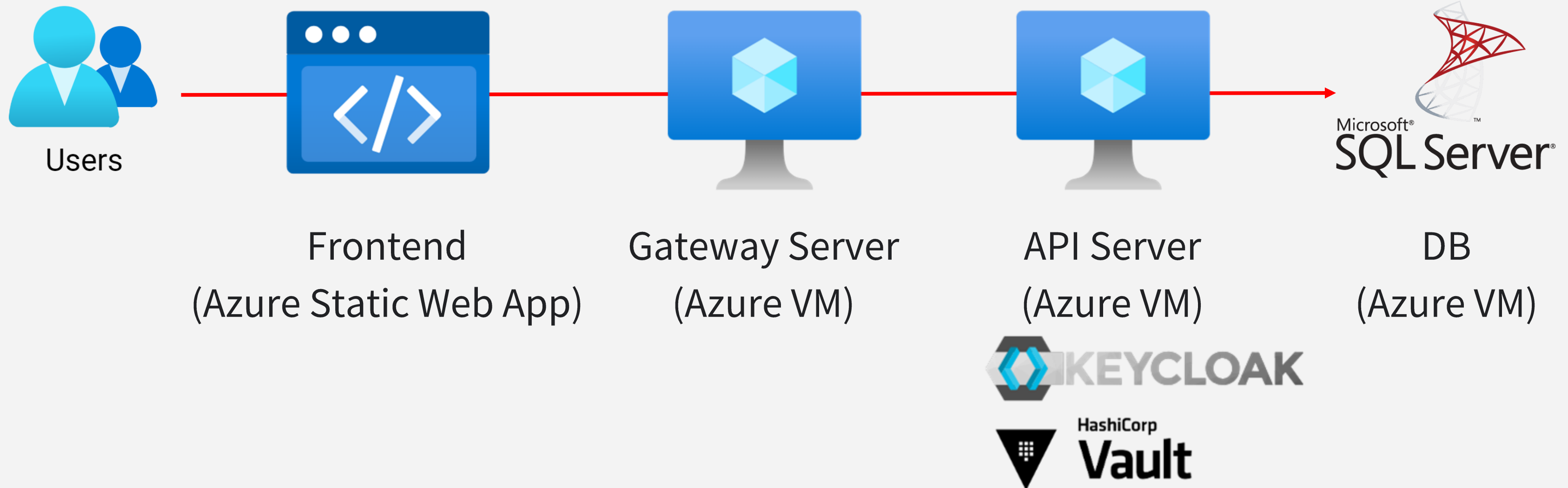


첫번째 프로젝트, Merlin [머린]



Question!

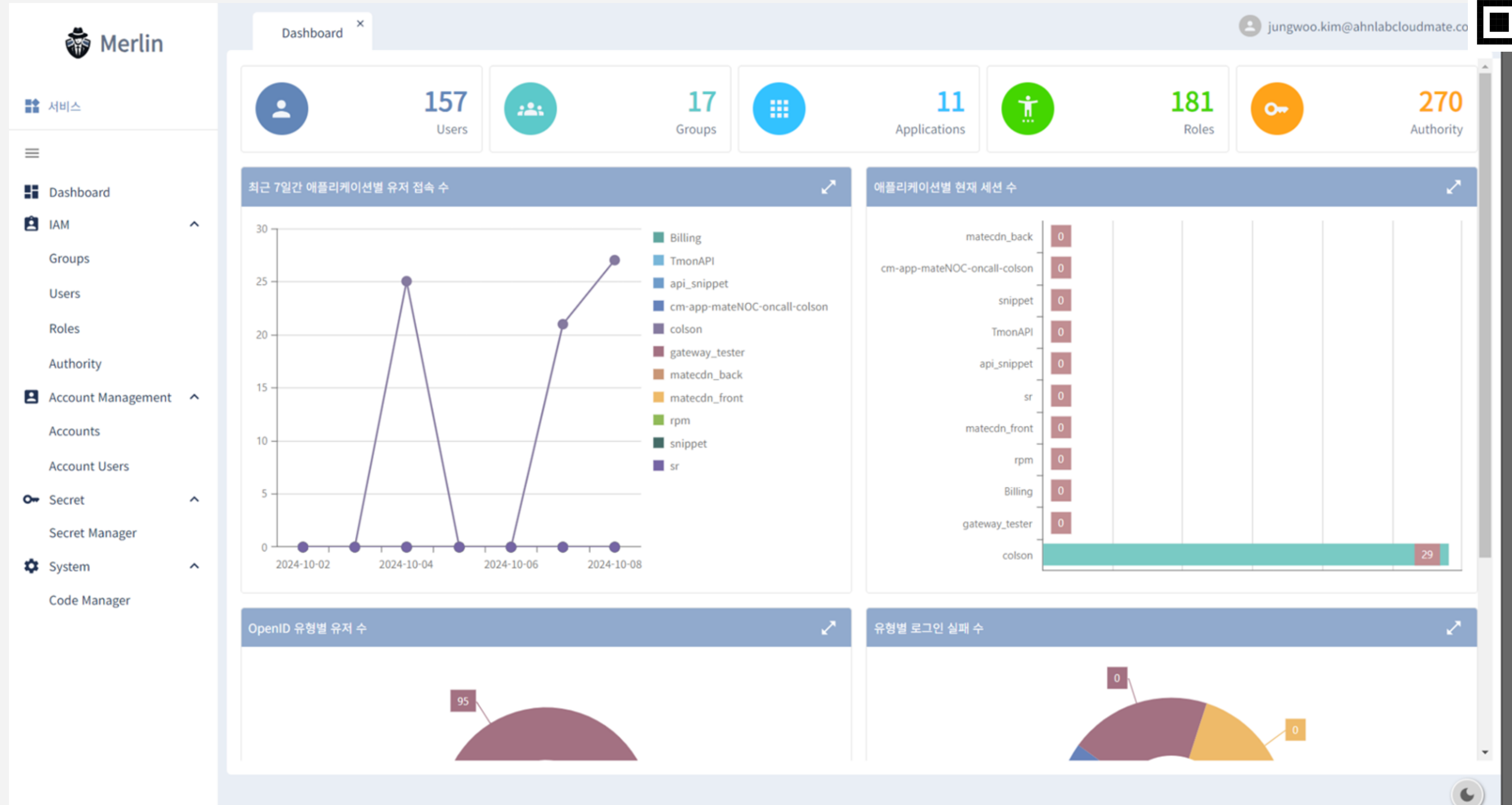
Merlin은 각 유저에게 RBAC를 제공하고 Secret 관리 서비스를 제공합니다.



