

# ALFEIOS

WIZEYE를 활용한 서울시 하천 관제 시스템

+

---

N3N / SEOUL

Web 파트 연구원  
김선호





## I. 주제 선정

- II. 프로젝트 설계
- III. ALFEIOS 솔루션 제작
- IV. 문제점
- V. Q&A

## I. 주제 선정

### 1. 프로젝트의 목적

# 목적

N3N WEB 파트에서 개발하고 있는 WIZEYE Web 버전을 통한 적절한 솔루션의 구축을 통하여, WIZEYE에 대한 이해도를 높이는 것

## 추가 고려사항

- WIZEYE에서 제공하는 최대한 많은 기능을 사용해 볼 것
- 새로 추가된 ADP를 사용하여 DATA FLOW를 구성할 것
- 단순한 Dummy가 아닌 의미 있는 데이터를 사용할 것

## 1. 주제 선정

## 2. WIZEYE의 목적에서 살펴본 적절한 프로젝트 주제의 형태

## WIZEYE

기업의 각 영역별 담당자들이 공통적으로 높은 Insight 를 가지고 협업을 통해 빠른 시간 내에 강력한 문제 해결 및 개선은 물론 비즈니스 요구에 적시적으로 대응할 수 있는 새로운 패러다임의 플랫폼 서비스

가지고 있는 데이터를 **목적에 맞게 그 관계를 규명 및 구조화**하고  
데이터를 언제나 **구조·계층화할 수 있는 기반을 제공하는데 목적**이 있다

(출처 - WIZEYE BROCHURE v5.2)



- 데이터의 **계층 구조**를 형성할 수 있을 것
- 데이터의 변화를 확인할 수 있도록 **데이터의 값이 변화**할 것
- 데이터 자체만으로 전체적인 변화를 **쉽게 예측하기 어려울** 것

## I. 주제 선정

### 3. 서울시 하천 수위 데이터의 주제 적합성

#### ➡ 계층 구조로 나타낼 수 있는 데이터인가?

서울시 하천은 제 1지류를 기준으로 계층을 분류하면, 서울 시내 하천 지류들은 고덕천, 성내천, 탄천, 중랑천, 반포천, 봉원천, 홍제천, 안양천, 향동천, 창릉천, 굴포천, 아라천 12개의 지류로 나뉘어 계층 구조로 나타내기에 적합하다.

#### ➡ 변화하는 데이터인가?

서울 열린데이터 광장(<http://data.seoul.go.kr/>)에서 제공하는 서울시 하천 수위 현황 API는 관측소의 이름 및 관측소가 속한 지류의 이름, 제방고의 높이 및 현재 수위 등을 1분마다 갱신해줌으로써, 지속적으로 데이터의 변화를 제공한다.

#### ➡ 쉽게 파악되지 않는 데이터인가?

전체적인 하천이 이어져있음에도 불구하고 관측소의 정보가 지류 별로 구분되는 것이 아닌, 병렬적으로 전달되기 때문에 제1지류와 제2지류, 본류와의 관계성을 파악하기 어렵다.

