

LiveOps : 네이버앱의 실시간 운영과 크래시 핸들링 솔루션

이건준
NAVER



NAVER DEVVIEW 2023

0. 발표자 소개

About me

Career

- 2017.9 ~ : NAVER 앱 개발
- 2019.10 : 사내 개발자 컨퍼런스 참여 (Engineering Day)
- 2021.11 : 사내 오픈소스 페스티벌 참여
- 2021.11 : 사내 개발자 컨퍼런스 참여 (N INNOVATION)
- 2022.10 : 사내 개발자 컨퍼런스 참여 (Engineering Day)

Public speaking

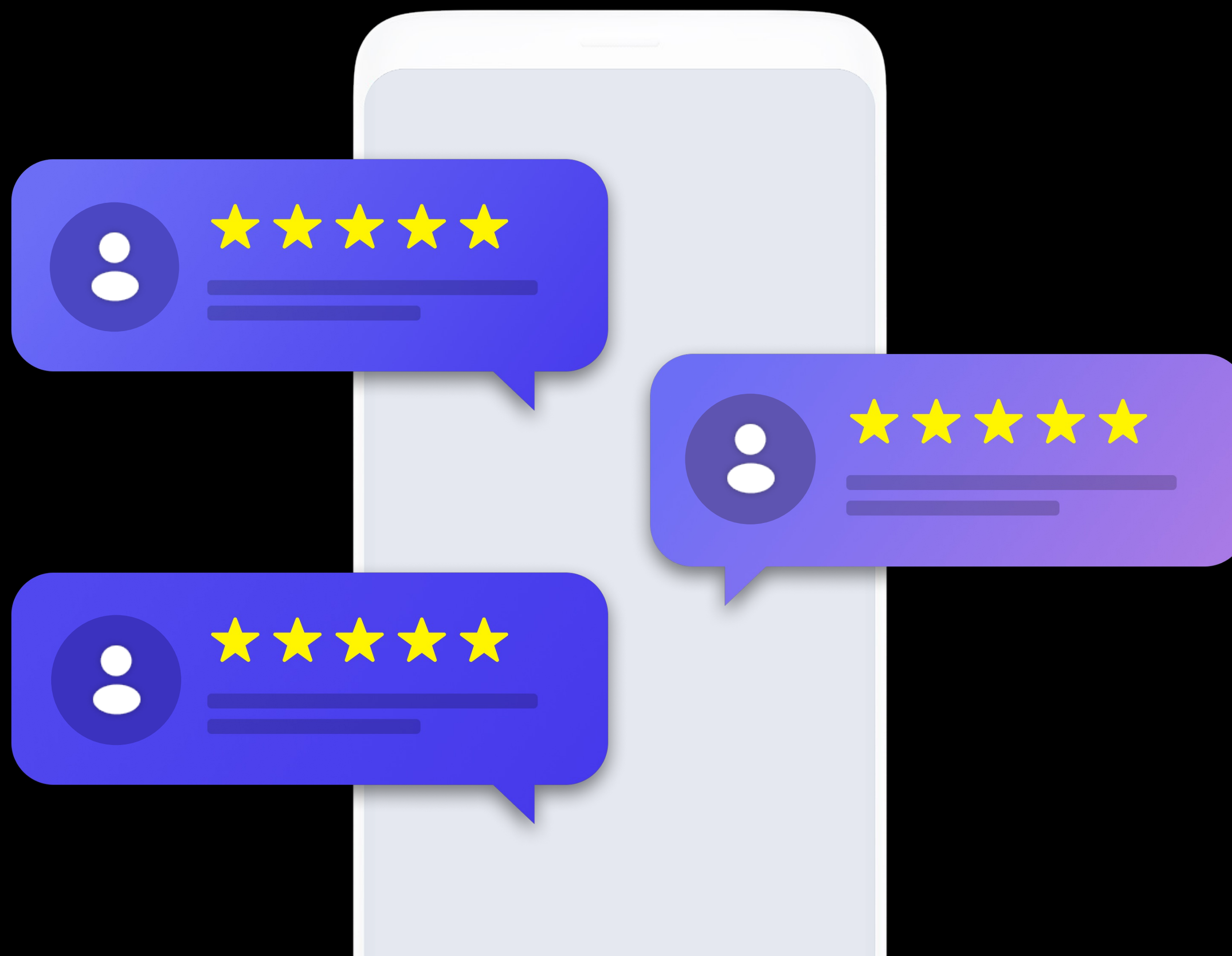
- 전무

Contents

1. LiveOps 도입 배경
2. LiveOps 소개
3. LiveOps 플로우 및 구조
4. LiveOps 사용 사례
5. LiveOps 도입 효과

1. LiveOps 도입 배경

1.1 모바일 앱 서비스의 성공

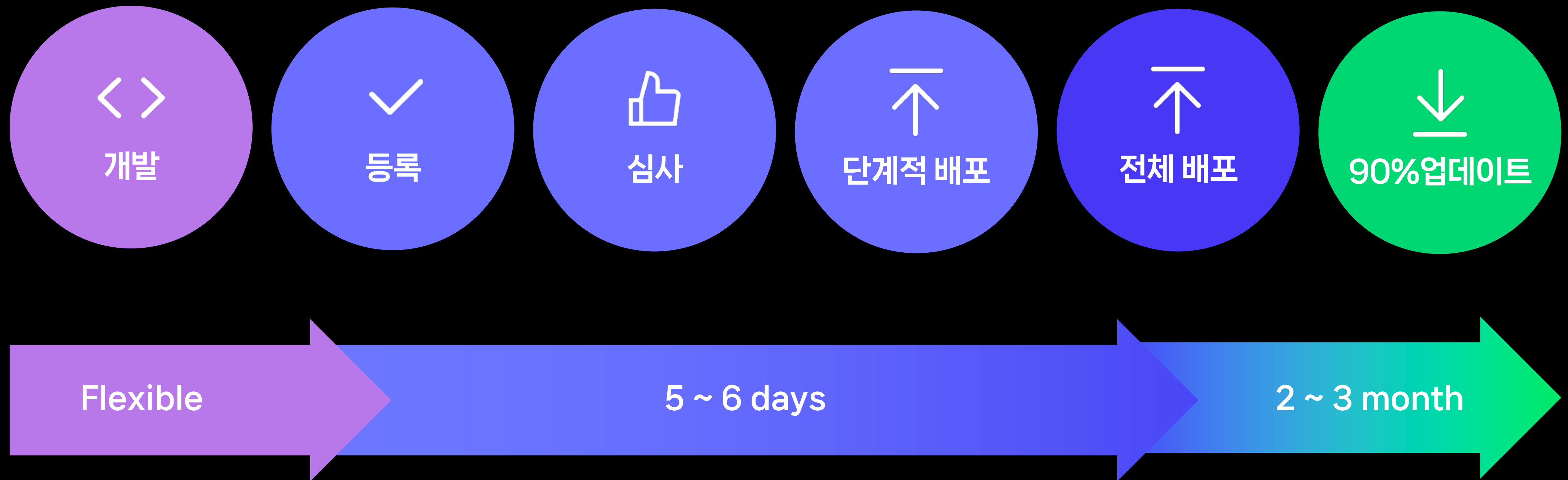


1.1 모바일 앱 서비스의 성공



1.2 모바일 플랫폼의 한계

모바일 플랫폼 운영 방식



1.3 기존 대응 방식의 한계

기존 운영 정책 및 크래시 대응 방식

운영 정책 방식

- Lead time 개선
- 강제 업데이트

장애 대응 방식

- 예외 사항 처리
- Test code 작성

1.3 기존 대응 방식의 한계

기존 운영 정책 및 크래시 대응 방식

운영 정책 방식

- Lead time 개선
- 강제 업데이트

장애 대응 방식

- 예외 사항 처리
- Test code 작성

1.3 기존 대응 방식의 한계

기존 운영 정책 및 크래시 대응 방식

운영 정책 방식

- Lead time 개선 ✕
- 강제 업데이트 ✕

장애 대응 방식

- 예외 사항 처리
- Test code 작성

1.3 기존 대응 방식의 한계

기존 운영 정책 및 크래시 대응 방식

운영 정책 방식

- Lead time 개선 ✕
- 강제 업데이트 ✕

장애 대응 방식

- 예외 사항 처리 ✕
- Test code 작성 ✕

1.3 기존 대응 방식의 한계

모바일 플랫폼 및 기존 대응 방식의 한계

모바일 플랫폼

- 코드 수정 불가 ✕
- 긴 배포 프로세스 ✕

운영 정책 방식

- Lead time 개선 ✕
- 강제 업데이트 ✕

장애 대응 방식

- 예외 사항 처리 ✕
- Test code 작성 ✕

1.4 네이버앱에서의 의미

네이버앱 운영과 크래시의 파급력



1.5 LiveOps 가 추구하는 가치

네이버앱의 철학과 이를 기반으로 생성된 LiveOps 솔루션

- 더 좋은 서비스를 빠르게 알리고 delivery 할 수 있도록 한다.
- 보다 안정적이고 최적화된 서비스를 운영할 수 있도록 한다.
- 사용자를 더욱 이해하고 개별 요구 사항을 충족할 수 있도록 한다.

2. LiveOps 소개

2.1 LiveOps의 의미

LiveOps란?

- Live Operations의 줄임말로 보통 새로운 버전을 출시하지 않고 변경 사항 또는 향상된 기능을 적용하는 것

2.1 LiveOps의 의미

LiveOps란?

- Live Operations의 줄임말로 보통 새로운 버전을 출시하지 않고 변경 사항 또는 향상된 기능을 적용하는 것

게임에서의 LiveOps 사용 예시

- 업데이트 없이 새로운 캐릭터, 아이템 추가와 같이 콘텐츠 추가를 통한 흥미 유발
- 주말과 같은 특정 날짜에 2배 경험치 이벤트와 같은 이벤트를 통해 사용자 유입 유도

2.1 LiveOps의 의미

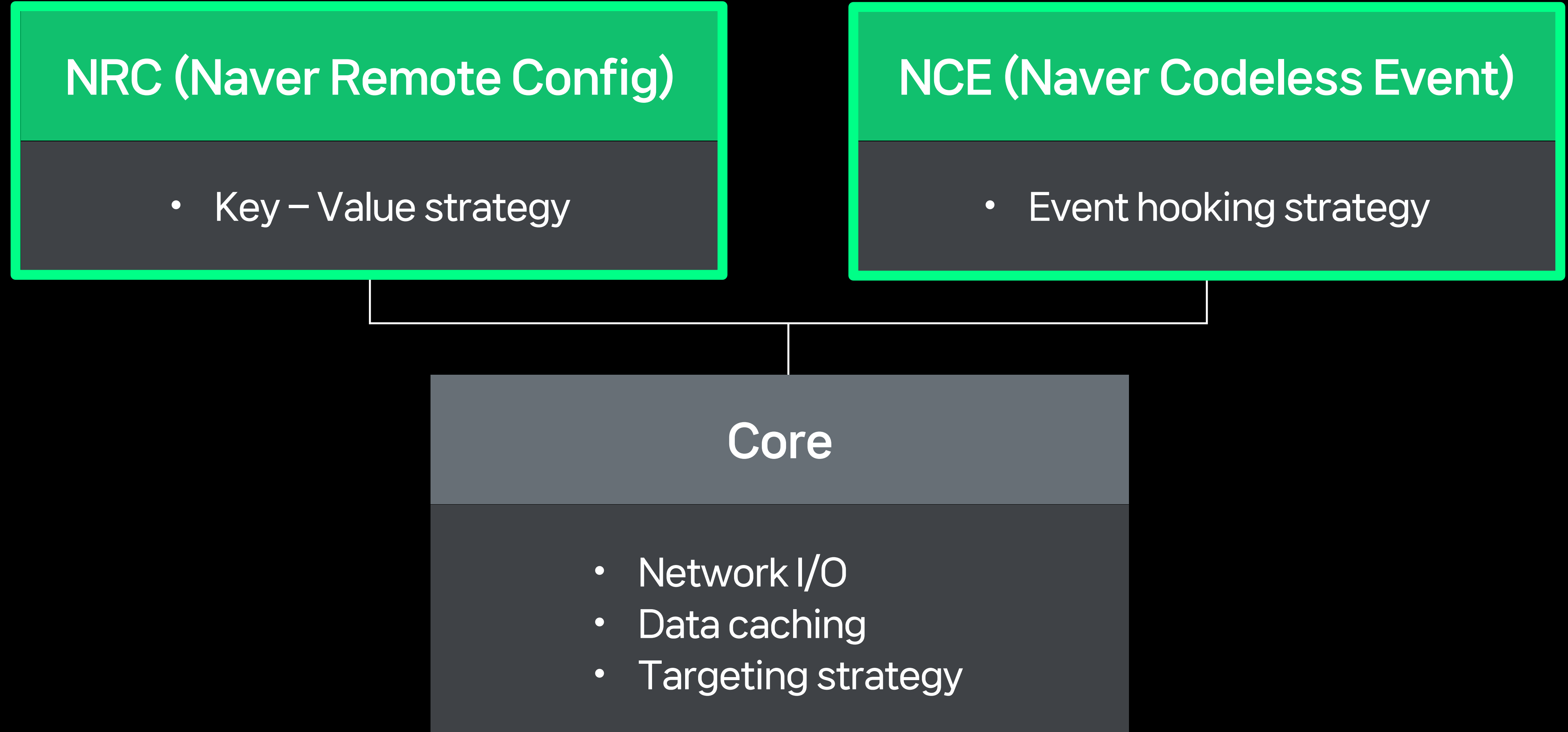
LiveOps란?