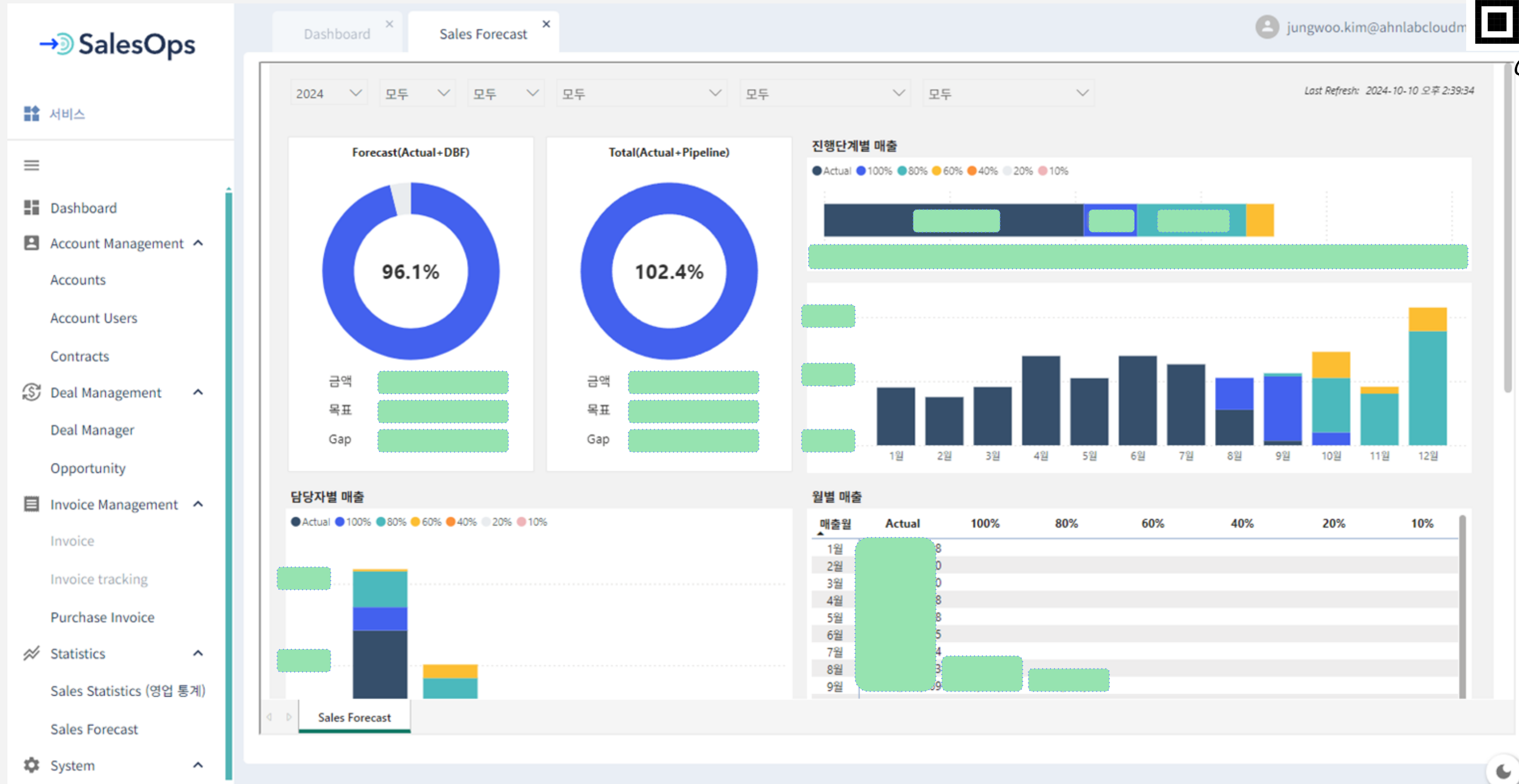
Merlin v1.0 - 계정 권한 관리 서비스



Question!



SalesOps - 경영정보관리(MIS) 서비스



Question!

SalesOps - 경영정보관리(MIS) 서비스



Question!

SalesOps

서비스

Dashboard

Account Management

Accounts

Account Users

Contracts

Deal Management

Deal Manager

Opportunity

Invoice Management

Invoice

Invoice tracking

Purchase Invoice

Statistics

Sales Statistics (영업 통계)

Sales Forecast

System

Code Manager

Dashboard

Sales Statistics (영업 통계)

Drop Filter Fields Here

Total

Margin

Tran Date Year

Tran Date Quarter

Tran Date Month

2024

Q3

September

Q3 Total

Q4

October

Q4 Total

2024 Total

Account Name

Total

Margin

Total

Margin

Total

Margin

Total

Margin

Total

Margin

2,964,610.00

1,732,953.00

8,250,000.00

289,300.00

27,107,571.00

2,225,887.00

2,757,150.00

1,882.00

287,041.00

7,921,049.00

24,979,829.00

13,971,793.00

2,770,845.00

5,500,000.00

810,918.00

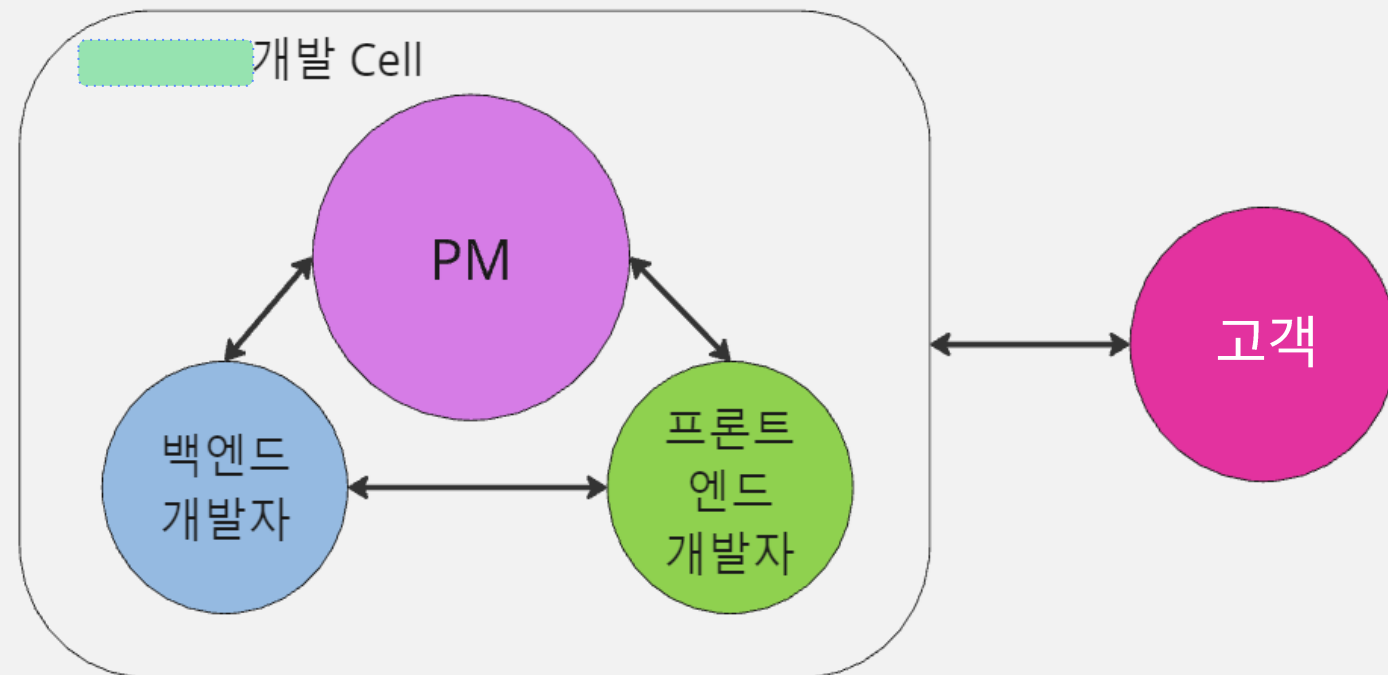
다들 힘들어할 때 성과를 끌어올리기 위한 노력들



Question!