I. 주제 선정



## II. 프로젝트 설계

III. ALFEIOS 솔루션 제작

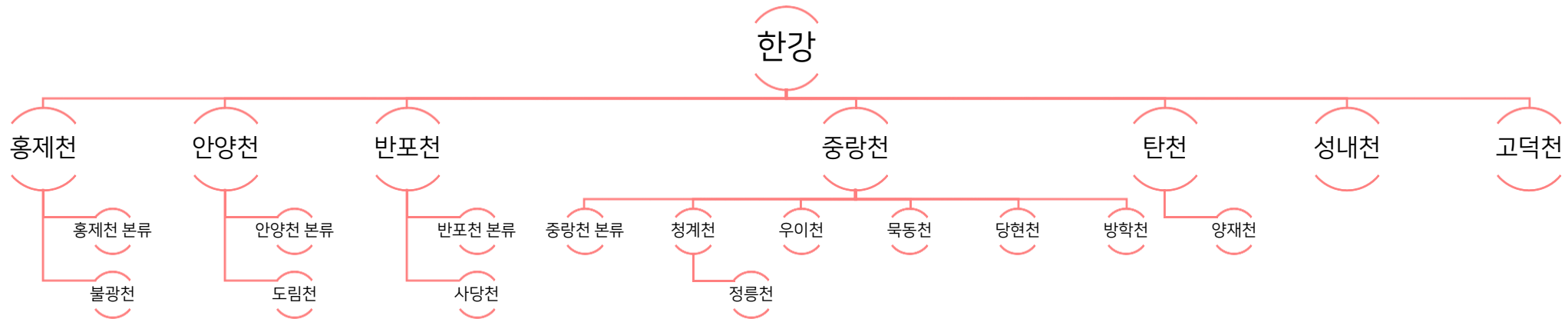
IV. 문제점

V. Q&A

## II. 프로젝트 설계

### 1. 필요사항 정의하기

- 하천 간의 관계도



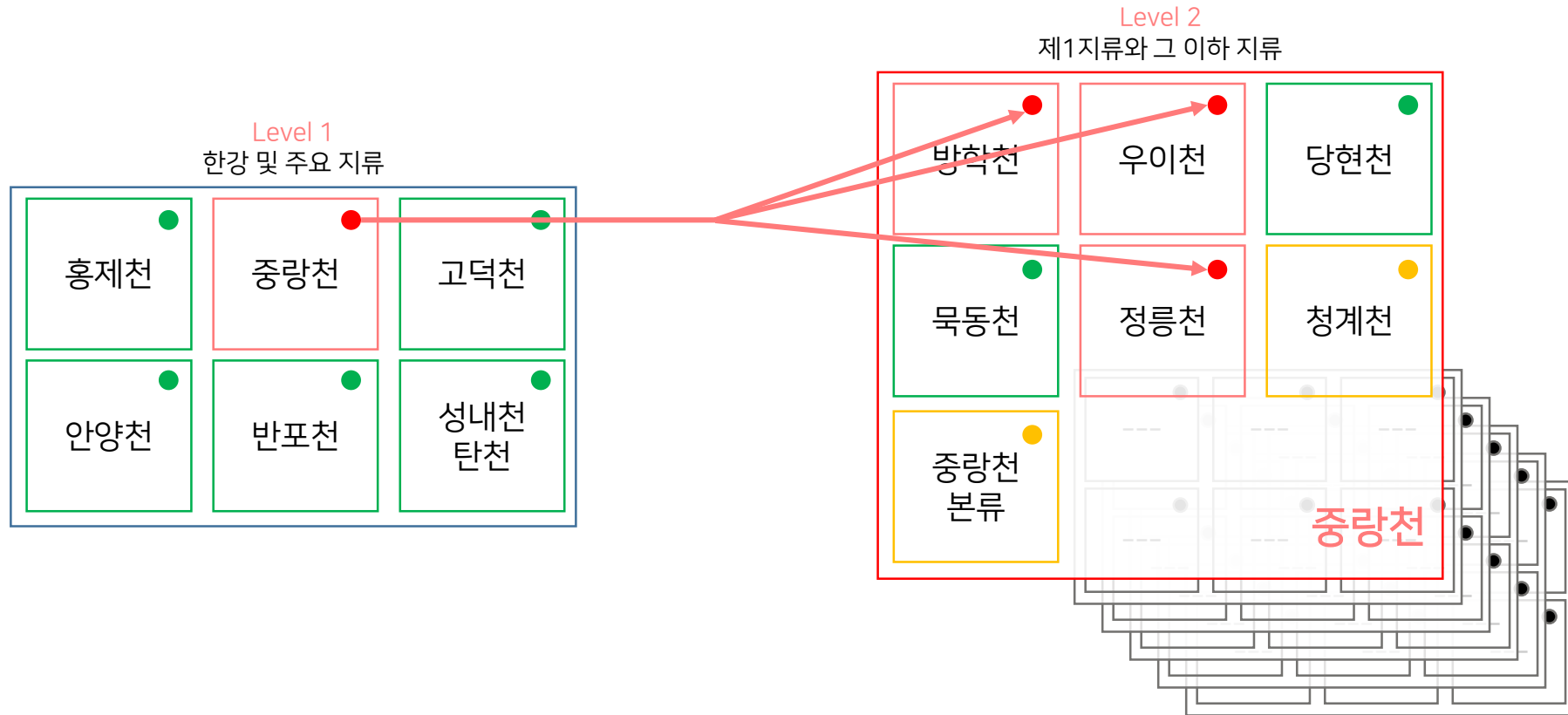
출처 - 하천 지리 정보 시스템(2011)  
(수위를 관측할 수 없는 복개천이나 관측소가 없는 하천은 제외)

범람 위험도에 대한 상태는 다음과 같이 정의한다.

