

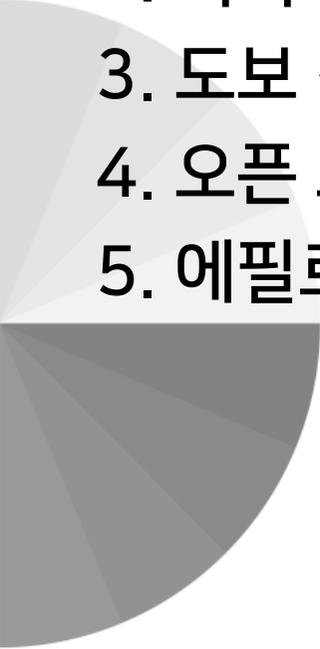
주니어 개발자의 도보 길찾기 서버 개발기



이석원 NAVER



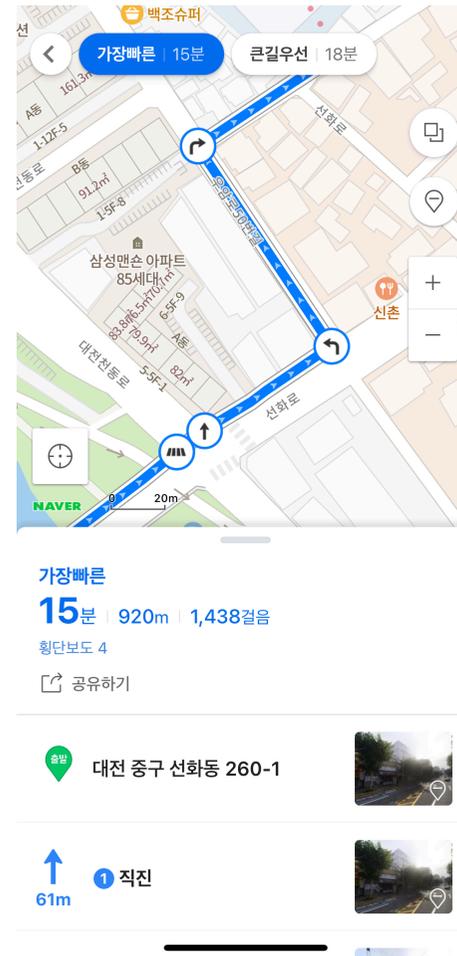
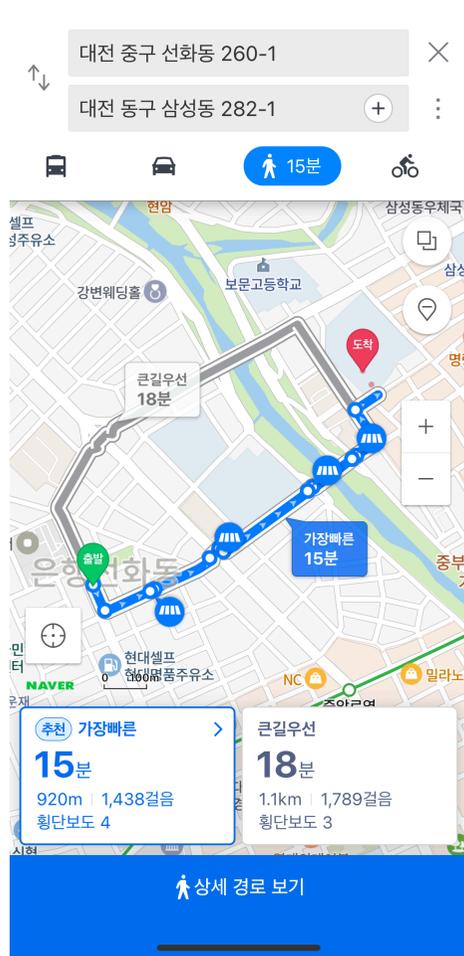
CONTENTS

1. 프롤로그
 2. 서버 아키텍처
 3. 도보 경로 탐색
 4. 오픈 소스 활용기
 5. 에필로그
- 



1. 프롤로그

1.1 도보 길찾기 서비스

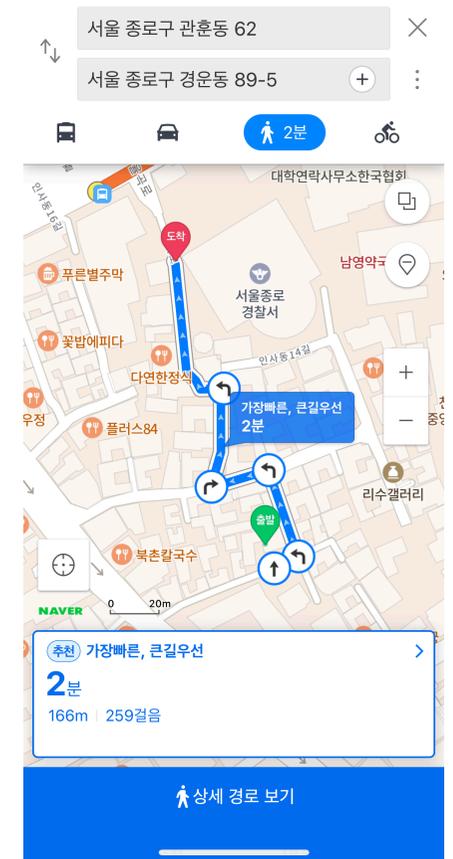
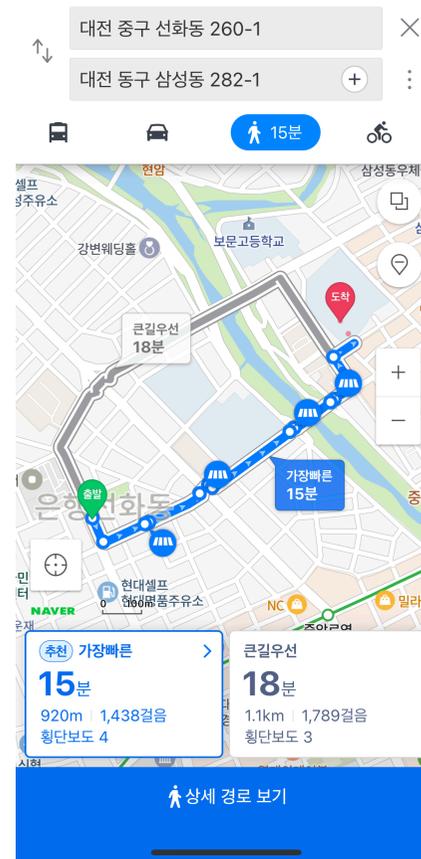


1.1 도보 길찾기 서비스

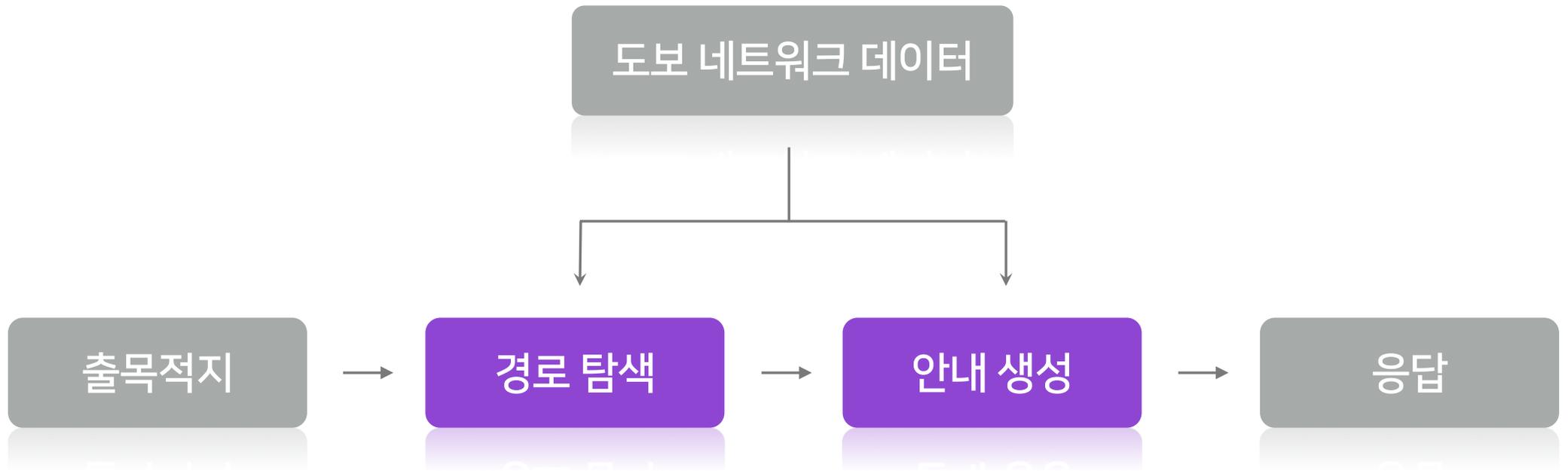
좋은 길을

유용한
안내와
함께

빠르게
알려주자!