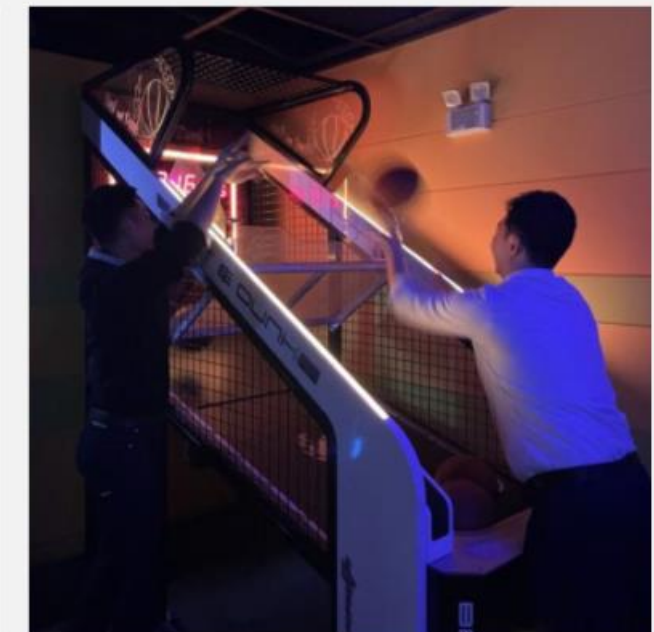
1. 개발 셀은 신규 PM을 중심으로 강력한 협업체제 구축이 필요합니다.

목표: 고객의 변칙적인 **요구를 잘 해석**하고, 이것을 **코드로 산출**해내는 것
커뮤니케이션의 총량이 순수 **개발의 양**(loc) 보다 더 중요한 케이스



2. 기여하고 성취의 기쁨을 만끽하자. 내 것으로 만들자.

우리는 여행도 가야 되고 차도 사야 하고, 망량이 부양도 해야 해서
어차피 노동을 해야 합니다. 노동은 신성한 것. 노동을 즐겨봅시다.

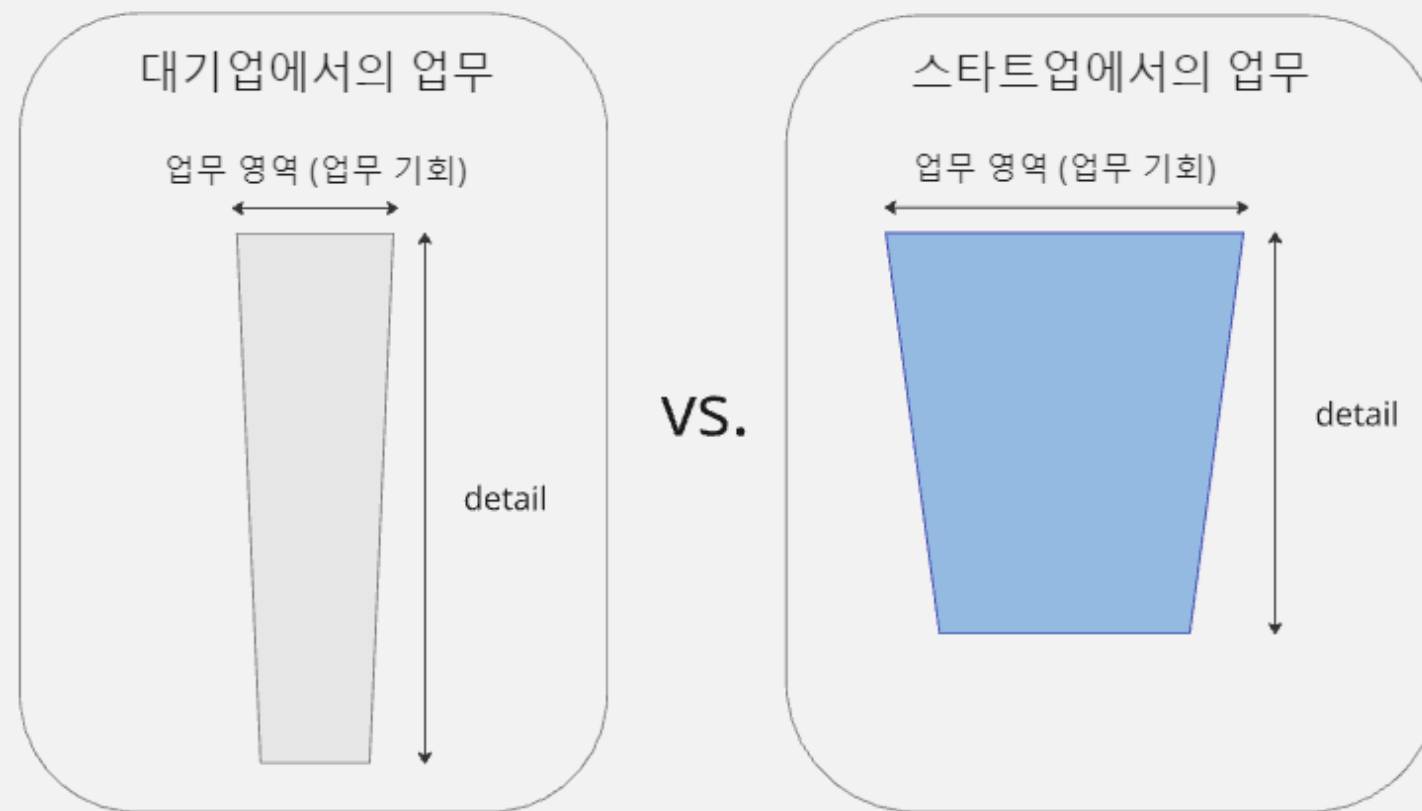


다들 힘들어할 때 성과를 끌어올리기 위한 노력들

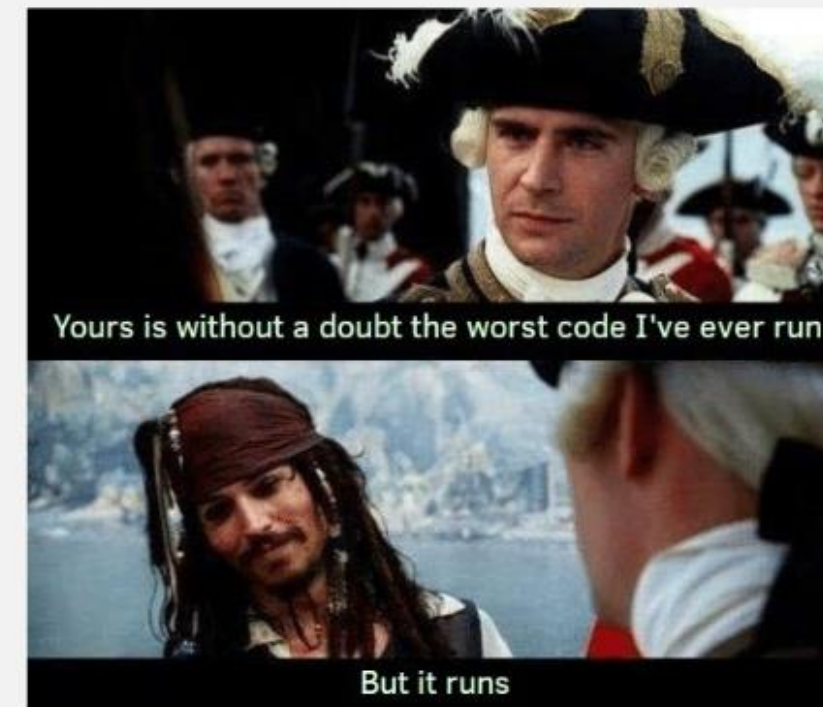


Question!