- 현재 수위와 최대 수위(제방고의 높이)를 감안하여 범람위험도를 %로 표기한다
- 범람 위험도 30% 이하 : 정상, 70% 이상 : 위험

## II. 프로젝트 설계

## 2. 커스텀 맵 설계



최상위 레벨에서 하위 하천의 범람 유무를 알 수 있도록 설정

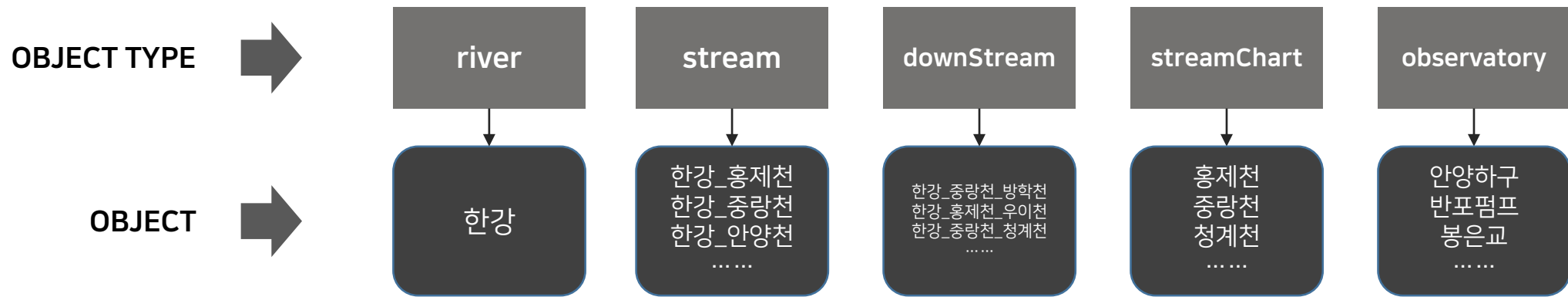
- 하천 레벨을 기준으로 하위 레벨 범람 시 상위 레벨 하천으로 범람된 하천의 개수를 카운팅하도록 설정