1.1 도보 길찾기 서비스



1.2 마주한 상황



버그 수정

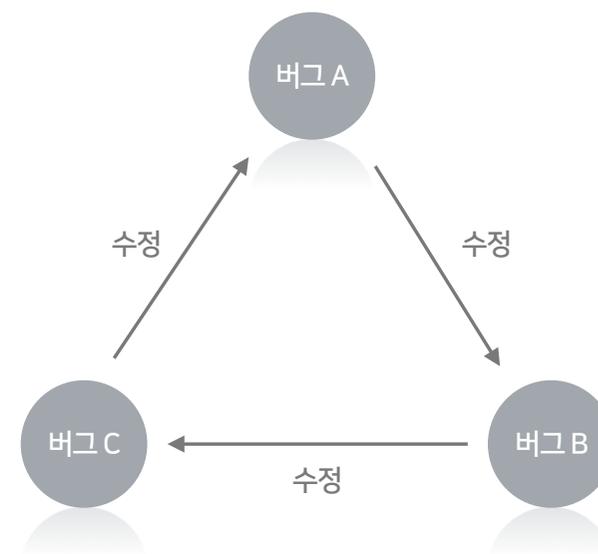


데이터
변경 대응



기능 추가

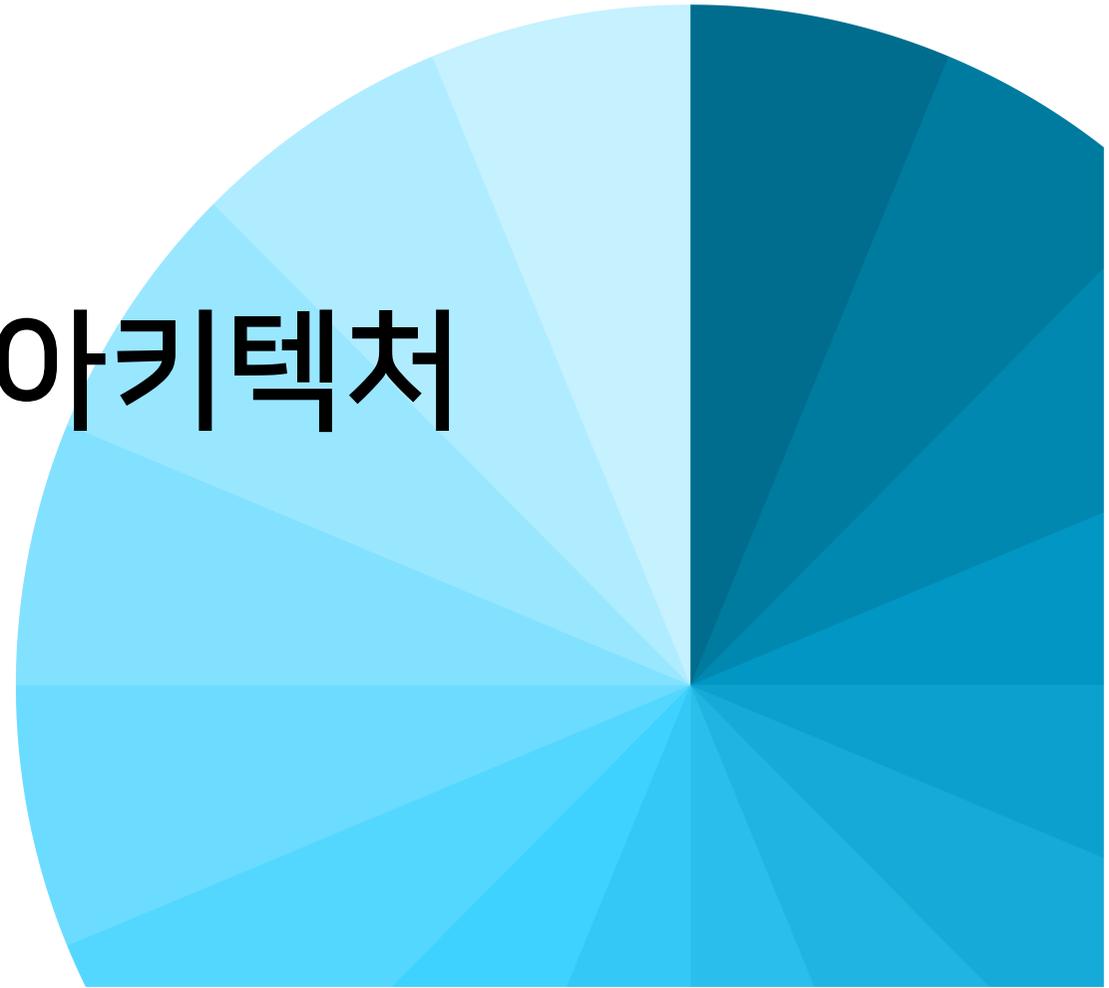
1.2 마주한 상황



1.2 마주한 상황

도보 길찾기 서버를 새로 작성하기로 결정!

2. 서버 아키텍처



2.1 서비스 요구사항

무엇을 지원해야 하는가?