3. 대기업에서의 업무 vs. 스타트업에서의 업무



4. 잘 작성된 문서 보다, 작동하는 코드가 더 중요하다. (적어도 아직까지는)



다들 힘들어할 때 성과를 끌어올리기 위한 노력들



Question!

5. 커뮤니케이션의 대역폭 (bandwidth)

대면 대화 > 비디오 콜 > 보이스 콜 >>>> 채팅 > 이메일 > 봉화(??)

<- 대역폭 큼

대역폭 작은 ->



6. 여러분들의 목소리에 귀를 기울이겠습니다.

우리는 이익을 내기 위해 모인 사람들입니다.

성과에 방해되는 모든 것을 제보해주세요.

모든 일은 다 지나갑니다. 커리어 성장 관점에서 지속적으로 새로운 도전 및 성취를 할 수 있도록 업무 분배를 하겠습니다.



🐱 문제 해결의 3요소



Zero to One



Question!

- 첫번째 프로덕트를 출시하면서 대단한 크런치모드를 경험
- Zero에서 여기까지 일궈냈다는 성취감
- 집단적 자아효능감 상승
- 이제 앞으로 더 민첩하게 프로덕트를 찍어낼 수 있을거란 자신감

그래서 맛있는걸 먹으러 갔습니다.



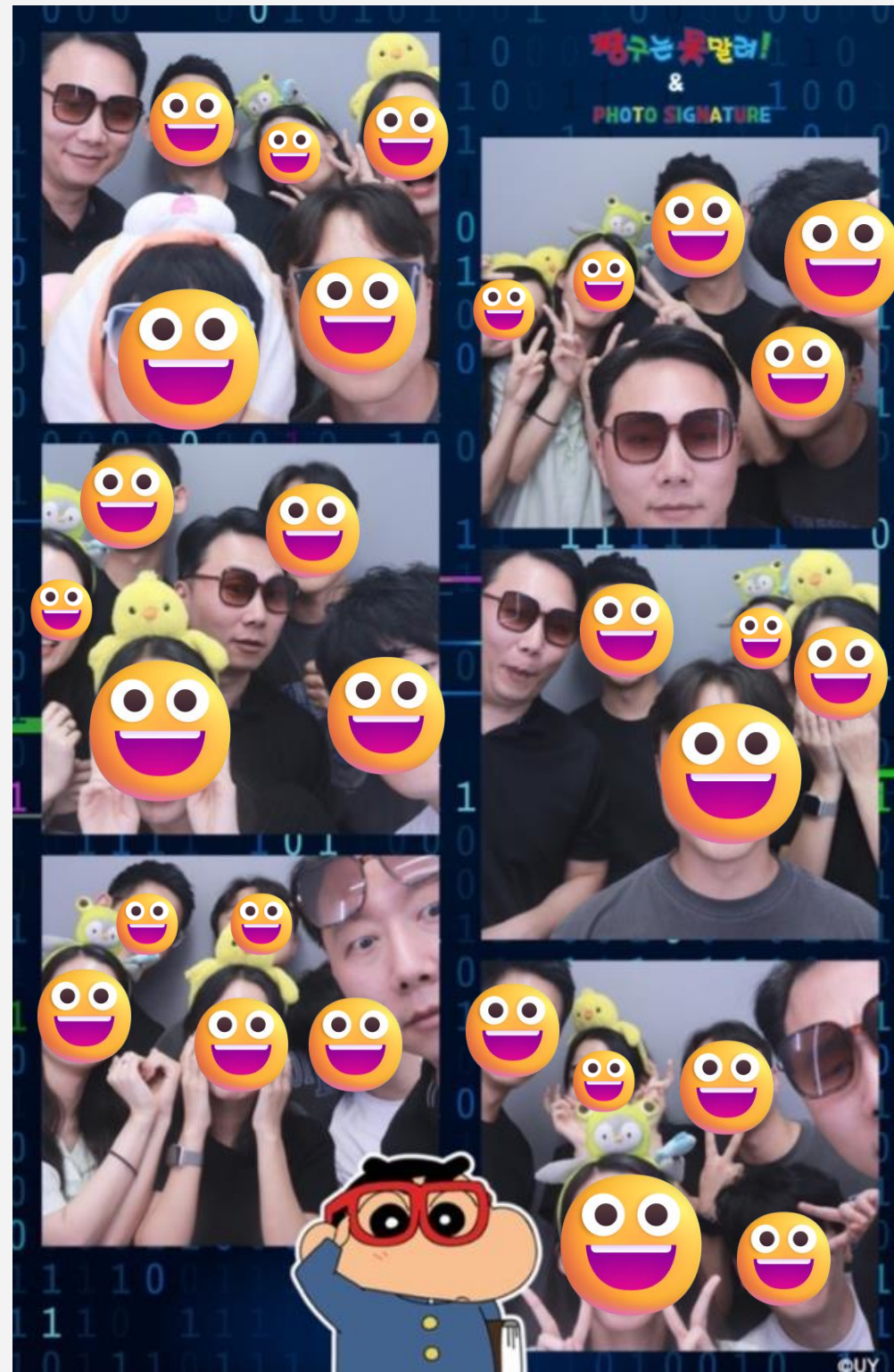
Question!



그래서 맛있는걸 먹으러 갔습니다. (2)



Question!



그래서 맛있는걸 먹으러 갔습니다. (3)



Question!