- Live Operations의 줄임말로 보통 새로운 버전을 출시하지 않고 변경 사항 또는 향상된 기능을 적용하는 것

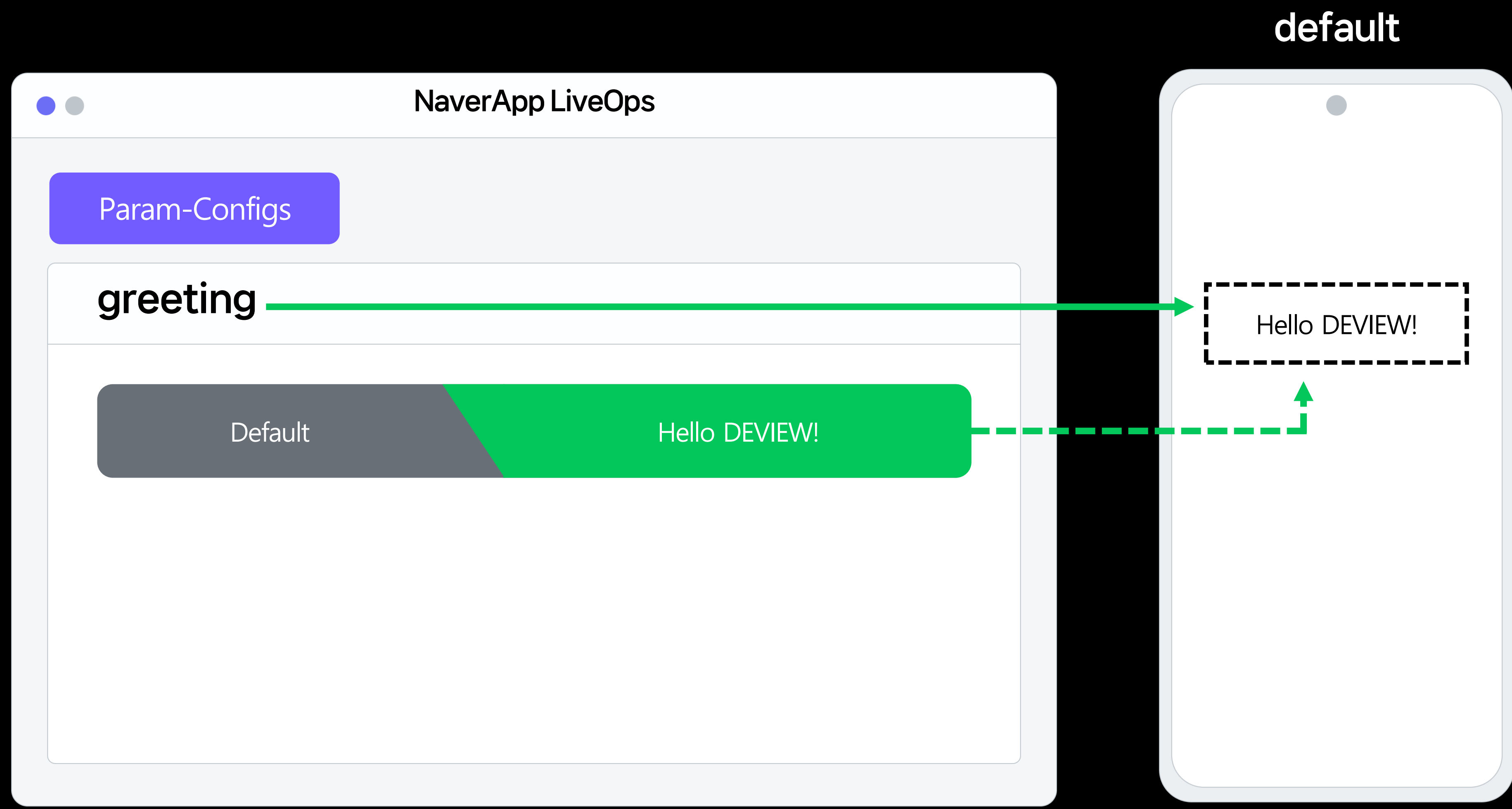
네이버앱에서의 LiveOps 사용 예시

- 업데이트 없이 특정 기능의 on/off 를 통한 운영
- 특정 날짜에 특정 타겟층에게만 이벤트성 툴팁, 배너를 노출
- 장애 상황에 접근을 시도할 경우, 사용자가 장애 화면에 도달하지 못 하도록 우회시키는 전략

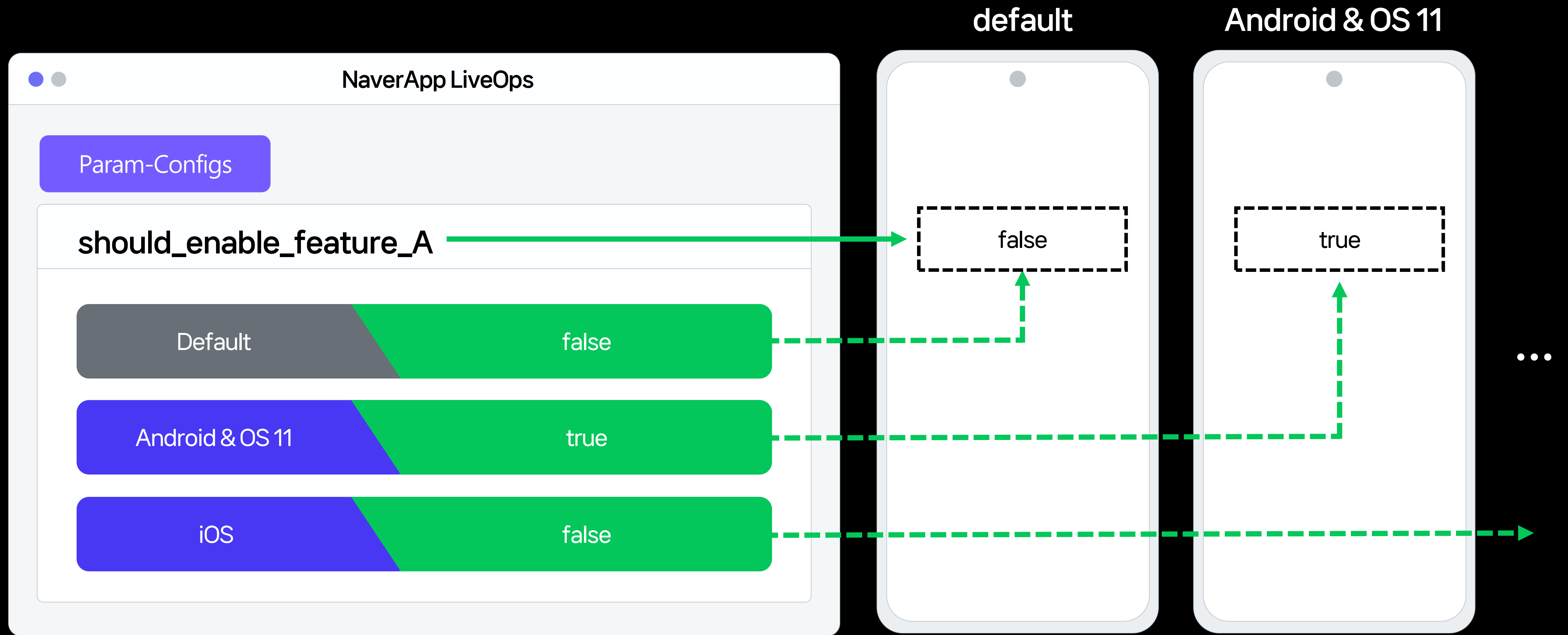
2.2 LiveOps의 구성



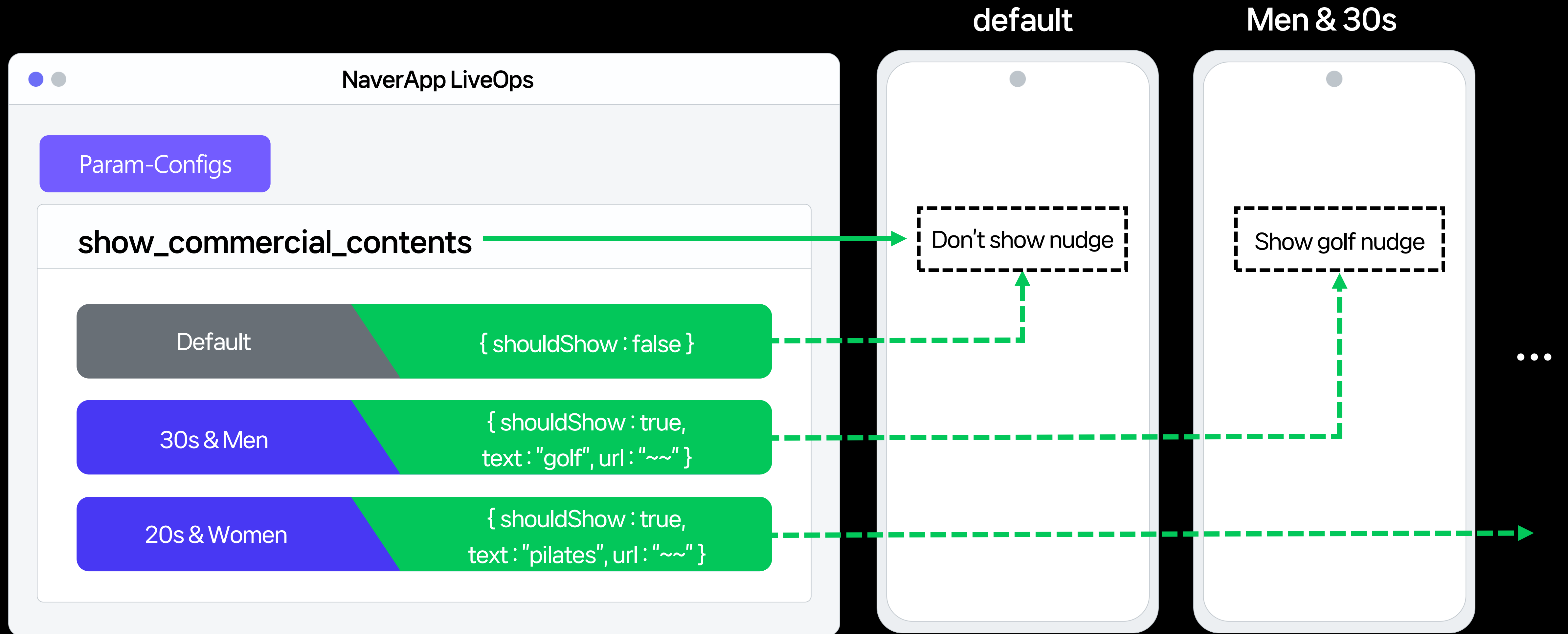
2.3 LiveOps 소개 - NRC (Naver Remote Config)



2.3 LiveOps 소개 – NRC (Naver Remote Config)



2.3 LiveOps 소개 - NRC (Naver Remote Config)



2.3 LiveOps 소개 – NRC (Naver Remote Config)

어디서 비슷한 기능을 가진 솔루션을 본 것 같은데?!