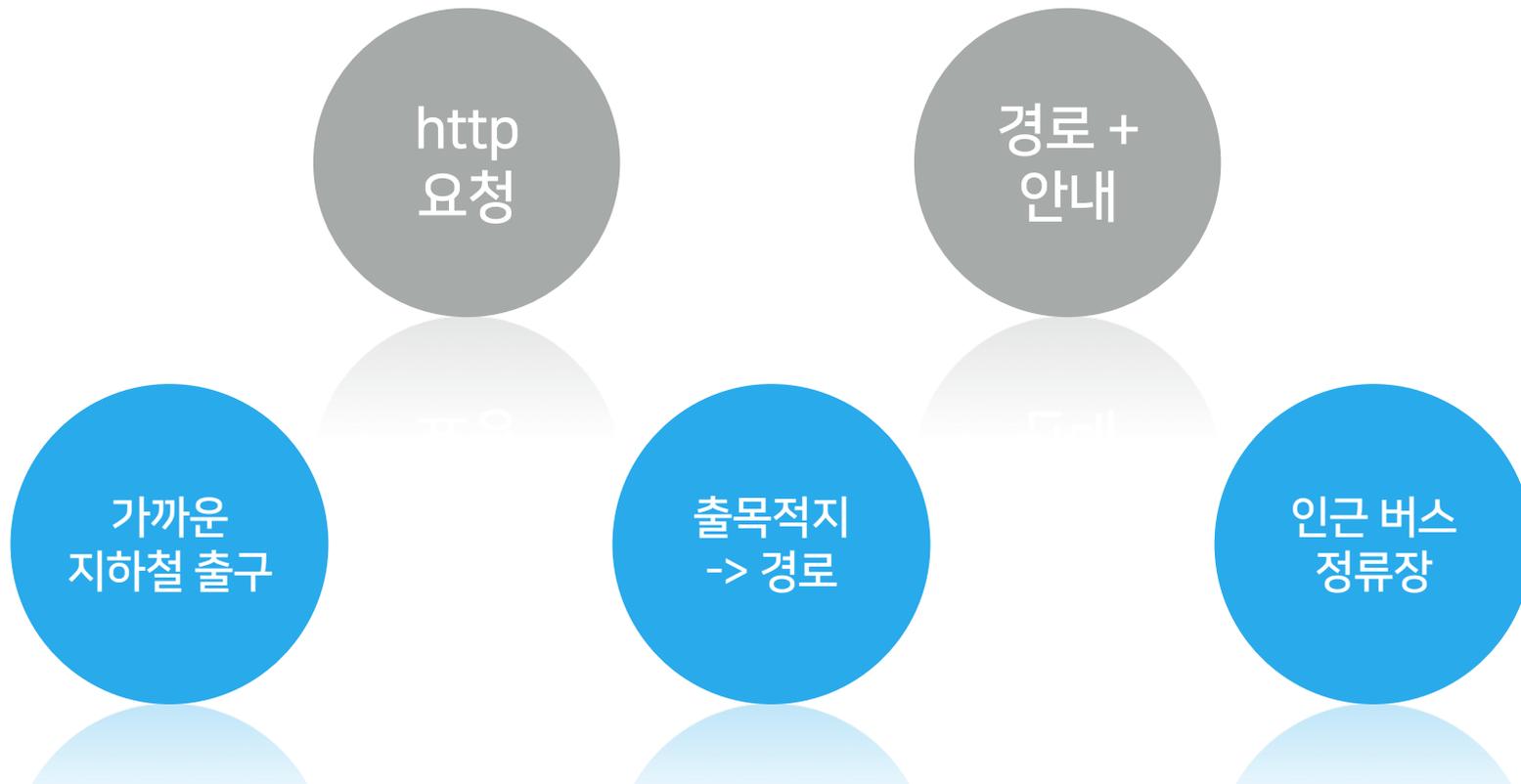
http
요청



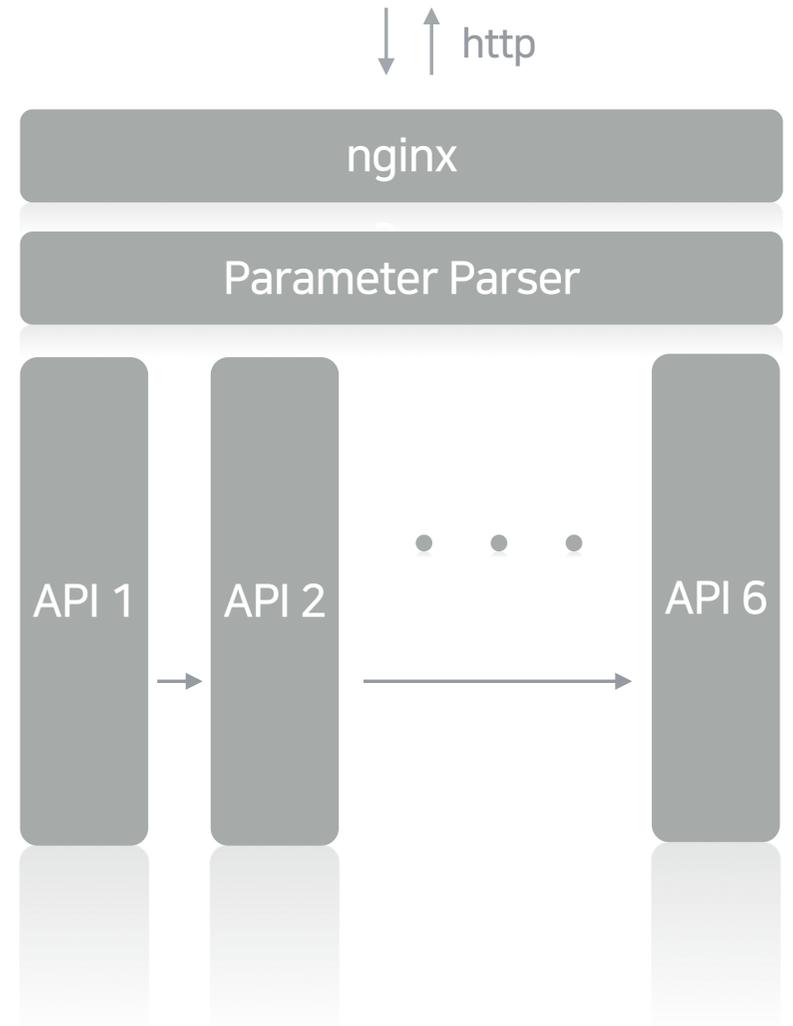
경로 +
안내

2.1 서비스 요구사항

무엇을 지원해야 하는가?