SPoF 제거

고객의 긍정적인 피드백

확장 가능한 아키텍처

05

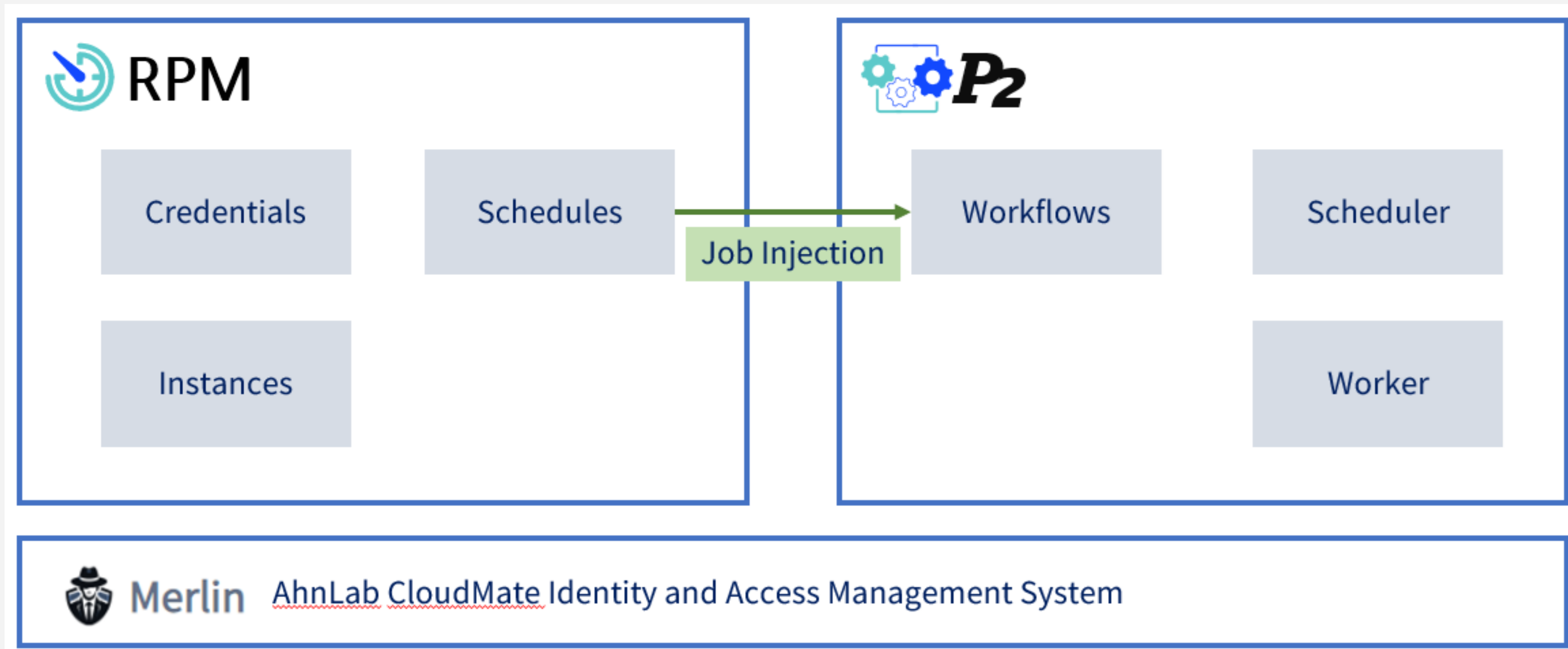
새로운 프로덕트 개발 및 확장

지속적인 DAU, MAU 증가

새로운 프로덕트 개발



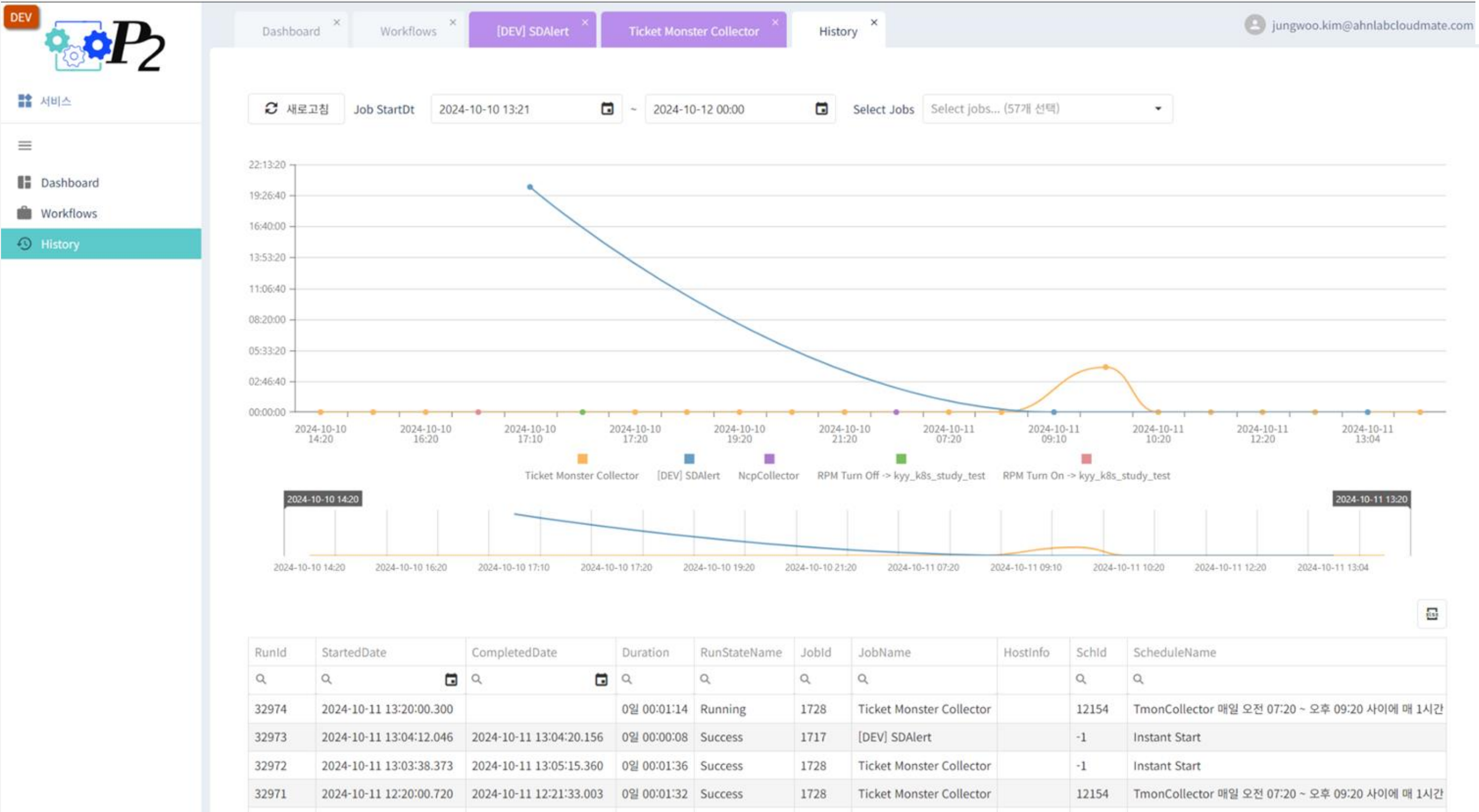
Question!