- Firebase remote config 에서 제공해주고 있는 기능과 비슷한 기능을 제공하는 NRC SDK
- LiveOps 에서는 remote config 기능을 한 단계 발전시킨 codeless event 기능도 제공해준다.
- NCE SDK 는 NRC SDK 와는 다르게 미리 앱에 코드를 심어줄 필요가 없다

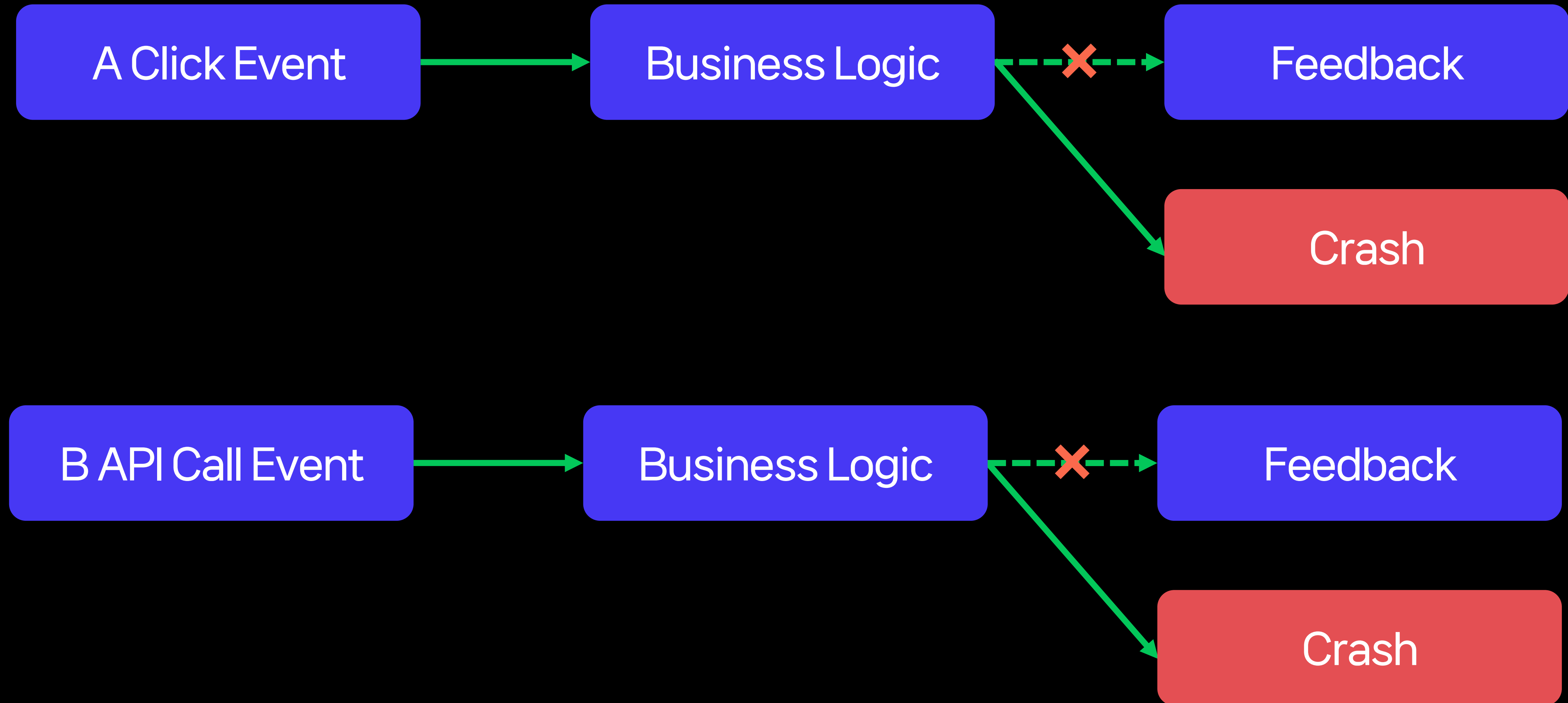
2.4 LiveOps 소개 – NCE (Naver Codeless Event)



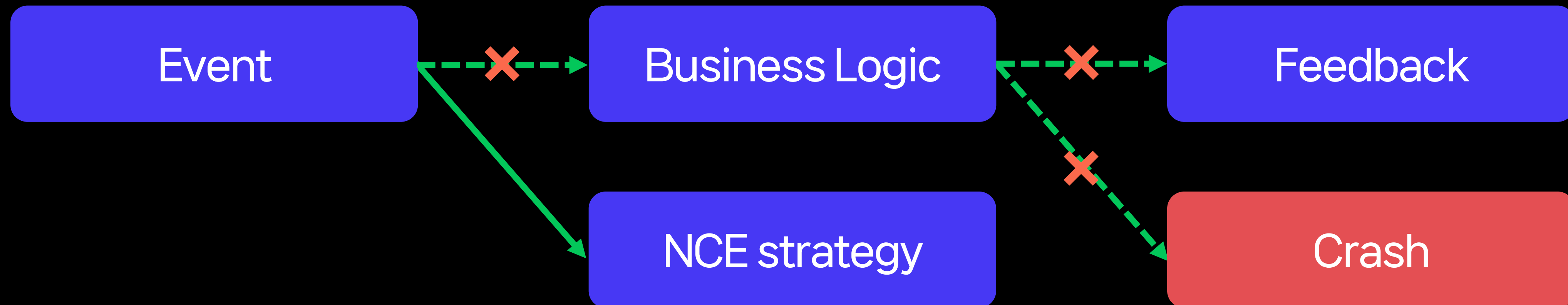
2.4 LiveOps 소개 – NCE (Naver Codeless Event)



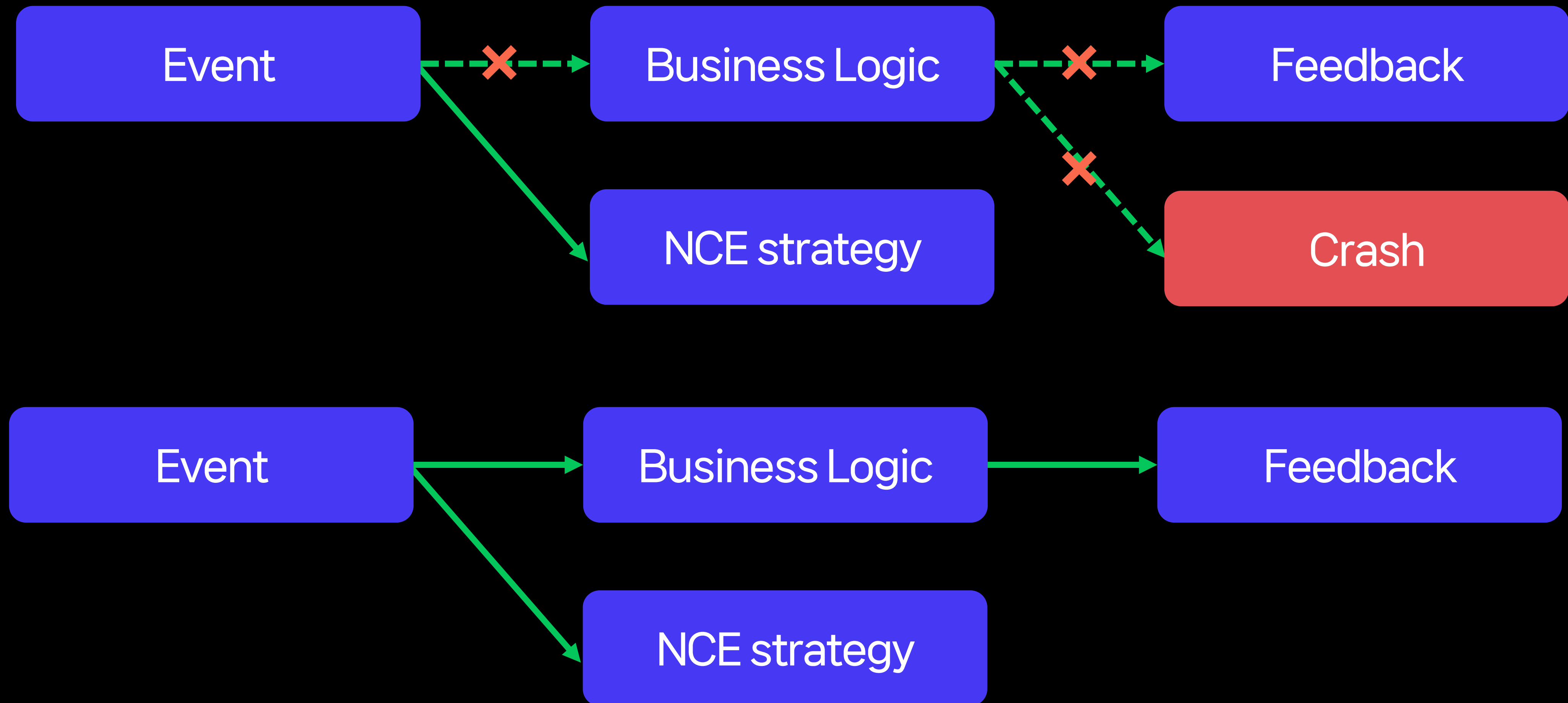
2.4 LiveOps 소개 - NCE (Naver Codeless Event)



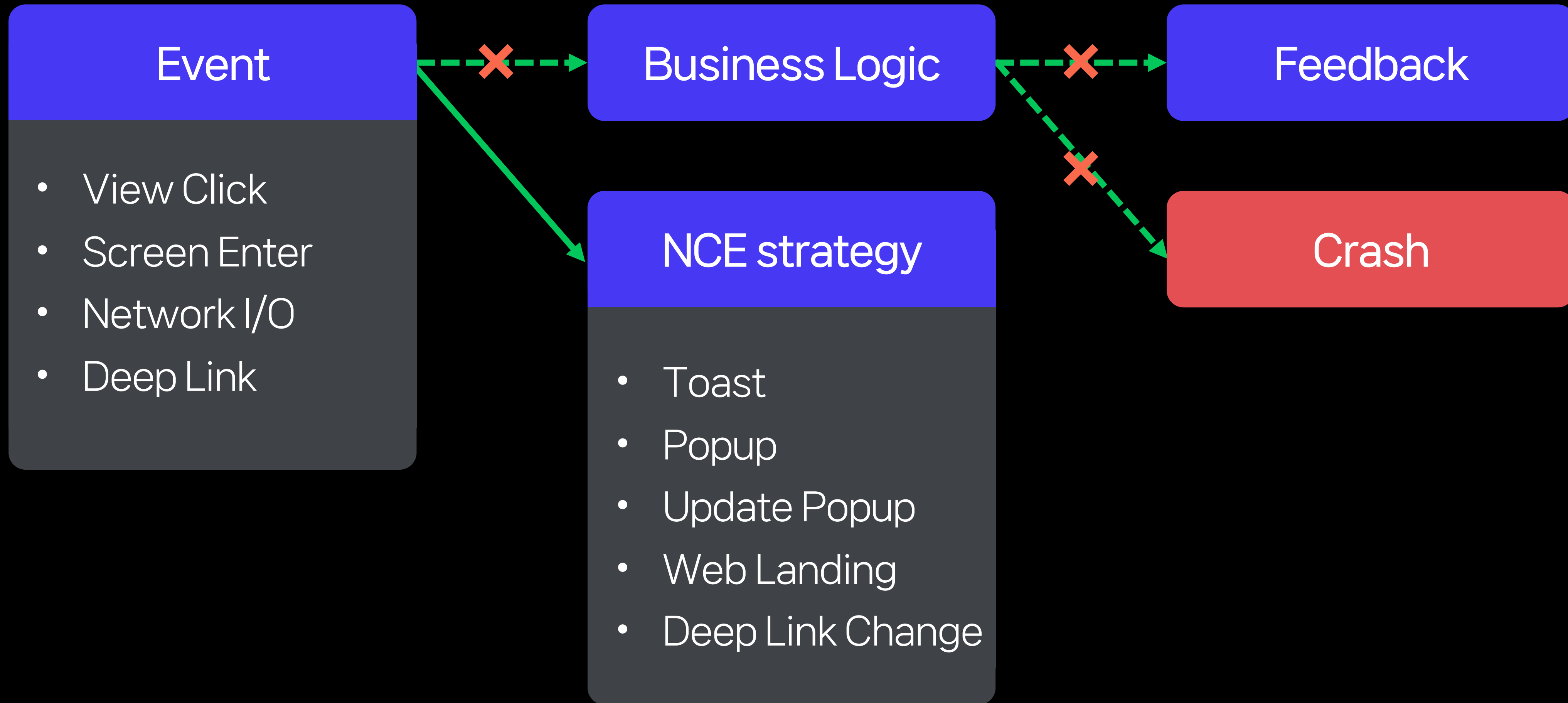
2.4 LiveOps 소개 – NCE (Naver Codeless Event)