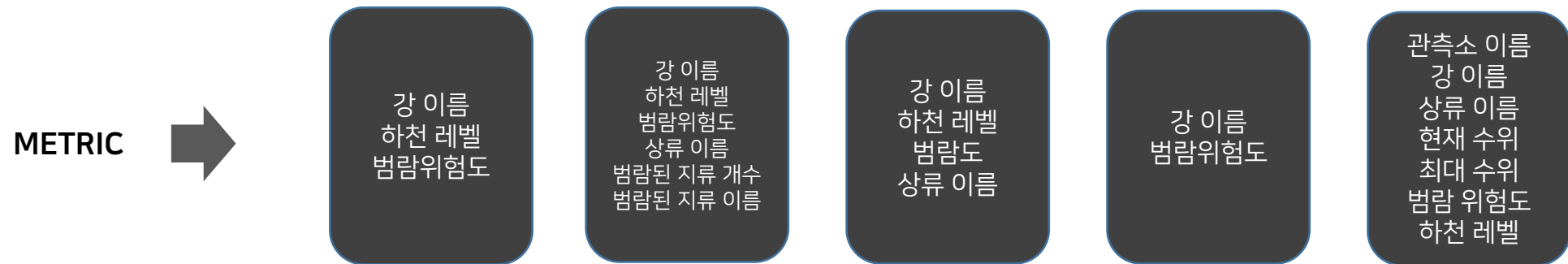
## II. 프로젝트 설계

## 3. DATA 설계하기

## 1단계. OBJECT와 OBJECT TYPE을 정의



## 2단계. OBJECT 공통적인 속성인 METRIC 정의



## II. 프로젝트 설계

### 3. DATA 설계하기

#### 3단계. 맵 바인딩할 METRIC 값들의 상태를 정의

1. 범람위험도



범람위험도 < 30

범람위험도 < 70

범람위험도 >= 70

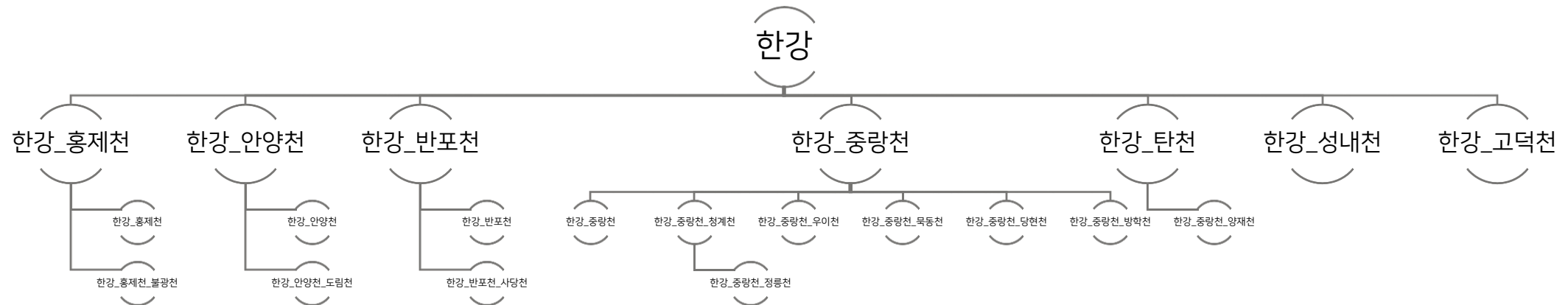
2. 범람된 지류의 개수



범람된 하천의 개수 = 0

범람된 하천의 개수 > 0

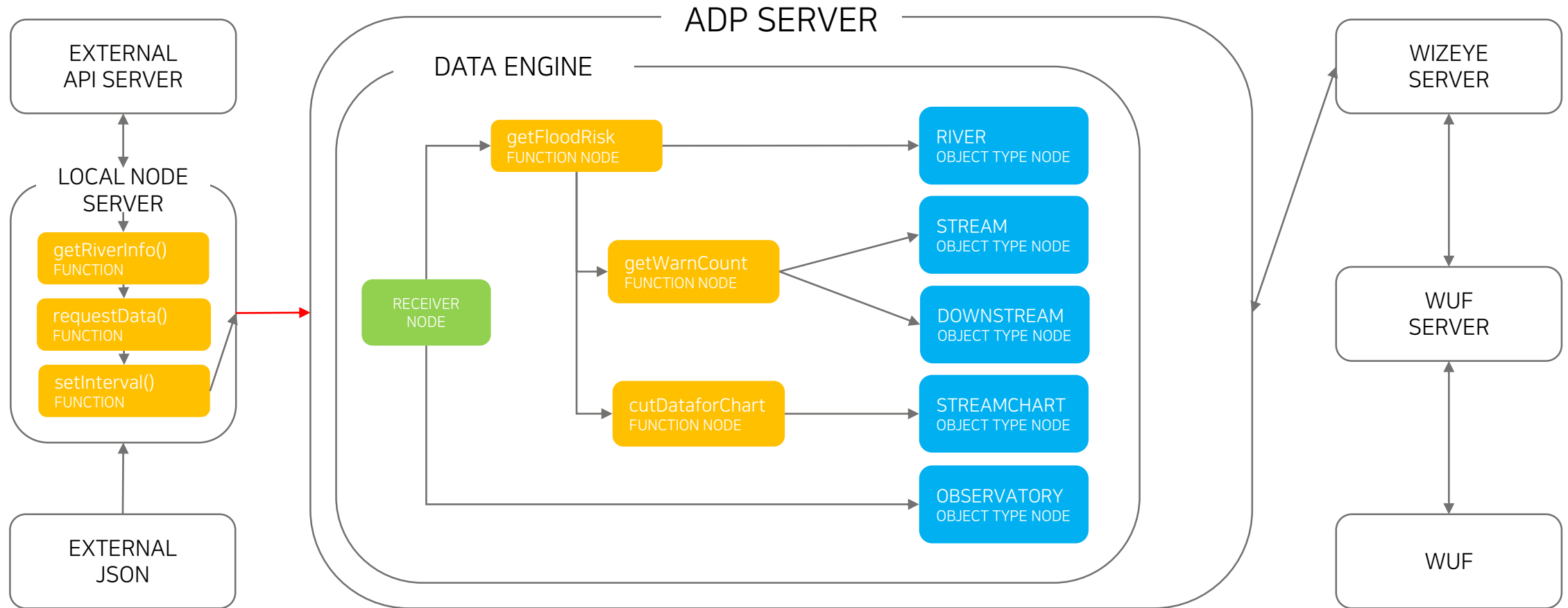
#### 4단계. OBJECT 간 상·하 관계 정의



## II. 프로젝트 설계

### 3. DATA 설계하기

#### 5단계. DATA FLOW 정의



- I. 주제 선정