새로운 프로덕트 개발



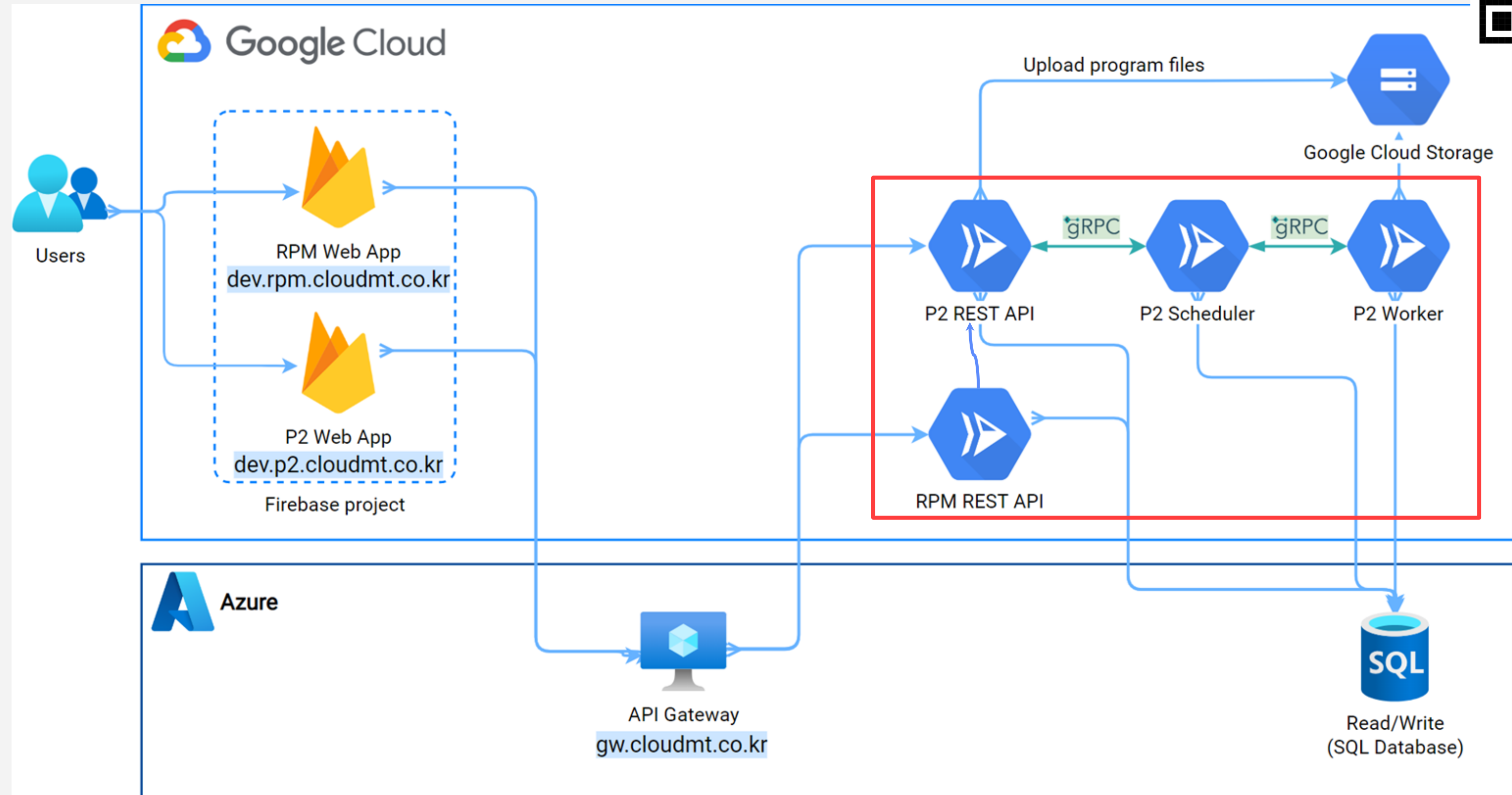
Question!



Infra Structure - Google Cloud



Question!



서비스 규모의 성장



Question!



User count

1,500% 증가

MAU

755.6% 증가

👉 서비스 개시 2022년 대비

서비스 규모의 성장



Question!

 SDP 멀티 클라우드 환경을 위한 보안 진단 플랫폼	 RPM 클라우드 리소스 파워 관리자	 Mate365 365일 기술지원 서비스	 Billing 클라우드 사용량 및 비용 관리 포털
 MateNOC 클라우드 장애 알림 서비스	 P2 경량 작업 스케줄러	 IAM 보안을 위한 계정 관리 SaaS	 SalesOps 최적의 세일즈 관리 SaaS
 SmartReport 빠르고 정확한 리포트 생성 서비스	 CacheCat 간편한 캐시 엔진 솔루션	 MateCDN CDN 통계 서비스	 Snippet 코드 공유 플랫폼

Infra Structure 개선



Question!