2.4 LiveOps 소개 - NCE (Naver Codeless Event)



2.4 LiveOps 소개 – NCE (Naver Codeless Event)



2.4 LiveOps 소개 - NCE (Naver Codeless Event)



NaverApp LiveOps

Param-Events

A_button

event	click
strategy type	popup
strategy content	{ "content" : "api error!" }

2.4 LiveOps 소개 - NCE (Naver Codeless Event)



NaverApp LiveOps

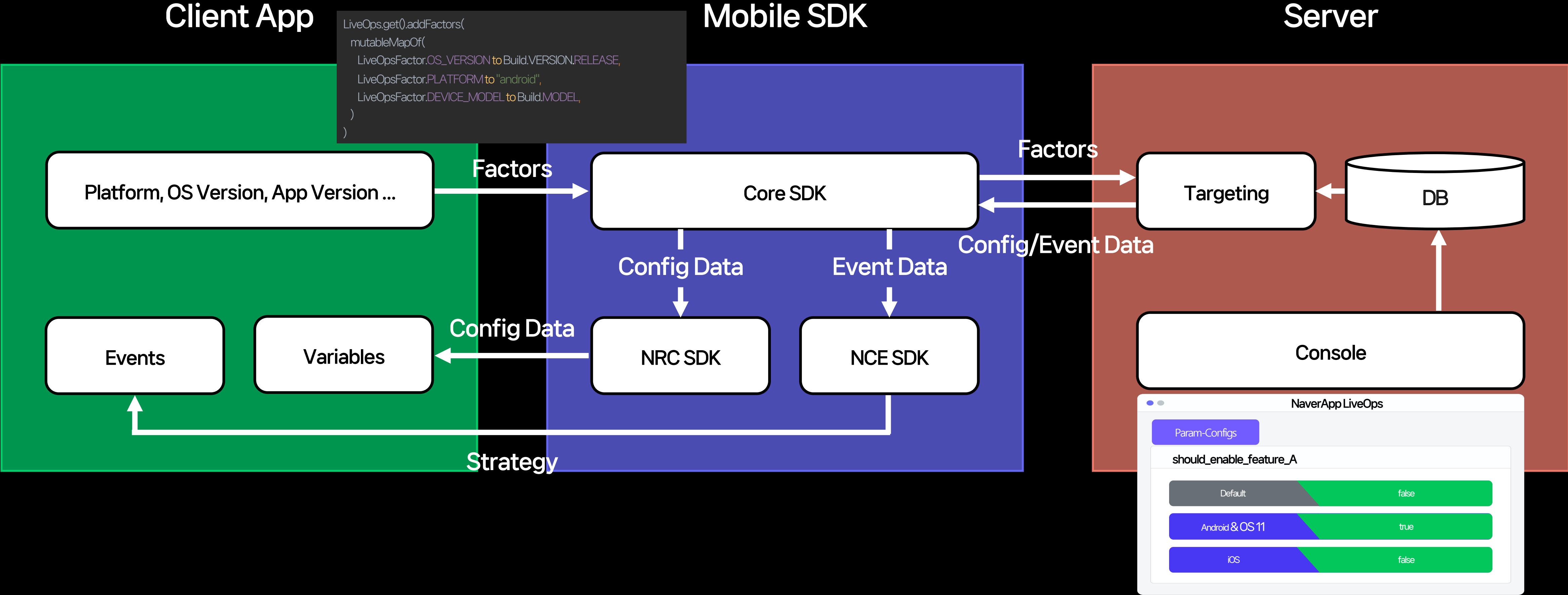
Param-Events

main_screen

event	screen_enter
strategy type	update_popup
strategy content	{ "content" : "new version is on market" }
condition	Less Than version 2.0.0

3. LiveOps 플로우 및 구조

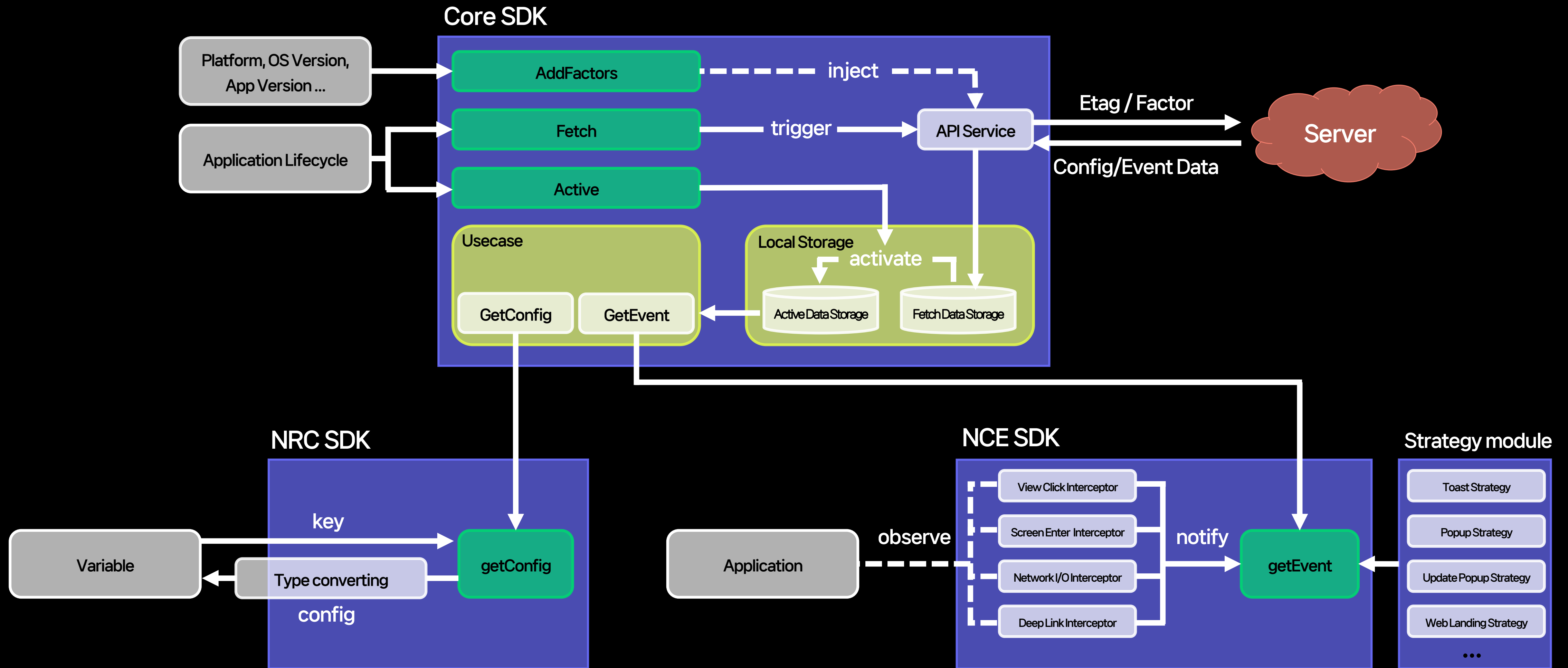
3.1 LiveOps 플로우



3.2 LiveOps 타킷팅 전략

	LiveOps Targeting
Factor	<ul style="list-style-type: none"> Platform, Device Model, OS version, App Version, Naver ID (hash), Date
Operator	<ul style="list-style-type: none"> Equal, Greater than, Greater equal, Less than, Less Equal
Condition	<ul style="list-style-type: none"> Factor 와 operator 를 조합하여 Condition 을 생성 <ul style="list-style-type: none"> ex 1) Platform Equal Android ex 2) OS version Greater equal 10
Rule	<ul style="list-style-type: none"> Condition 을 AND/OR 조건으로 연결할 수 있고 해당 Condition 들이 모두 만족할 때, 원하는 리턴값을 Rule 별로 지정 가능 Rule 간의 우선 순위 조정 가능 <ul style="list-style-type: none"> ex 1) (Platform Equal Android) AND (OS version Greater equal 10) 일 경우, true 를 리턴 ex 2) (Platform Equal Android) AND (Naver ID Equal "ABC") 일 경우, false 를 리턴

3.3 LiveOps 구조



3.4 Core 구조

Fetch / Active interface 제공

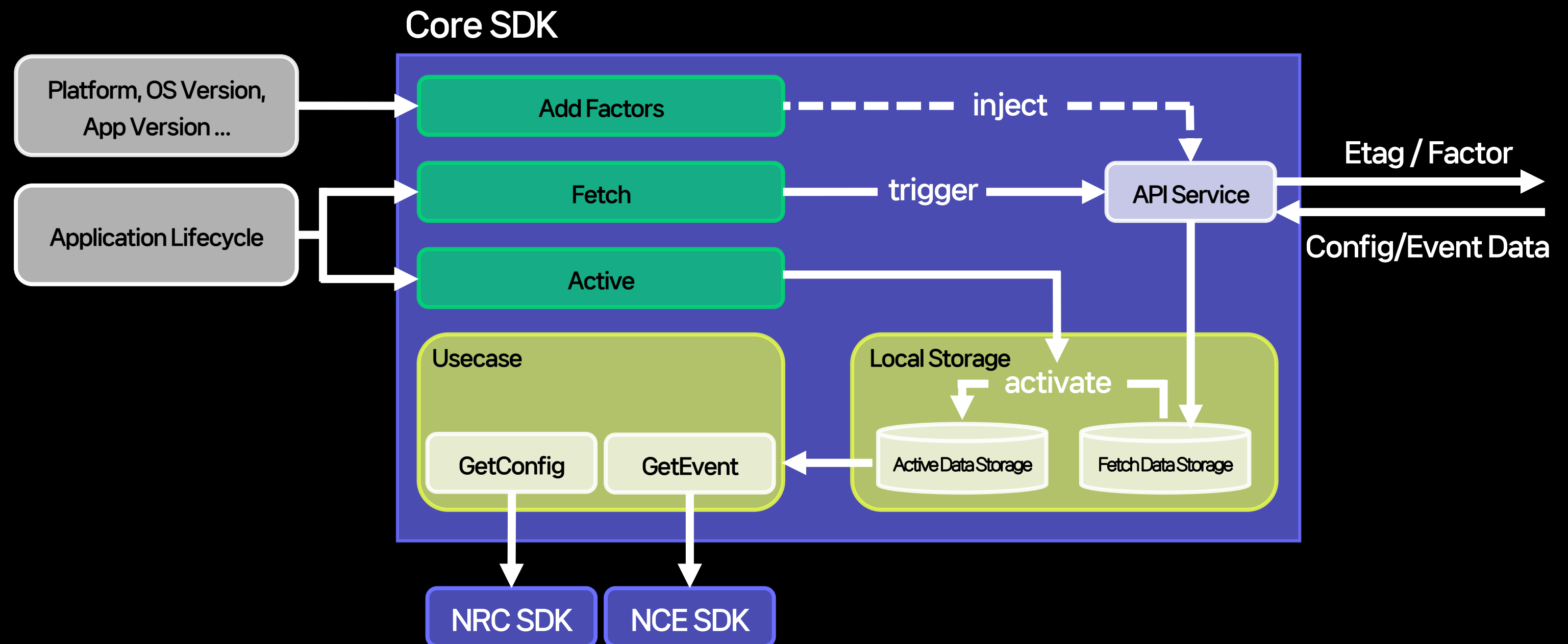
- Fetch/Active 함수를 통해 서버를 통해 data fetch 및 로컬 fetch/active 상태 관리