2.2 기존 아키텍처



2.3 신규 아키텍처 - 서버

각 프로그램이 하나의 일을 잘할 수 있게 만들 것

- 더글러스 매클로이

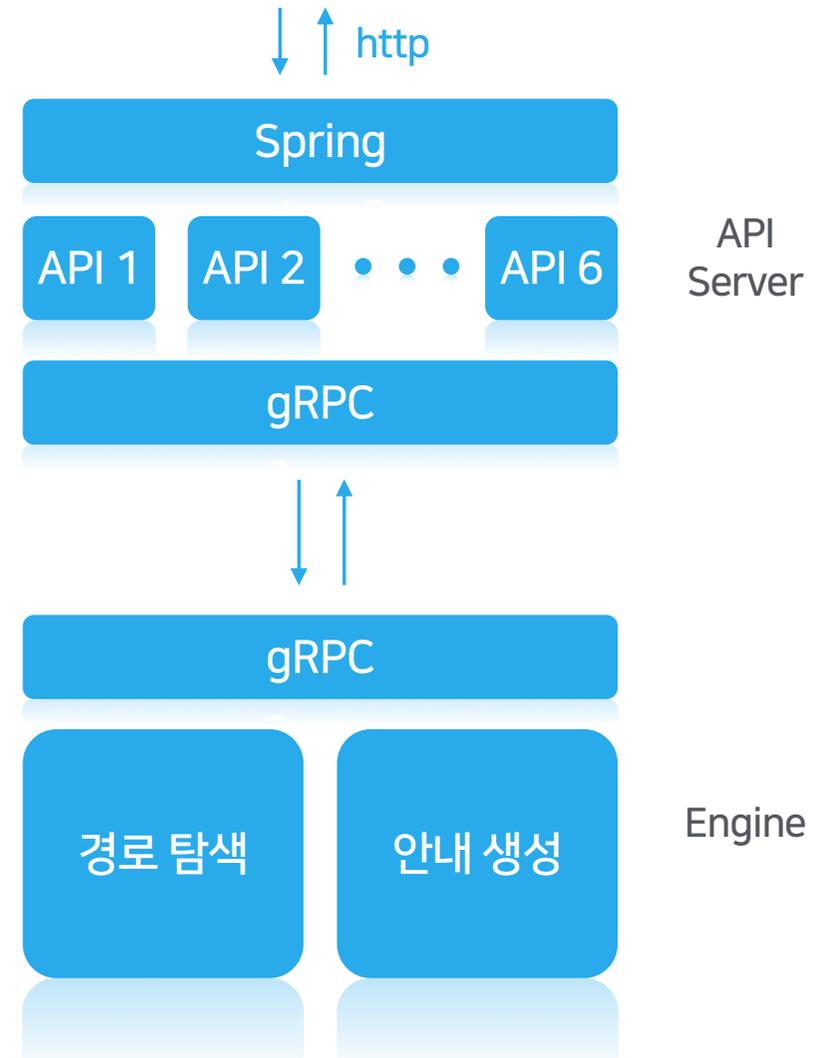
2.3 신규 아키텍처 - 서버

각 프로그램이 하나의 일을 잘할 수 있게 만들 것

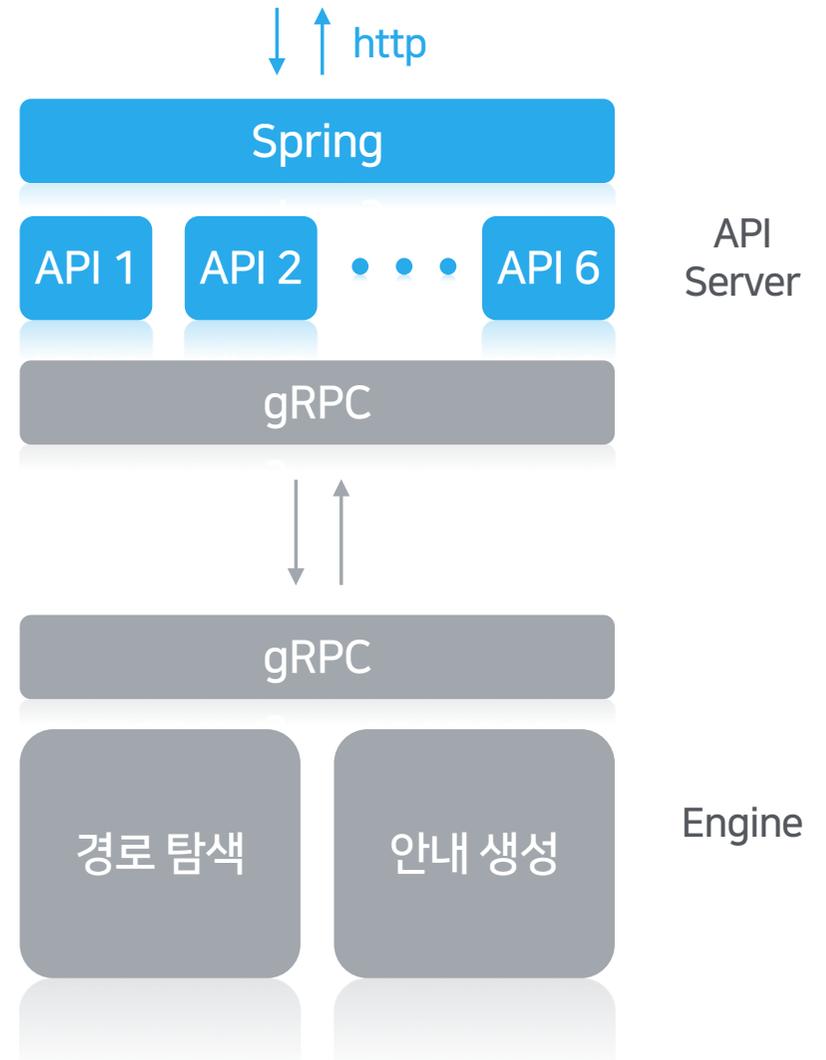
- 더글러스 매클로이



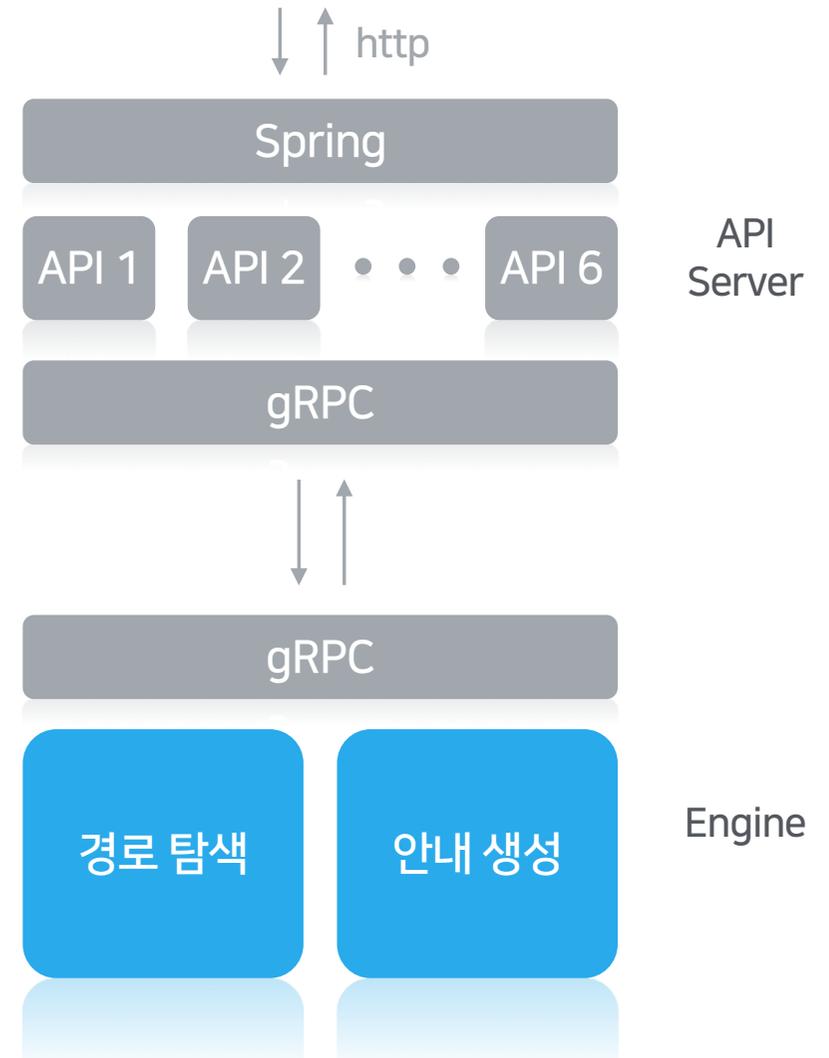
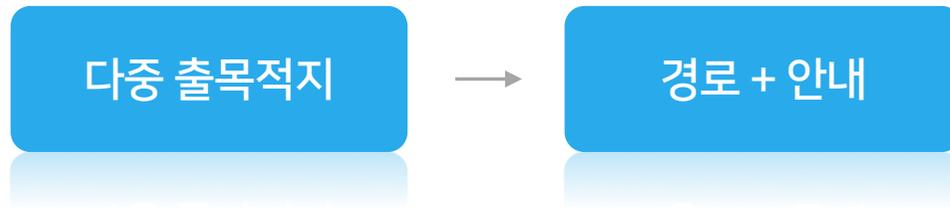
2.3 신규 아키텍처 - 서버