목표

- 늘어난 트래픽에 대응할 수 있도록 개선 필요

개선 사항

1. **내결함성**(Fault Tolerance) 및 **고가용성**(High Availability) 개선

👉 시스템이 장애를 감지하고 복구할 수 있는 능력

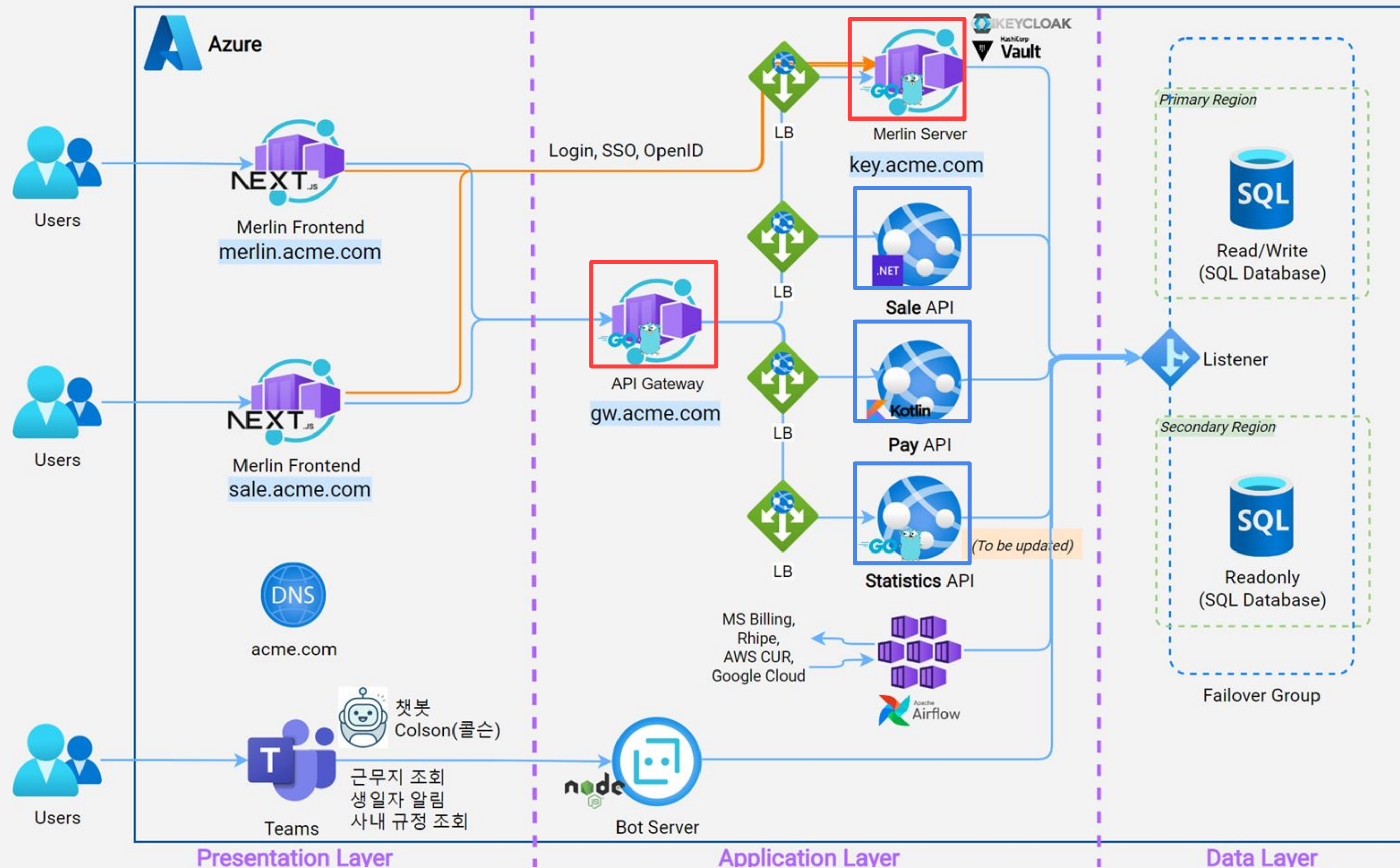
👉 시스템이 항상 사용 가능한 상태를 유지하는 능력

2. **단일 장애 지점**(SPoF, Single Point of Failure) 제거

3. **관측성**(Observability) 개선

👉 시스템의 출력으로 상태를 파악할 수 있는 정도

Infra Structure 개선



Question!

OpenTelemetry

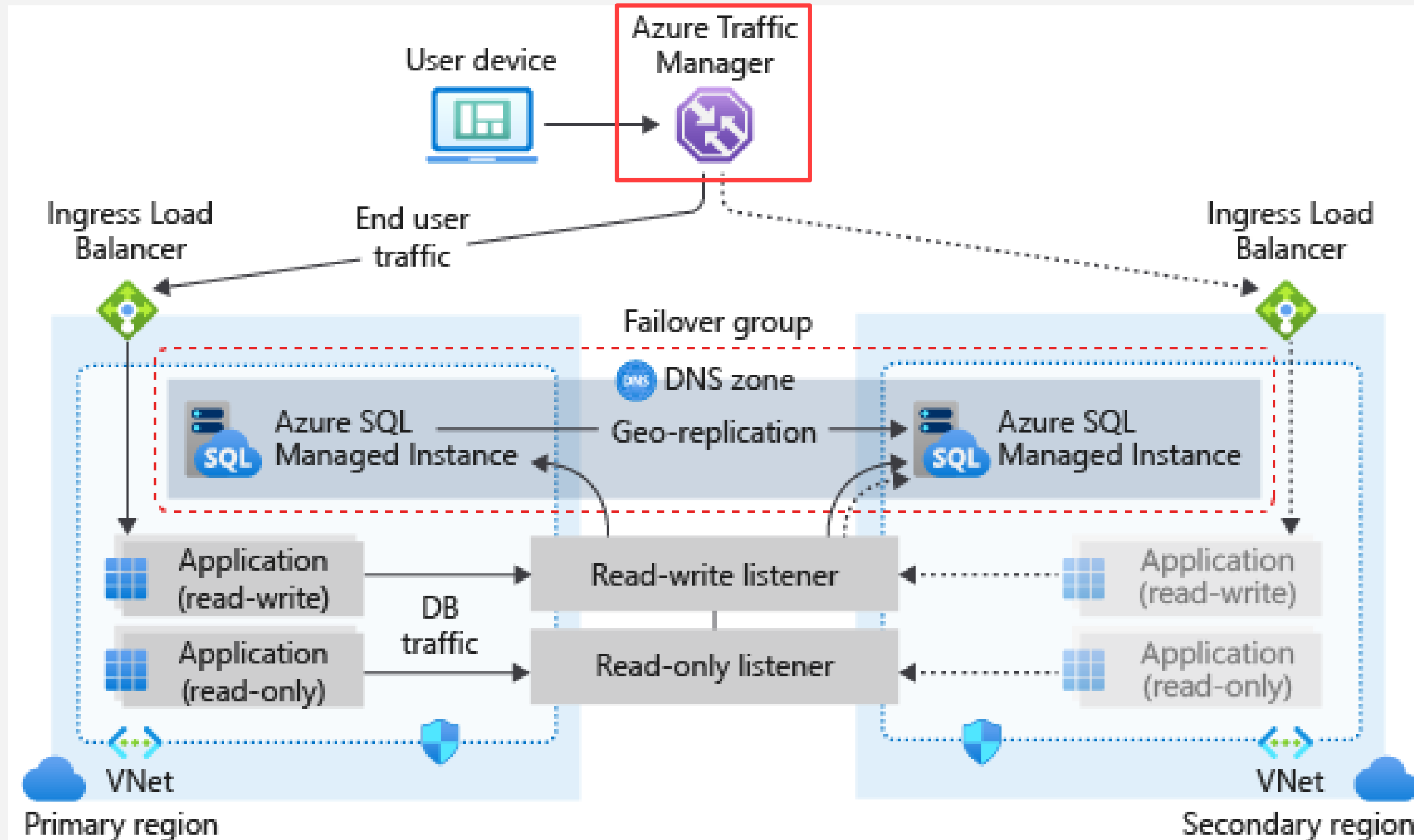
Grafana loki

Grafana

Infra Structure 개선 - DB 부분



Question!



클라우드 관리 플랫폼

개발 조직을 스쿼드 형태로!

앞으로 만들 프로젝트 구체화

06

앞으로의 여정

지금도 바쁘고 앞으로도 바쁘다

안랩클라우드메이트 통합법인 출범

AhnLab

회사소개

지속가능경영

투자정보

News Room

인재채용

Search



Question!

안랩, 클라우드 운영관리 서비스(MSP)전문 통합법인 '안랩클라우드메이트' 출범

2024.07.01

안랩, 클라우드 운영관리 서비스(MSP)전문 통합법인 '안랩클라우드메이트' 출범

- 7월 1일(월), 안랩의 클라우드 네이티브 전문 자회사 '클라우드메이트'와 자사의 클라우드서비스 사업조직 통합 완료, MSP 전문 통합법인 '안랩클라우드메이트' 출범
- 안랩의 '보안 프레임워크 기반 클라우드 구축·운영' 역량에 클라우드메이트의 '클라우드 네이티브' 전문성 결합, 클라우드 보안부터 활용성까지 아우르는 기술 경쟁력 강화 및 사업 기회 발굴 예정

안랩(대표 강석균, www.ahnlab.com)이 7월 1일(월), 클라우드 네이티브 전문 자회사 '클라우드메이트'와 자사의 클라우드서비스 사업조직 통합을 완료하고, 차세대 클라우드 운영관리 서비스(MSP) 전문 통합법인 '안랩클라우드메이트(대표 김형준, 고창규)'를 공식 출범했다고 밝혔다.