Data transmission

- 서버에서 console 상에 세팅된 값을 json 포맷으로 SDK 에 전달
- 서버 트래픽 비용 절감을 위해 ETAG 기반의 서버 캐시 운영

Targeting 전략

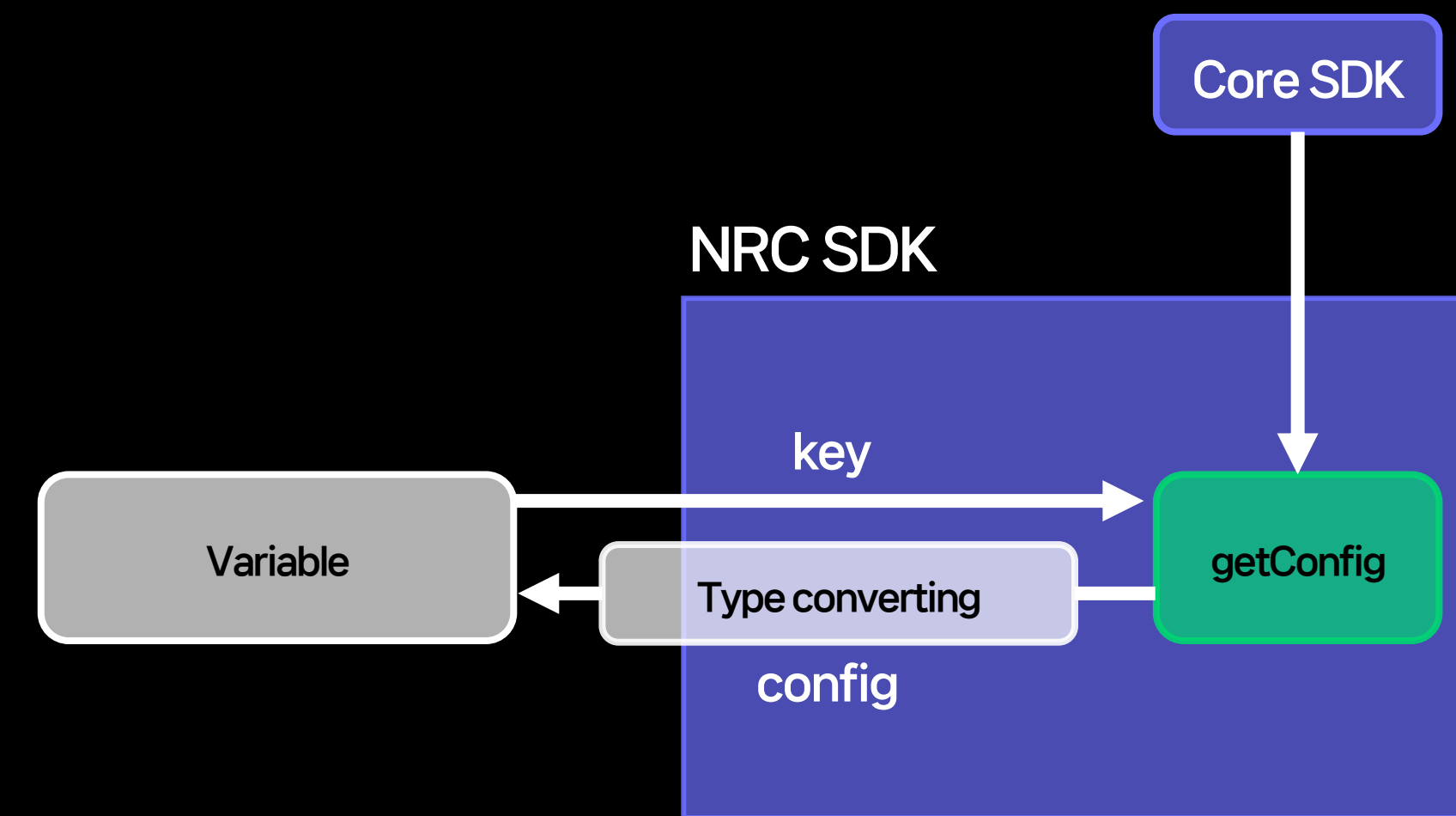
- 앱에서 주입하는 다양한 정보를 기반으로 환경 및 사용자 대상 targeting 지원



3.5 NRC (Naver Remote Config) 구조

getConfig interface 제공

- getConfig 함수를 통해 콘솔을 통해 받아온 특정 key 값에 대한 value 값 제공



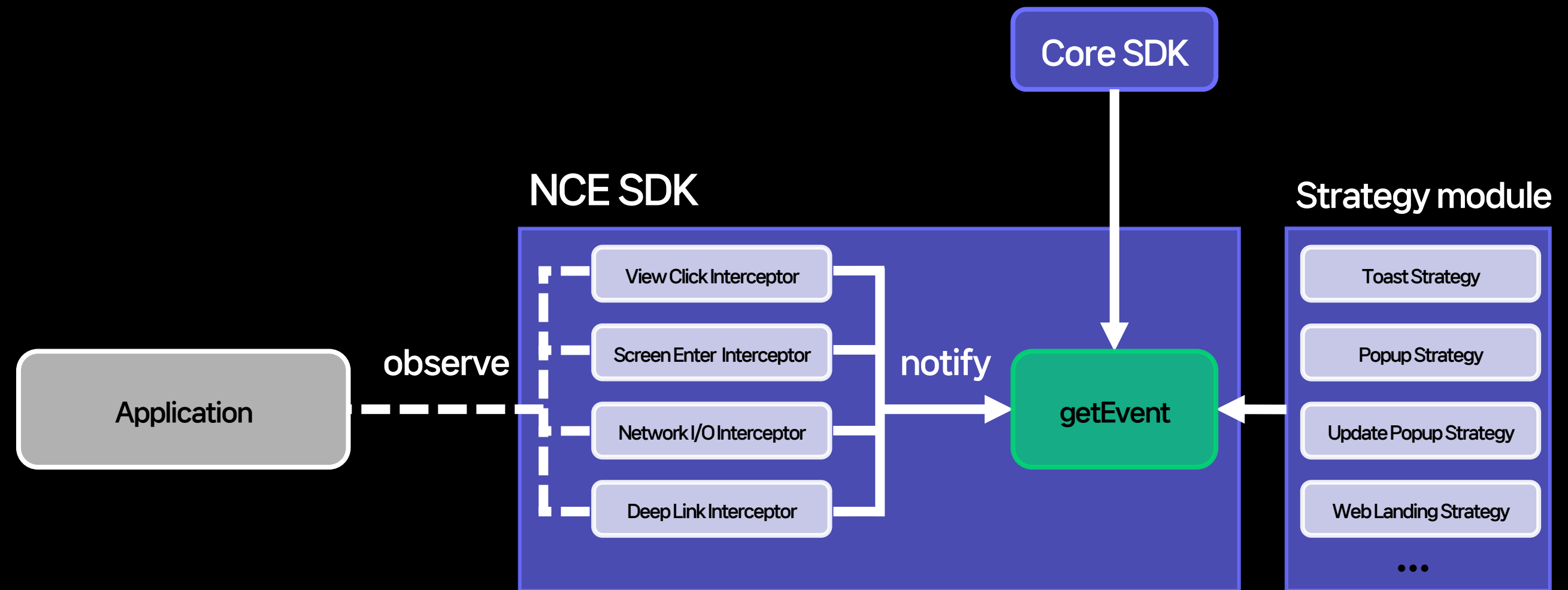
3.6 NCE (Naver Codeless Event) 구조

Auto Hooking & Strategy

- Event 를 hooking 하여 제공되는 전략 수행

Event Blocking & Counting

- 전략 주입 시 기존 로직 실행 여부를 optional 하게 설정 가능
- 전략 수행 횟수 설정 가능

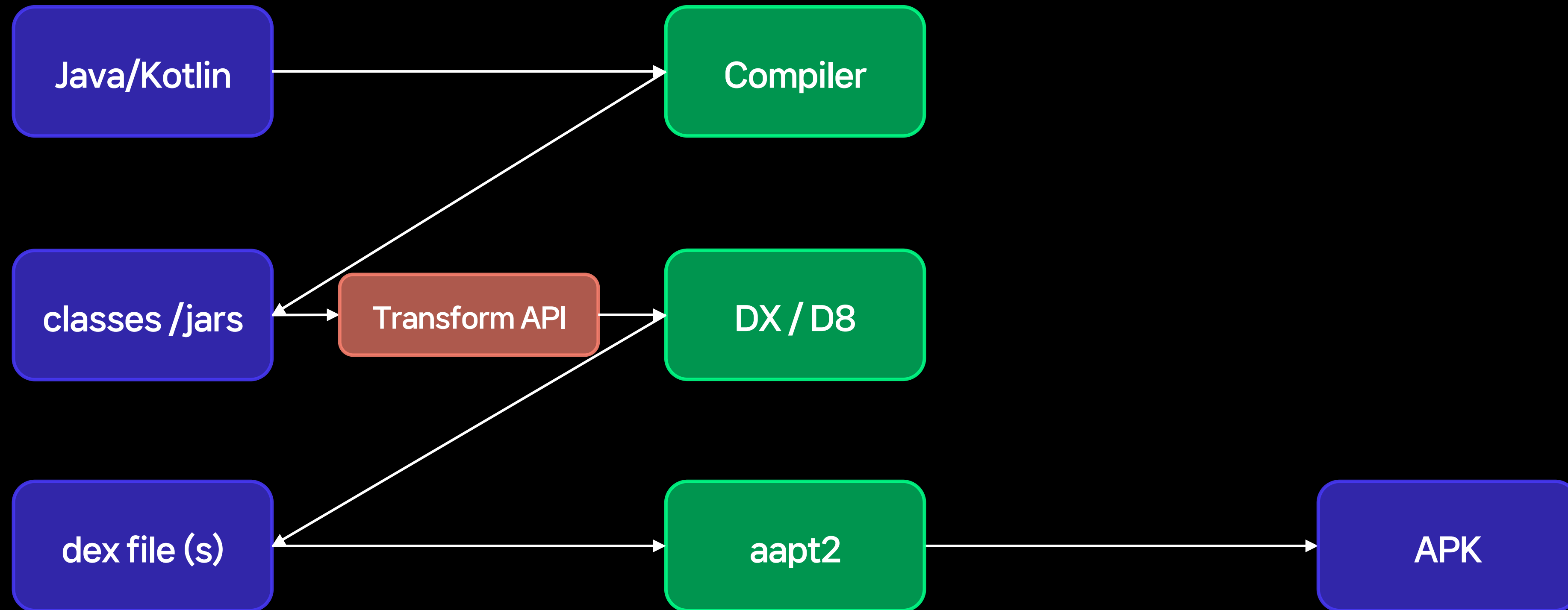


3.6 NCE (Naver Codeless Event) 구조

	앱 코드 수정 필요 여부	이벤트 후킹 방식
View Click	O	<ul style="list-style-type: none"> setOnClickListener 함수를 통해 OnClickListener 를 등록하는 것이 아닌 OnNCEClickListener 를 등록해주면, OnNCEClickListener 클래스 내부적으로 클릭 시점 후킹
Screen Enter	X	<ul style="list-style-type: none"> NCE SDK 내부적으로 init 시점에 ActivityLifecycleCallback 을 상속 받은 NCELifecycleCallback 을 등록하여 모든 Acitivity OnStart 시점 후킹
Network I/O	X	<ul style="list-style-type: none"> BCI 기술을 활용하여 Network I/O 를 담당하는 함수를 NCE SDK 내부에 선언된 함수로 교체하여 이벤트 시점 후킹
Deep Link	O	<ul style="list-style-type: none"> 딥링크를 통해 들어온 Activity 의 onCreate, onNewIntent 시점에 NCEDeepLinkInterceptor 클래스의 process 함수 호출을 통해 deep link 호출 시점 후킹