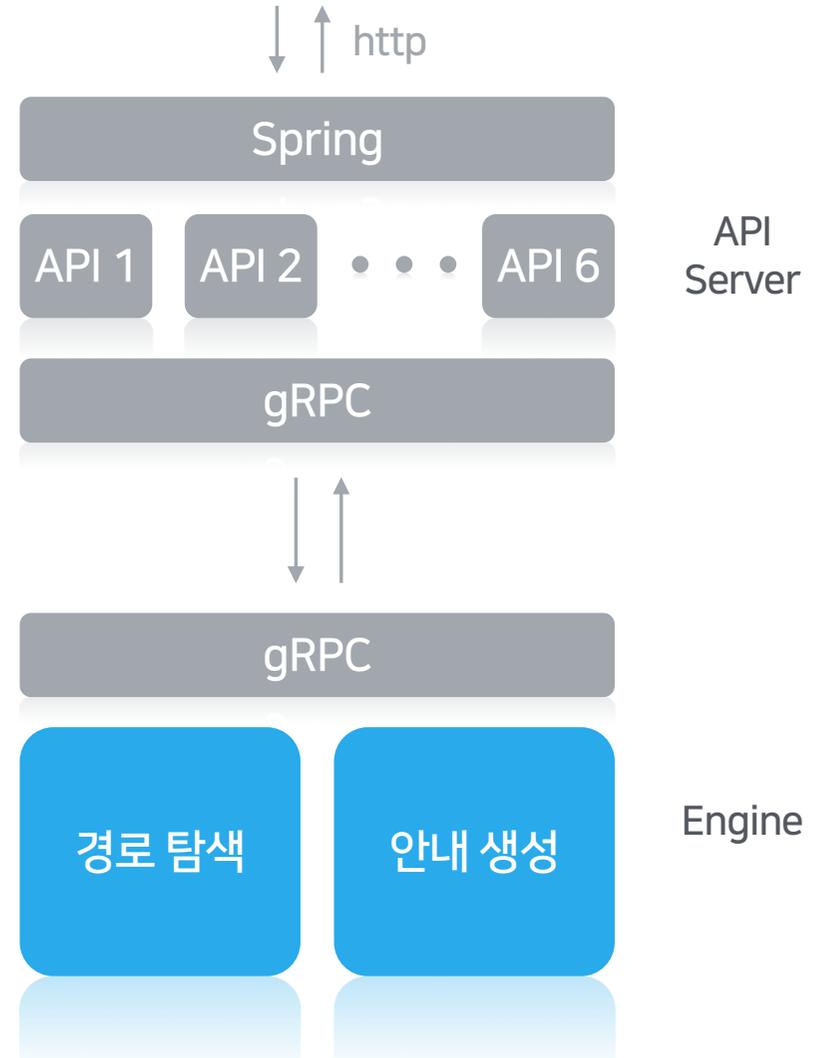
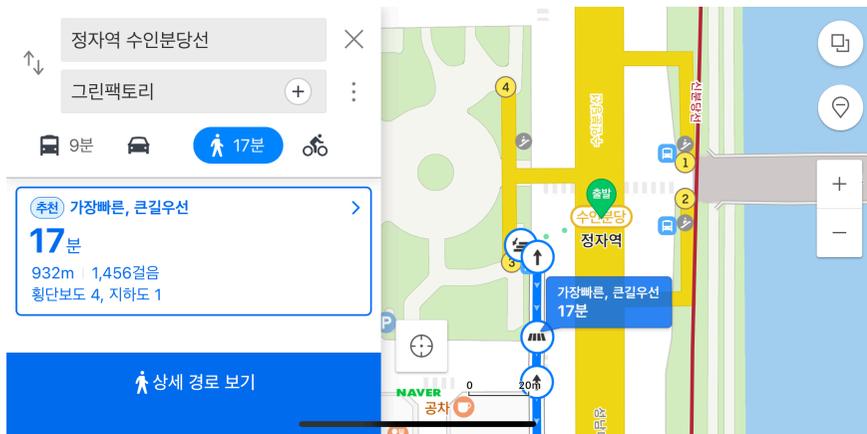
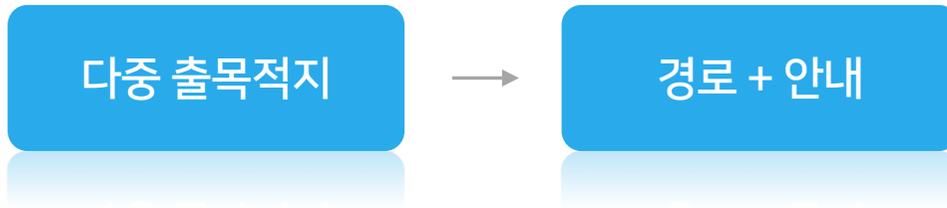
2.3 신규 아키텍처 - 서버