- II. 프로젝트 설계



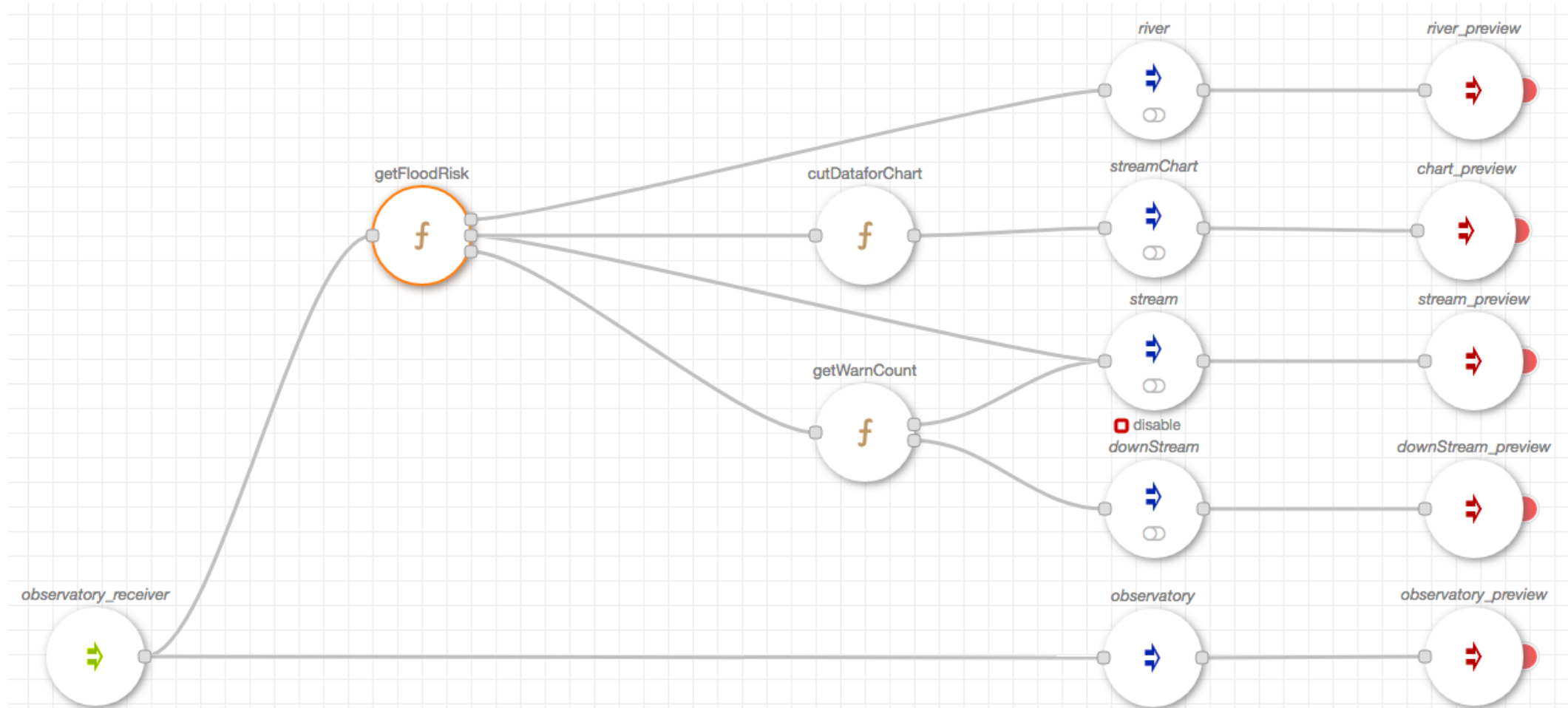
## III. ALFEIOS 솔루션 제작

- IV. 문제점
- V. Q&A

## III. ALFEIOS 솔루션 제작

## 1. ADP를 활용한 DATA FLOW 제작

## 1단계. ADP의 DATA ENGINE을 활용한 NODE 정의



## III. ALFEIOS 솔루션 제작

## 1. ADP를 활용한 DATA FLOW 제작

## 2단계. 각 FUNCTION NODE 기능 정의

## getFloodRisk

1. 전달받은 관측소 데이터를 riverName 이라는 metric의 값을 기준으로 통합
2. curRiverGauge와 maxRiverGauge라는 metric을 통해 구해진 floodRisk라는 이름의 metric 값을 추가
3. level에 맞게 3 집단으로 분류하여 각각의 output으로 분배