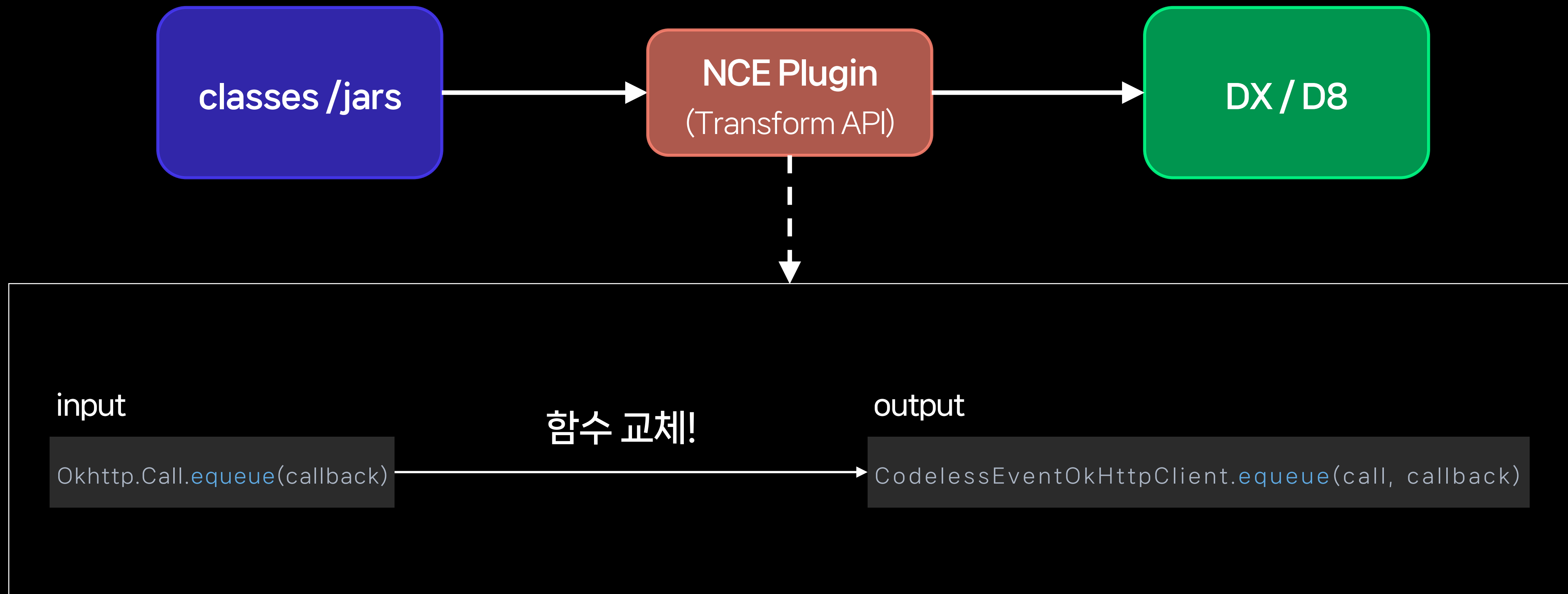
3.6 NCE (Naver Codeless Event) 구조

Bytecode Instrumentation (BCI)



3.6 NCE (Naver Codeless Event) 구조

Bytecode Instrumentation (BCI)

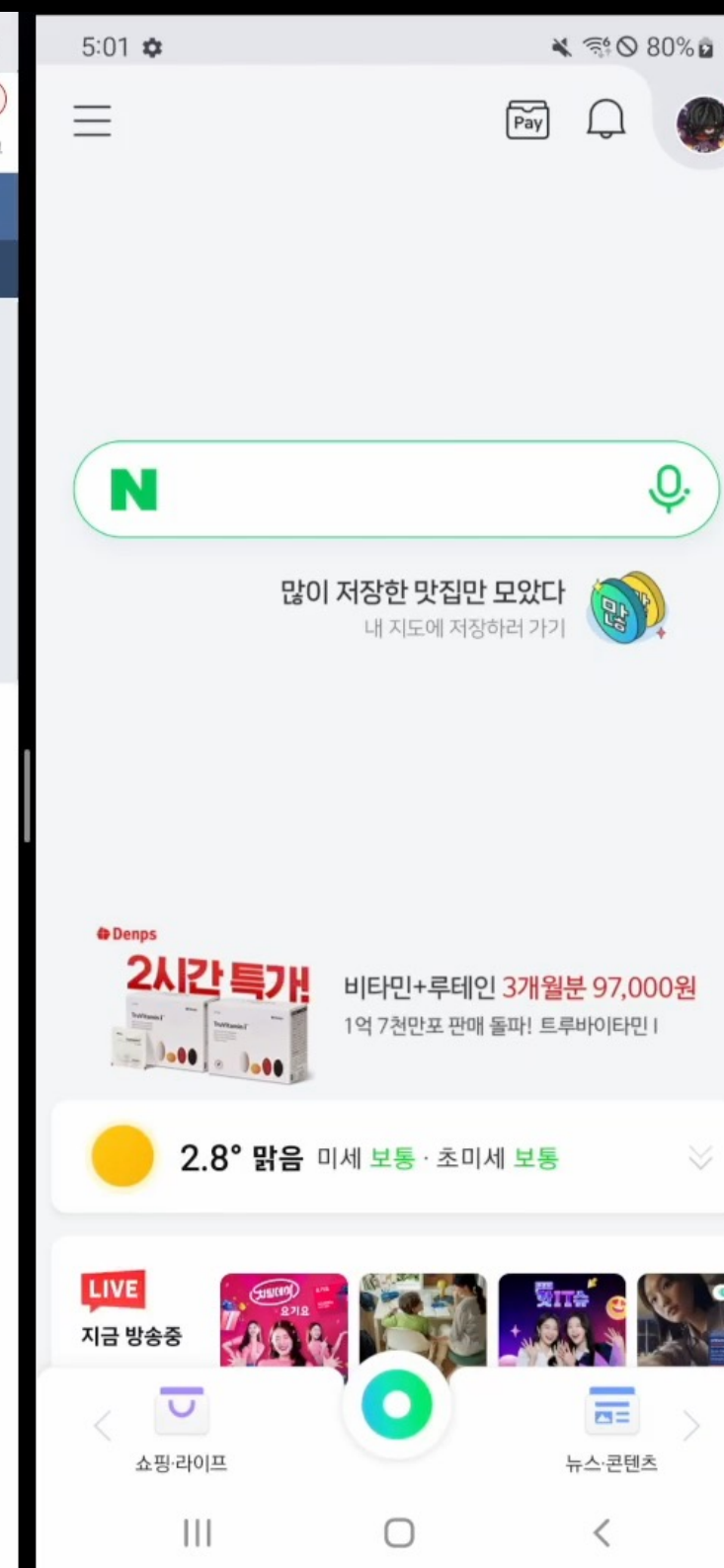
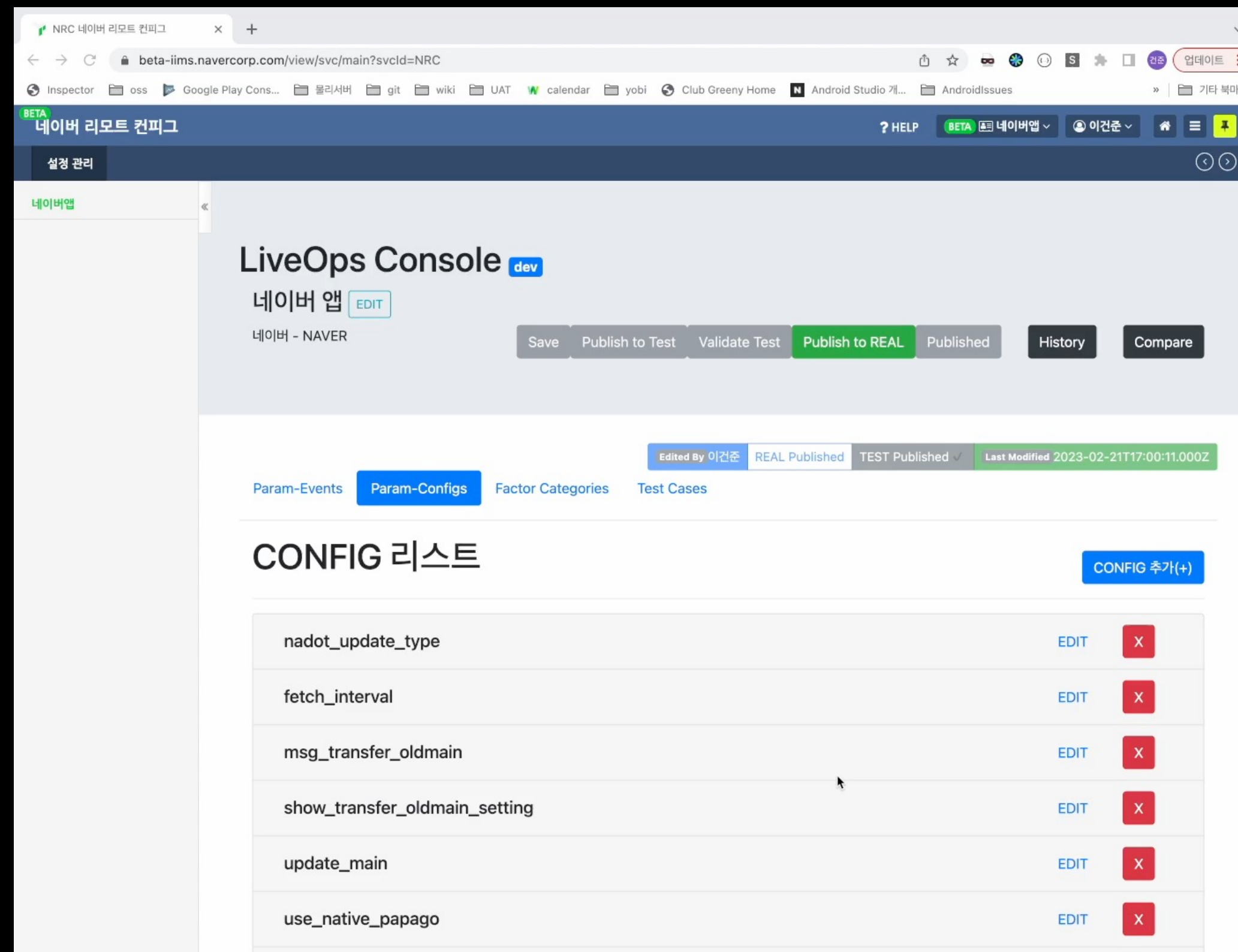