2.3 신규 아키텍처 - 서버



2.3 신규 아키텍처 - 서버



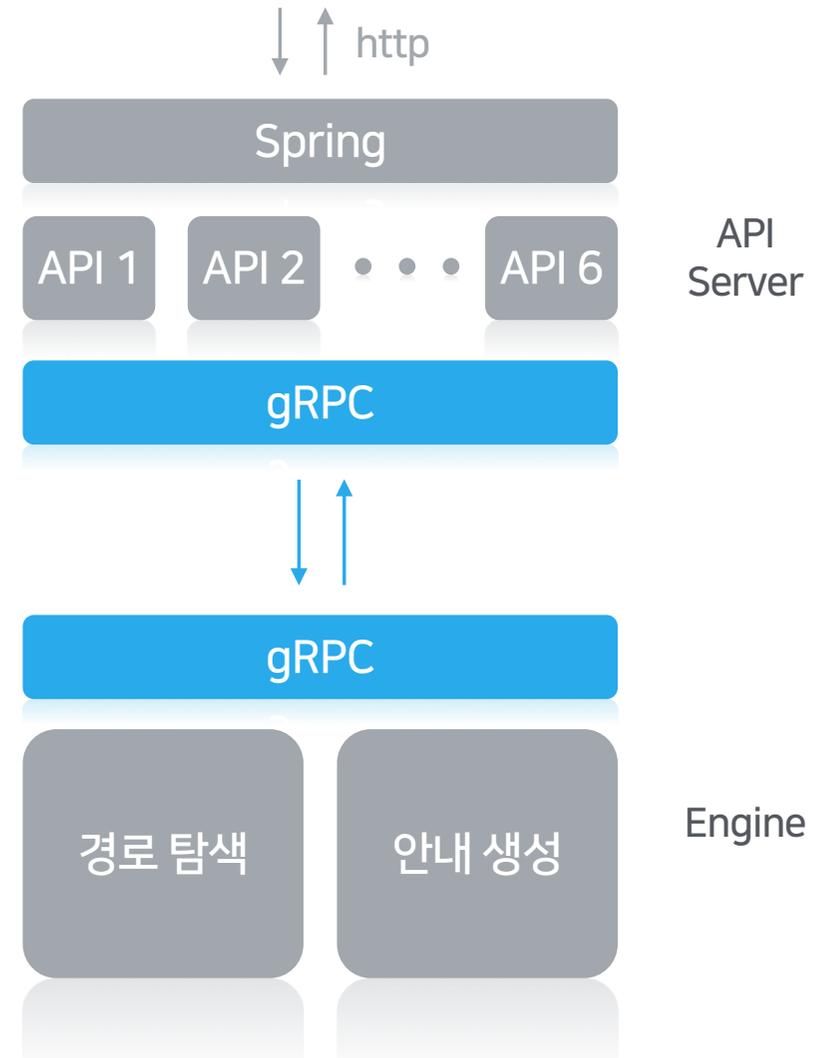
2.3 신규 아키텍처 - 서버



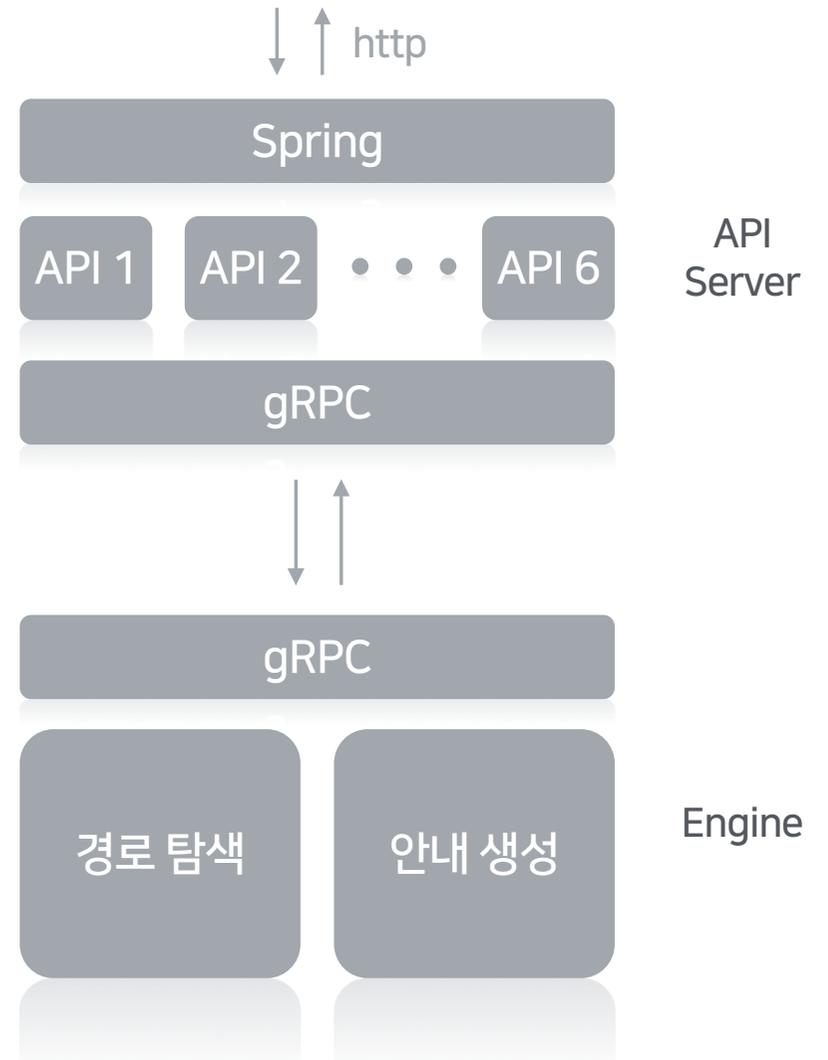
성능



유지보수



2.3 신규 아키텍처 - 서버



2.4 신규 아키텍처 - 엔진

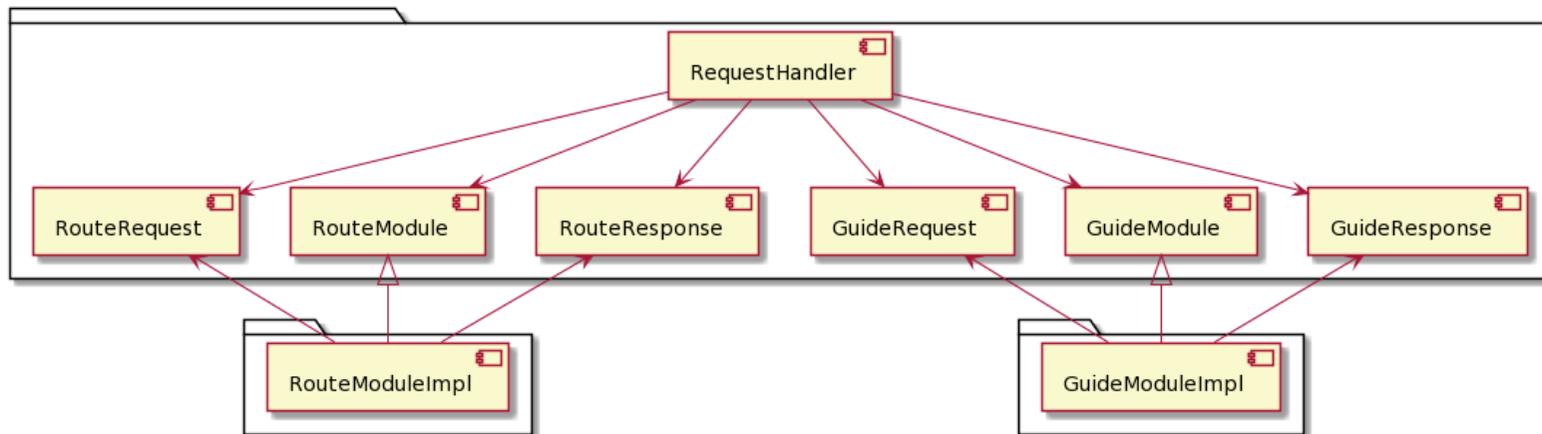
확장/수정
용이

수정량은
범위에
비례

하위
컴포넌트와
분리

상위
컴포넌트와
분리

2.4 신규 아키텍처 - 엔진



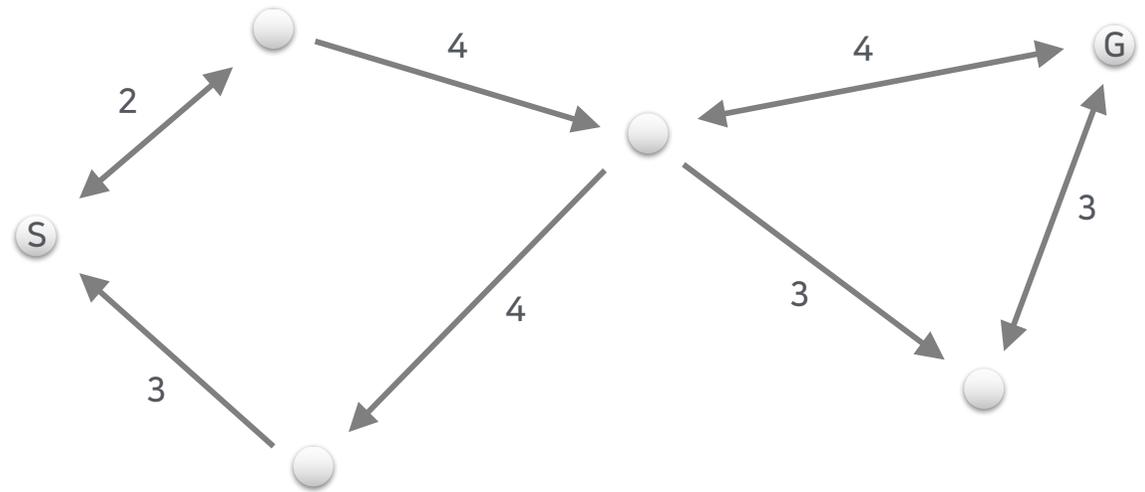
- 각 모듈의 인터페이스를 상위 컴포넌트가 관리한다.
- 모듈의 입력/출력 구조체도 상위 컴포넌트가 관리한다.