이번 통합법인 출범은 빠르게 성장중인 클라우드 시장에서 차세대 MSP로써 경쟁력을 강화하고 사업을 확장하기 위해 이루어졌다. 안랩클라우드메이트는 안랩의 '보안 프레임워크 기반 클라우드 구축·운영' 역량과 클라우드메이트의 '클라우드 네이티브' 전문성을 결합해, MSP 영역에서 클라우드 보안부터 활용성까지 아우르는 기술 경쟁력을 강화하고 사업 기회를 적극적으로 발굴해 나갈 계획이다.

양사의 조직 통합은 안랩이 클라우드메이트에 자사의 MSP를 비롯한 클라우드서비스 관련 사업을 이관(영업양수도)하는 방식으로 이루어졌다. '안랩클라우드메이트'의 대표는 김형준 대표(前 안랩 서비스사업부문장)와 고창규 대표(前 클라우드메이트 대표)가 공동으로 맡는다.

안랩 강석균 대표는 "양사가 보유한 역량을 합치면 강력한 시너지를 낼 수 있을 것이라는 판단 하에 이번 통합을 결정하게 됐다"며, "안랩클라우드메이트가 앞으로 고객에게 '안전한 클라우드네이티브'라는 차별화된 가치를 제공할 수 있는 차세대 MSP 기업으로 시장에 자리 잡을 것으로 기대한다"고 말했다.

안랩클라우드메이트 고창규 대표는 "최근 서비스 적시성(Time-to-Market)과 생성형 AI가 부각되며, 클라우드 네이티브에 기반한 기민한 대응과 정보보호 모두가 중요해졌다"며, "이번 통합으로 더 많은 고객들이 안전한 클라우드를 클라우드답게 활용할 수 있도록 지원할 것"이라고 말했다.

안랩클라우드메이트 김형준 대표는 "이번 통합은 안랩의 보안 강화 MSP 역량과 클라우드메이트의 '클라우드 네이티브' 전문성이 더해져 시너지를 낸다는데 의미가 있다"며, "이를 기반으로 다양한 고객사별 최적화된 클라우드 환경을 제공해 사업 경쟁력을 높여나갈 계획"이라고 말했다.

[보충자료] 안랩의 클라우드 관리 서비스(MSP) 사업 주요 연혁

- 2021년 2월 보안 특화 클라우드 관리 서비스(MSP) '안랩 클라우드(AhnLab Cloud)' 출시
- 2023년 9월 클라우드 네이티브 전문기업 클라우드메이트와 전략적 제휴 협약(MOU) 체결
- 2024년 4월 클라우드 네이티브 전문기업 클라우드메이트 인수
- 2024년 7월 통합법인 '안랩클라우드메이트' 출범

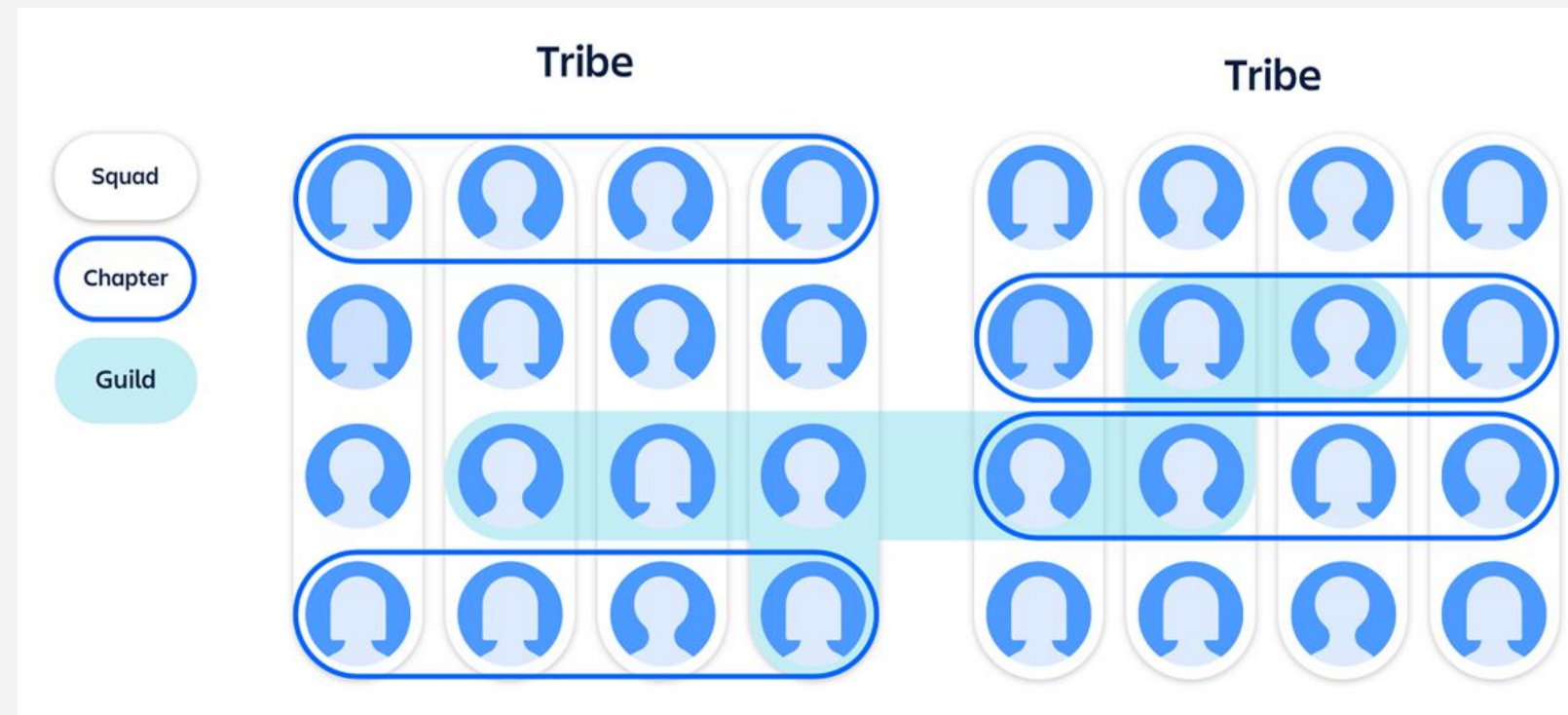
https://company.ahnlab.com/kr/news/press_release_view.do?seqPressRelease=8706

앞으로도 할 일들이 너무나 많습니다.



Question!

1. 현재는 NCP 연동 개발 중
2. 멀티 클라우드를 아우르는 **클라우드 관리 플랫폼**(CMP) 개발 예정
3. 개발 조직을 연구소로 설립
4. 조직을 Tribe, Chapter, Squad로 구성하여 여러 프로젝트를 동시에 진행



We're Hiring! 🗣️



Question!

클라우드 엔지니어 분들의 많은 관심과 지원 부탁드립니다.

<https://ahnlabcloudmate.career.greetinghr.com>

- Technical Engineer
- SaaS(네이버웍스, M365, GWS) 기술 지원 엔지니어



Thank You



Jungwoo (Jayden) Kim

AhnLab CloudMate

Dev 1 Team Lead