4. LiveOps 사용 사례

4.1 LiveOps NRC 사용 사례

네이버앱 네이티브/웹 검색홈 전환

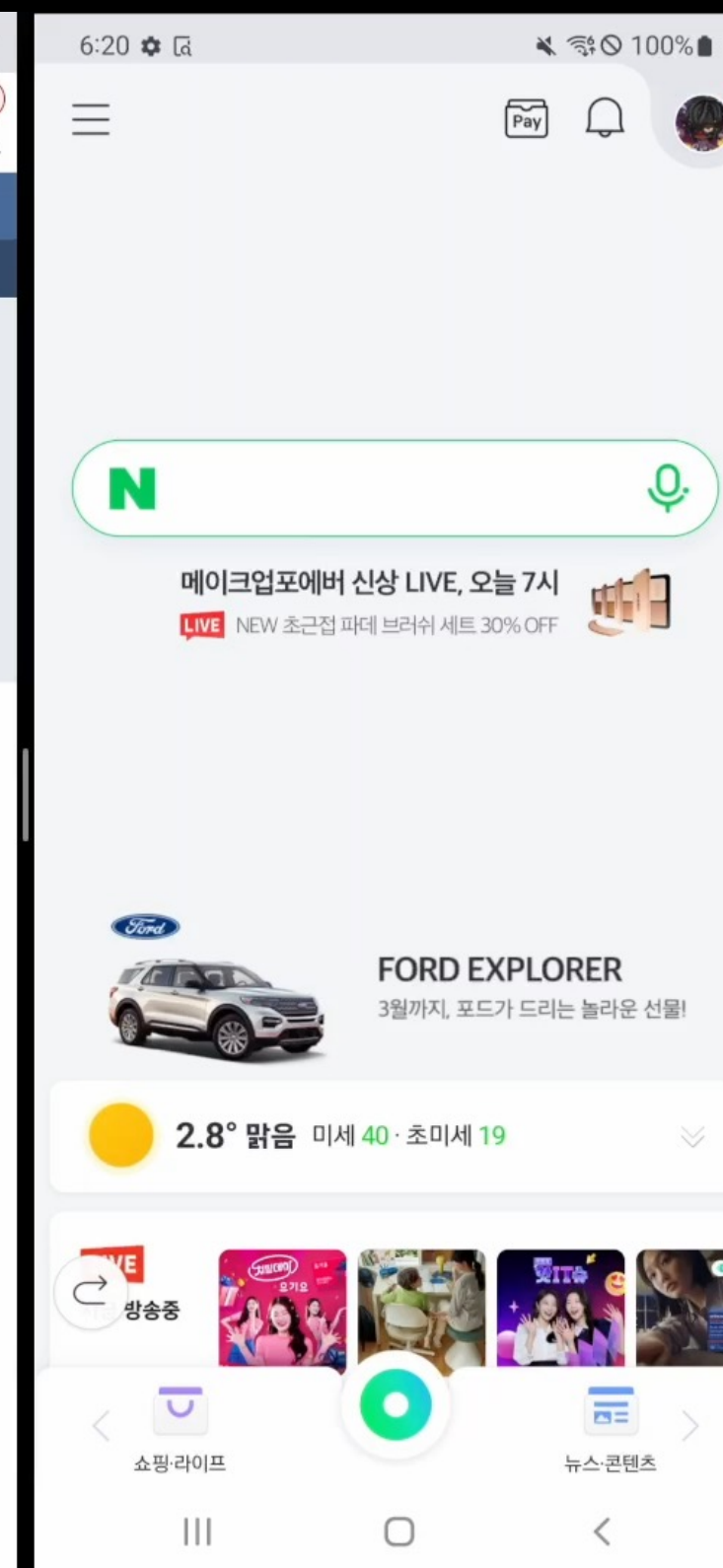
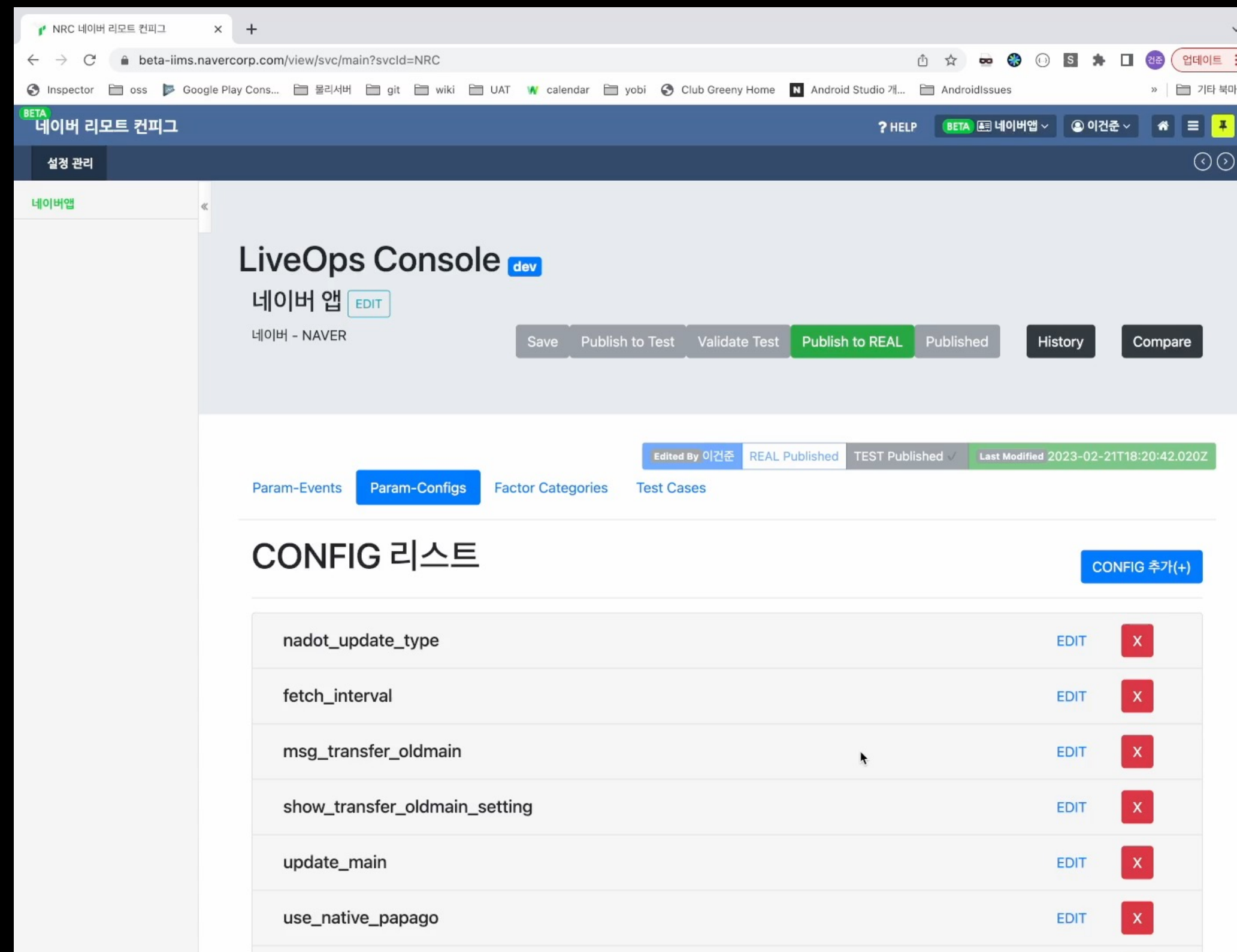
API 관련 이슈로 인해 네이티브 검색홈에서 장애가 발생하고 있는 상황에서 NRC 를 통해,
웹 검색홈으로 전환해주어 빠르게 장애 상황을 회피한 사례



4.1 LiveOps NRC 사용 사례

Static URL 변경

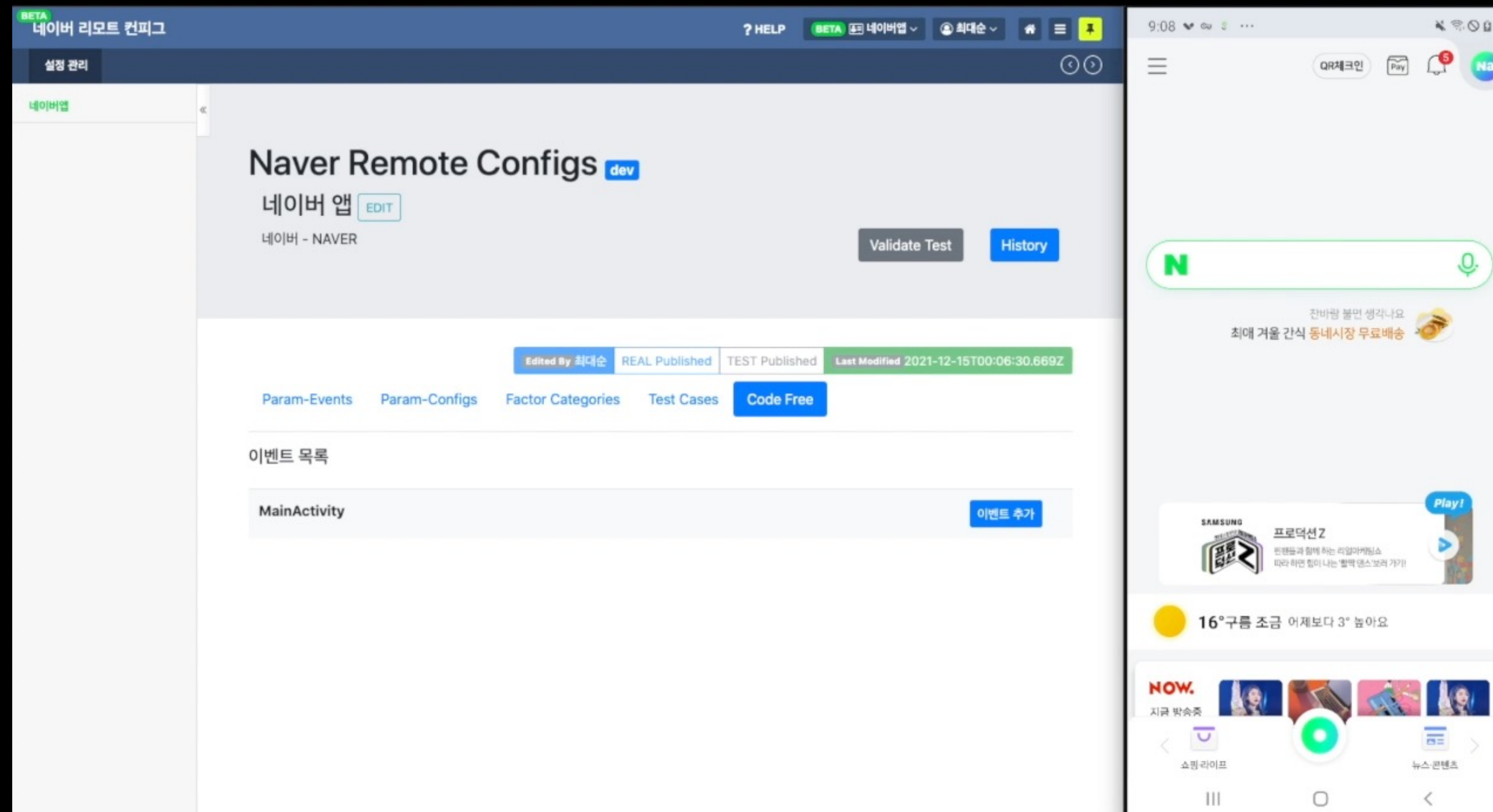
앱에서 static 하게 들고 있는 웹 URL 의 경우, 웹 URL 이 변경되거나 웹 서비스가 종료되는 등 더 이상 해당 URL 이 유효하지 않을 경우 대응이 불가능한 이슈를 대응한 사례