3. 도보 경로 탐색

3.1 좋은 경로 구하기

유형가중
그래프

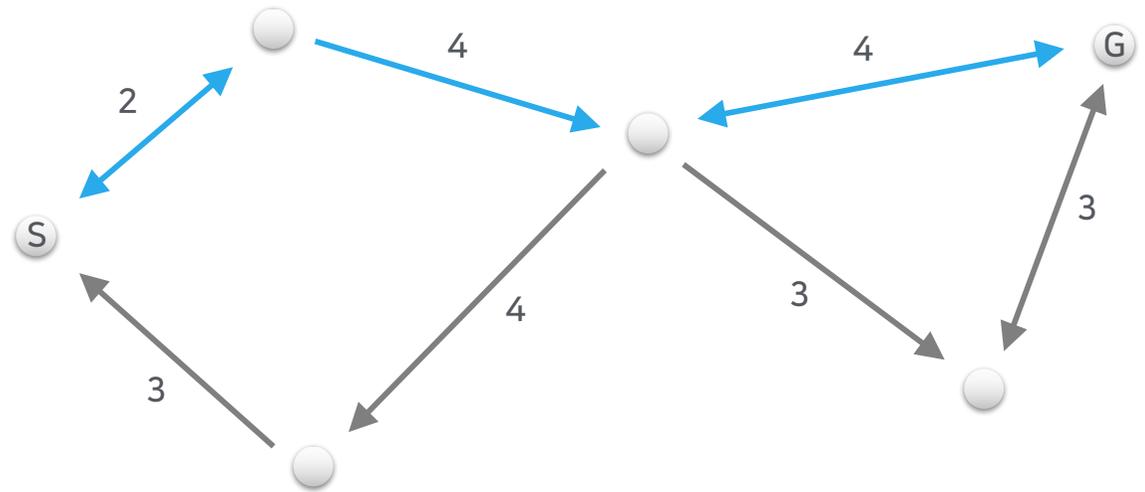
최단 경로



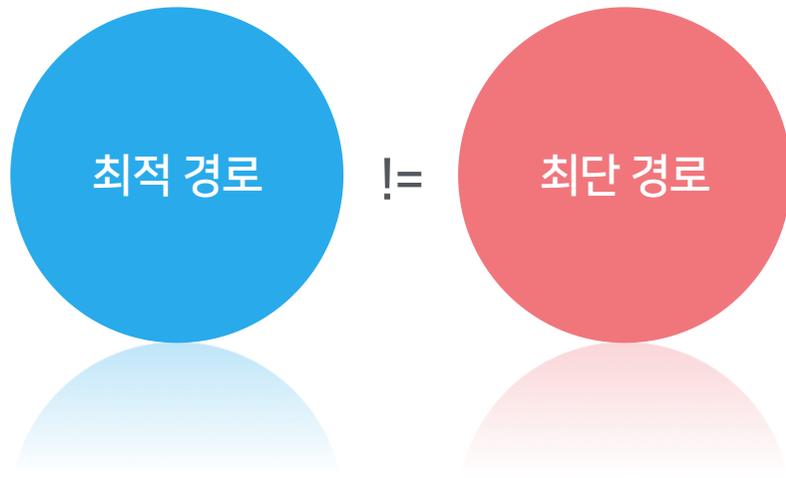
3.1 좋은 경로 구하기

유형 가중
그래프

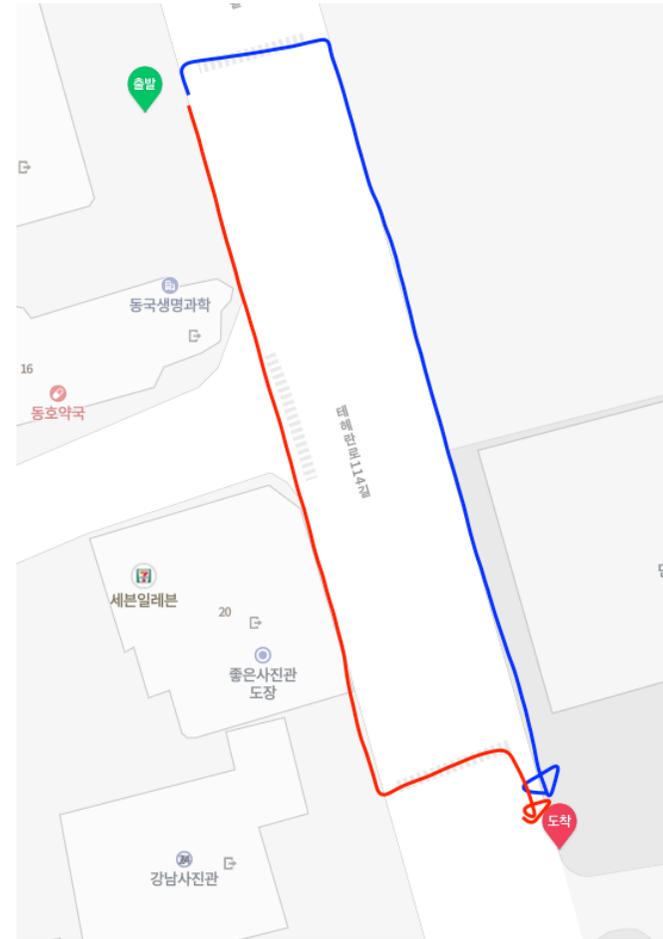
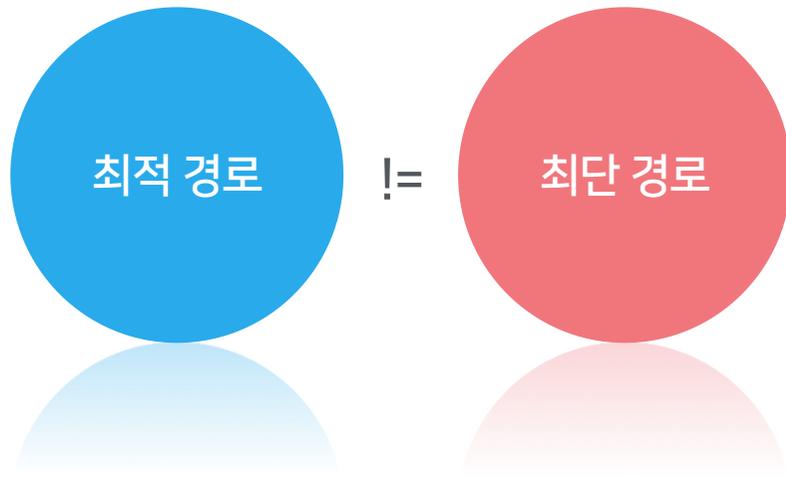
최단 경로



3.1 좋은 경로 구하기



3.1 좋은 경로 구하기



3.1 좋은 경로 구하기

도로 유형별
가중치

시설물 진입
페널티



3.1 좋은 경로 구하기

도로 유형별
가중치

시설물 진입
페널티



3.2 경로 탐색 모듈의 고려사항

도로 유형별
가중치 튜닝

시설물 진입
페널티 튜닝

다중 입구점
고려한 경로

3.2 경로 탐색 모듈의 고려사항

도로 유형별
가중치 튜닝

시설물 진입
페널티 튜닝

다중 입구점
고려한 경로

정적 데이터
사용

런타임 성능
(vs 전처리)

3.3 Contraction Hierarchies

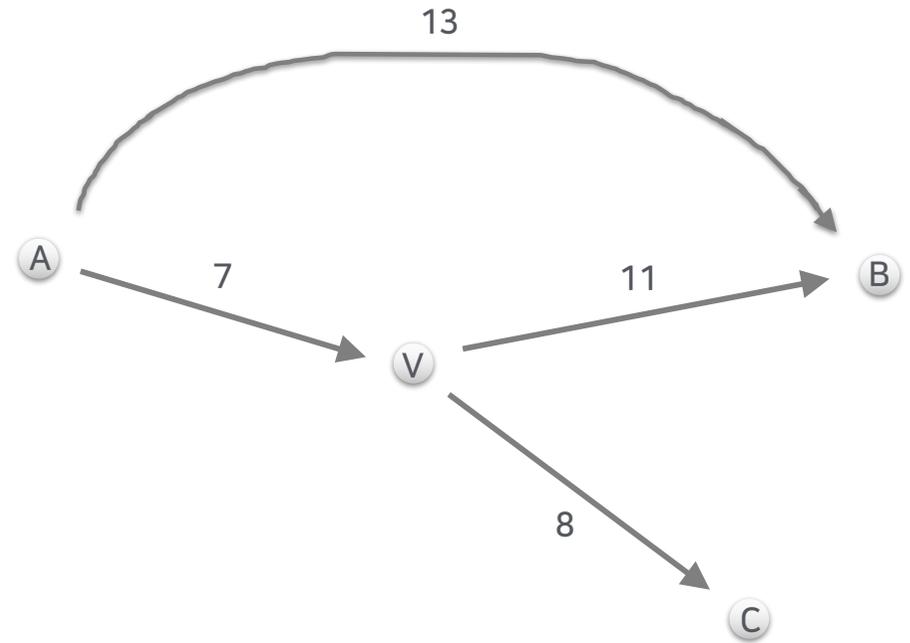
Dijkstra

A*

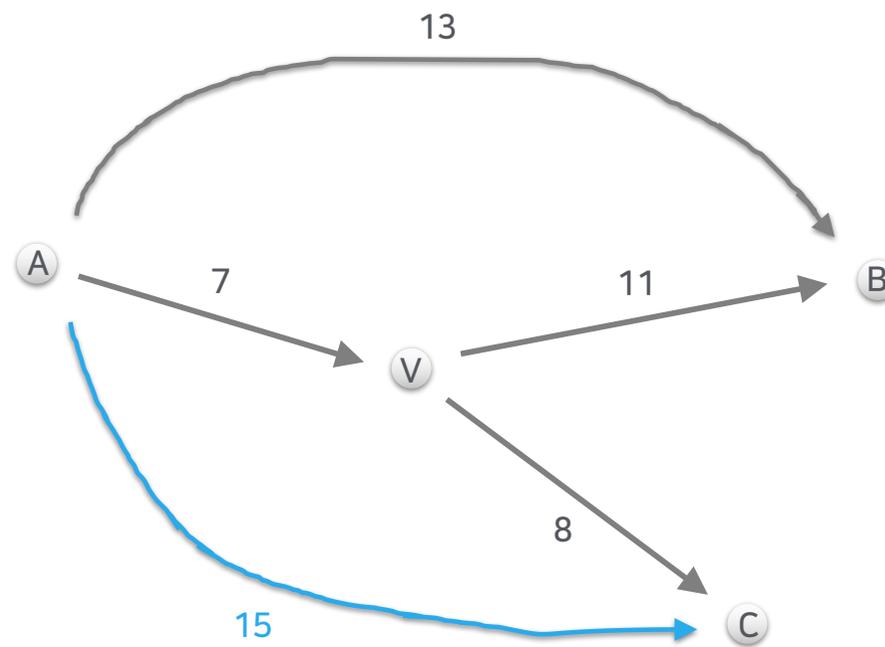
CRP (MLD)