## getWarnCount

1. 전달받은 최하위 하천 데이터 중 floodRisk가 70을 기준으로 데이터 분류
2. 분류된 데이터의 upStream metric 값을 프로퍼티로 하는 객체를 형성하여 범람 정보 전달
3. 범람 정보를 갖고 있는 객체를 stream object type node로 데이터 전송

\* Wizeye의 비동기 기능을 활용한 것으로, 추후 aggregate가 나오면 수정해야함

## cutDataforChart

현재 chart에 legend 값 필터링이 불가능하기 때문에 만들어진 임시 function node

floodRisk, riverName metric을 제외한 모든 metric을 제거

\* 상세 코드는 깃허브에서 확인 가능(<https://github.com/skyho31/alfeios>)

\*\*주의사항 - 송수신되는 데이터는 [Object].payload 안에 JSON 배열 형태로 만들어져야 합니다

## III. ALFEIOS 솔루션 제작

## 1. ADP를 활용한 DATA FLOW 제작

## 3단계. API 요청을 위한 LOCAL NODE SERVER 만들기

\* 상세 코드는 깃허브에서 확인 가능(<https://github.com/skyho31/alfeios>)

## requestData()

Node.js의 request 모듈을 통한  
ADP로의 data 전송

## getRiverInfo()

외부 API에 실시간 하천 정보 요청  
후 상수 데이터가 담긴 로컬 JSON  
파일과 정보를 통합하여  
requestData 함수로 전달

## EXTERNAL JSON

하천 데이터 중 변하지 않는 값을  
JSON의 형태로 저장하여,  
alfeios.js로 제공

## 특이사항

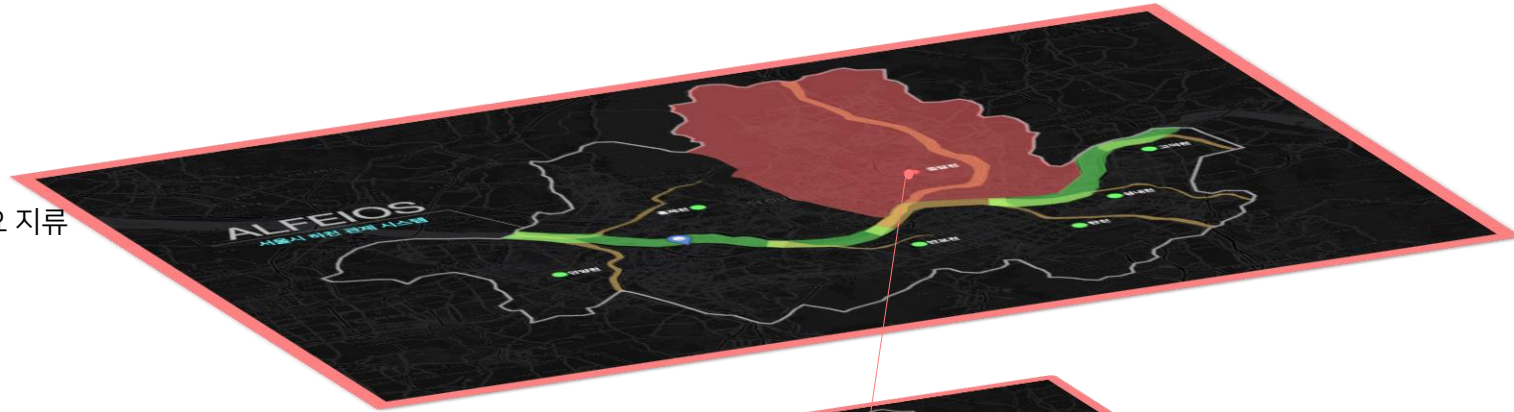
- ADP에 한글 값을 전달하면 preview node에서 ??로 나오는 현상 발생
  - API로 오는 모든 한글 값을 영문으로 변경해야 할 필요 발생
  - 하천 레벨 또한 따로 표기되어 있지 않아 수작업으로 분류해줘야 함
- 로컬 서버에서 각각의 값을 영문으로 변환해주는 함수 제작
- 효율성을 위해 관측소 이름을 프로퍼티 값으로 하는 상수 JSON 파일 제작 ( 현재 수위 데이터만 통합 후 전송 )

## II. 프로젝트 설계

### 2. 커스텀 맵 만들기

- 결과 Map

Level 1  
한강 및 주요 지류



Level 2  
제1지류와  
그 이하 지류