4.2 LiveOps NCE 사용 사례

주요 피쳐 장애 대응 방안

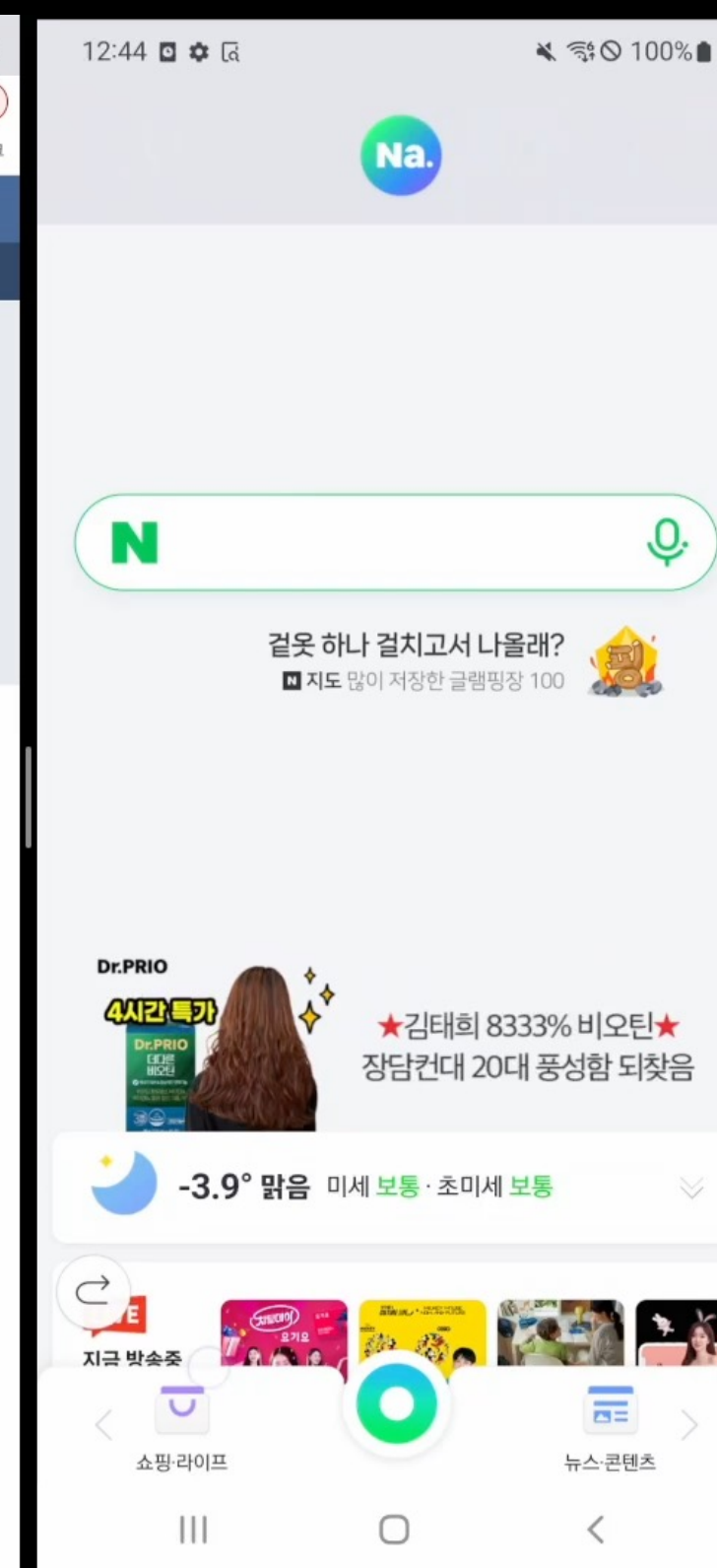
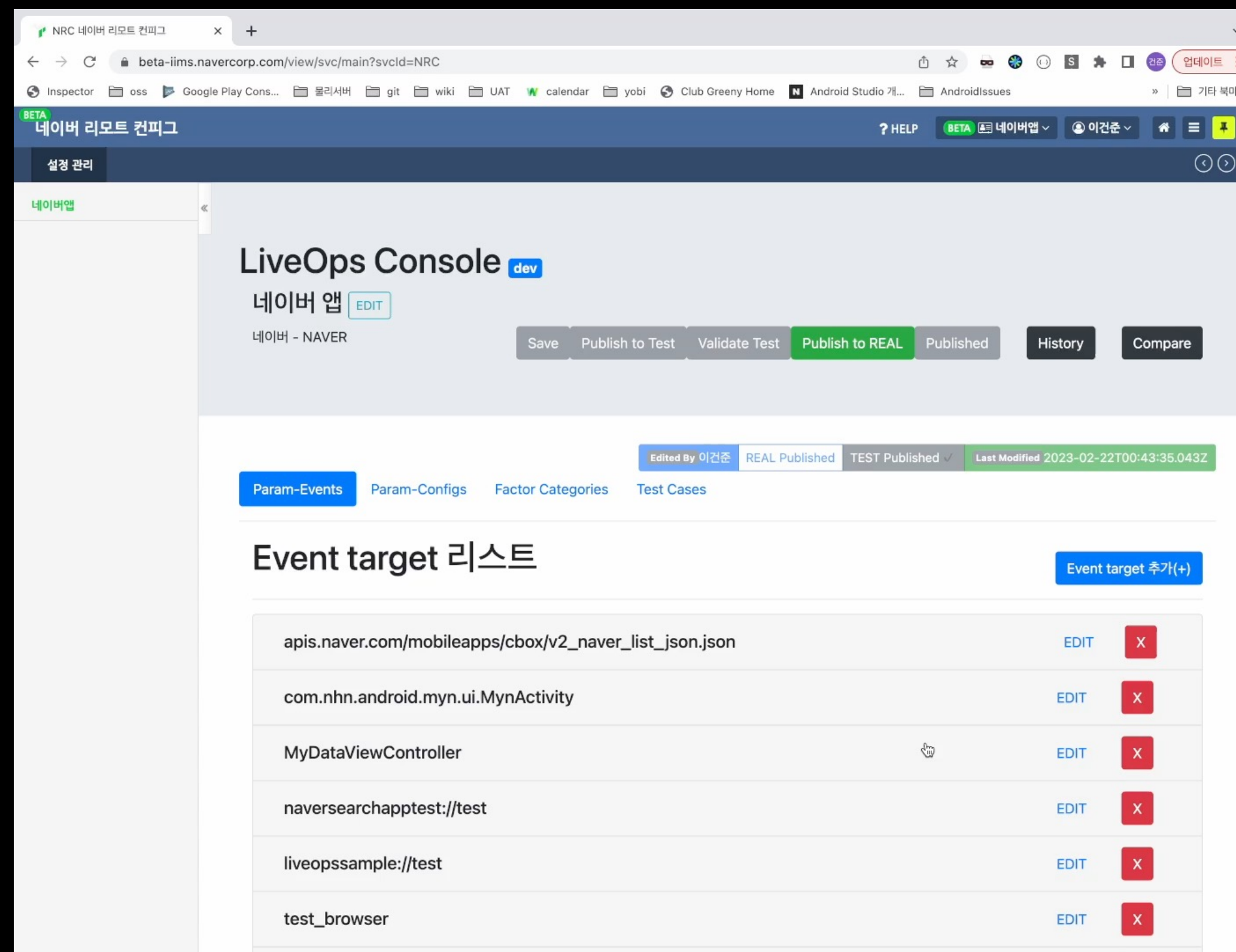
QR 체크인의 사용자가 많이 몰리는 상황을 대비하여 웹 QR 체크인으로 우회시키는 전략을 가지고 있었다



4.2 LiveOps NCE 사용 사례

NA. 신규 기능 공지 및 업데이트 유도

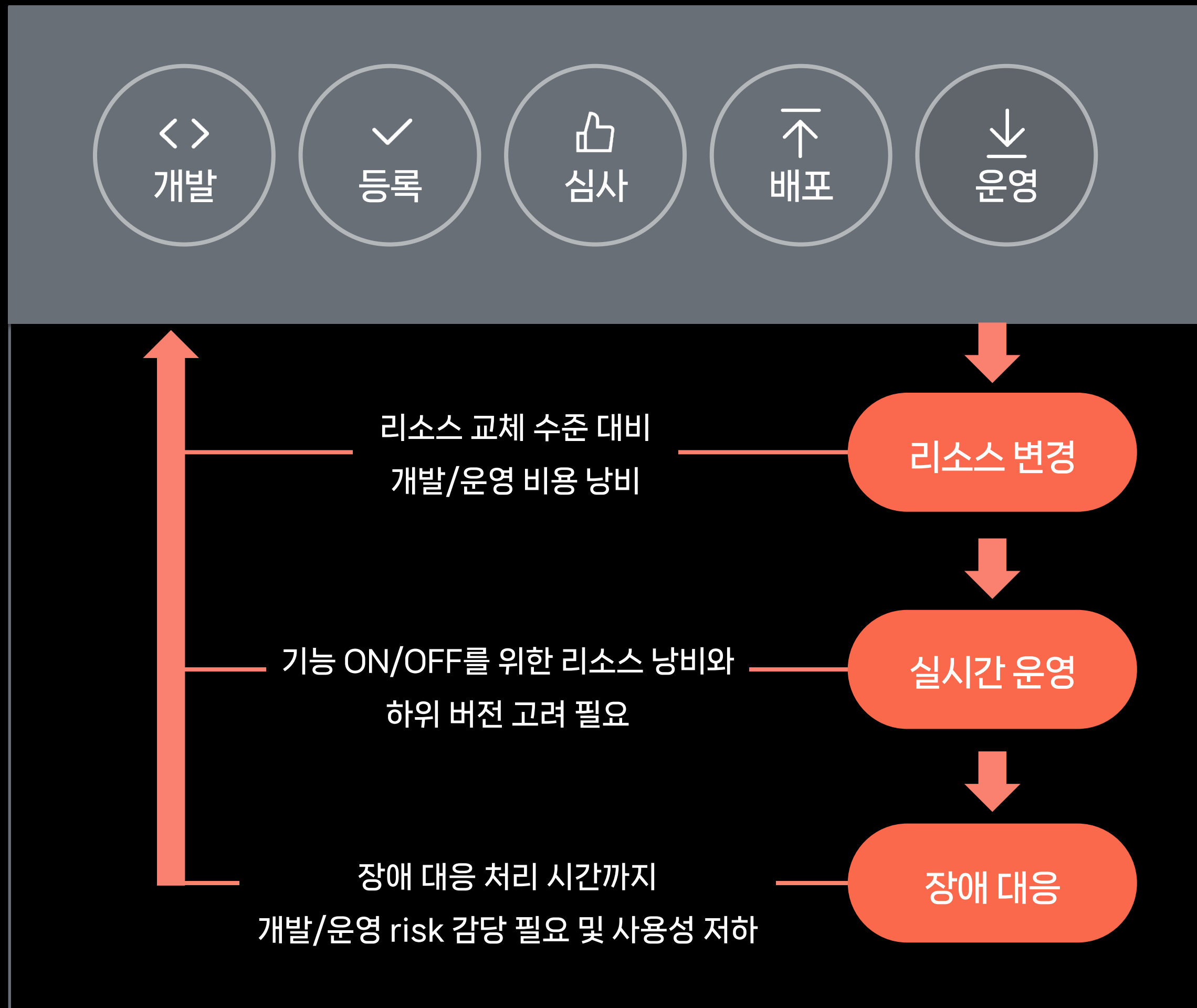
정부 정책 대응을 위해 신규 접종 증명서 기능 공지 및 업데이트 유도가 필요했던 사례



5. LiveOps 도입 효과

5.1 LiveOps 도입 효과

AS-IS



TO-BE



5.2 LiveOps 향후 계획

LiveOps 를 활용한 배포 프로세스

- Feature 점진적 배포
- Feature Rollback 프로세스
- A/B 테스트

NCE Event 후킹 시점 추가 및 전략 추가

- Event 후킹 시점 추가
- Event 후킹 시 활용 가능한 전략 추가

NCE Network I/O 기능 강화

- Network I/O Event 후킹 시 전략 추가

"앱이라서" 불가능합니다 ㅠ

"앱이여도" 가능합니다 ㅎㅎ

감사합니다

Q & A