Contraction
Hierarchies

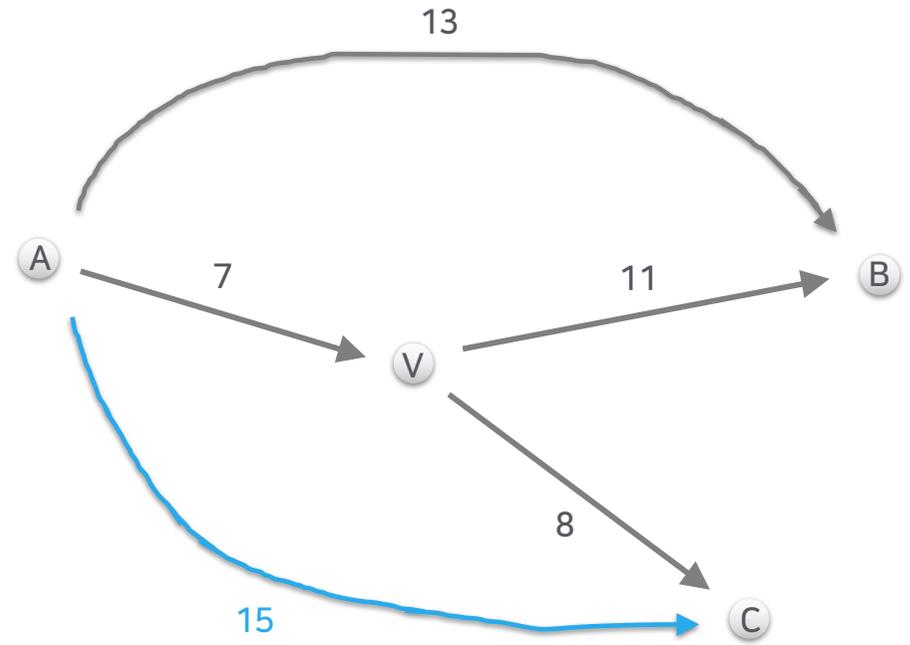
3.3 Contraction Hierarchies



3.3 Contraction Hierarchies

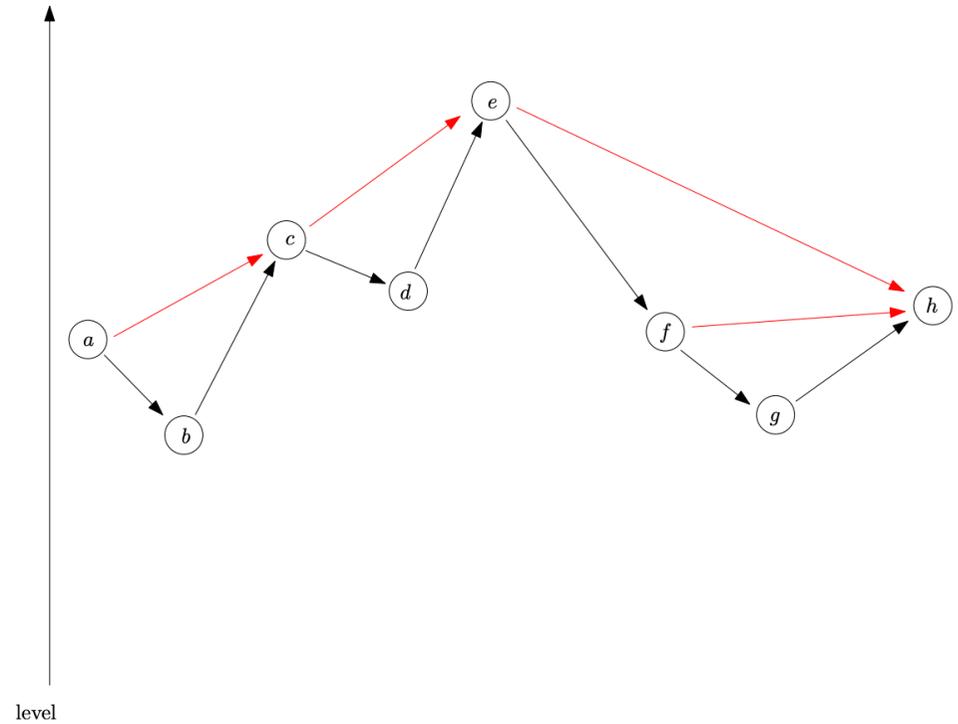
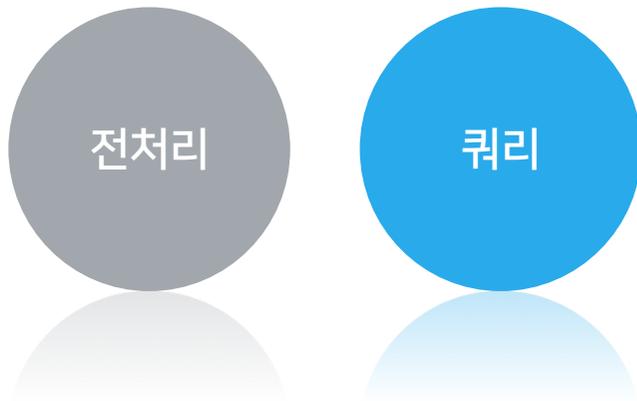


3.3 Contraction Hierarchies



- 각 노드에 level을 부여하고, level 순으로 처리

3.3 Contraction Hierarchies



Stefan Funke: "Contraction Hierarchies briefly explained" (2017)

3.4 오픈 소스 사용

다들 좋아하는 언어인가?

많이 쓰는가?

문서화가 잘 되어 있는가?

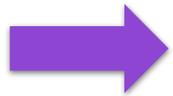
기능 지원 범위가 넓은가?

문서화

3.4 오픈 소스 사용

오픈 소스 (C++)	사용자 수	문서화	기능
Project-OSRM/osrm-backend	4k Stars 2.2k Forks	O	one-to-one many-to-many
vlarmet/cppRouting	49 Stars 4 Forks	△	one-to-one one-to-many
Robert Geisberger, KIT	?	X	-
Google MoNav	?	X	-

3.4 오픈 소스 사용



오픈 소스 (C++)	사용자 수	문서화	기능
	4k Stars 2.2k Forks	0	one-to-one many-to-many
vlarmet/cppRouting	49 Stars 4 Forks	△	one-to-one one-to-many
Robert Geisberger, KIT	?	X	-
Google MoNav	?	X	-

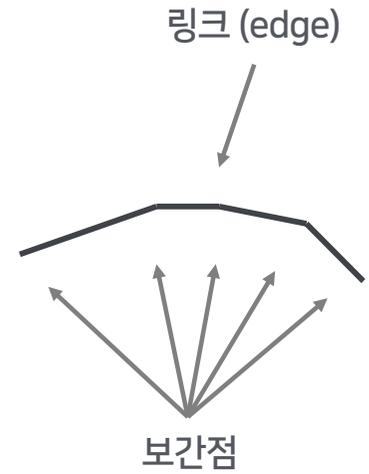
4. 오픈 소스 활용기

4.1 OSRM 도입

라이브러리

도로별
선호도
조절

보간점열
확보



4.2 오픈 소스와의 만남

OSRM은 기본적으로 자동차 routing machine.



도로 선호도
조절



신호등
대기시간



진입 페널티
부여

4.2 오픈 소스와의 만남

```
struct ExtractionTurn
{
    const double angle;
    const int number_of_roads;
    const bool is_u_turn;
    const bool has_traffic_light; // 이 값을 enum으로 바꿔 횡단보도, 엘리베이터 등을 처리해보자
    const bool is_left_hand_driving;

    // source info
    const bool source_restricted;
    const TravelMode source_mode;
    const bool source_is_motorway;
};
```

https://github.com/Project-OSRM/osrm-backend/blob/5.21/include/extractor/extraction_turn.hpp 일부 수정

4.2 오픈 소스와의 만남

```

struct ExtractionTurn
{
    const double angle;
    const int number_of_roads;
    const bool is_u_turn;
    const bool has_traffic_light; // 이 값을 enum으로 바꿔 횡단보도, 엘리베이터 등을 처리해보자
    const bool is_left_hand_driving;

    // source info
    const bool source_restricted;
    const TravelMode source_mode;
    const bool source_is_motorway;

```

https://github.com/Project-OSRM/osrm-backend/blob/5.21/include/extractor/extraction_turn.hpp 일부 수정

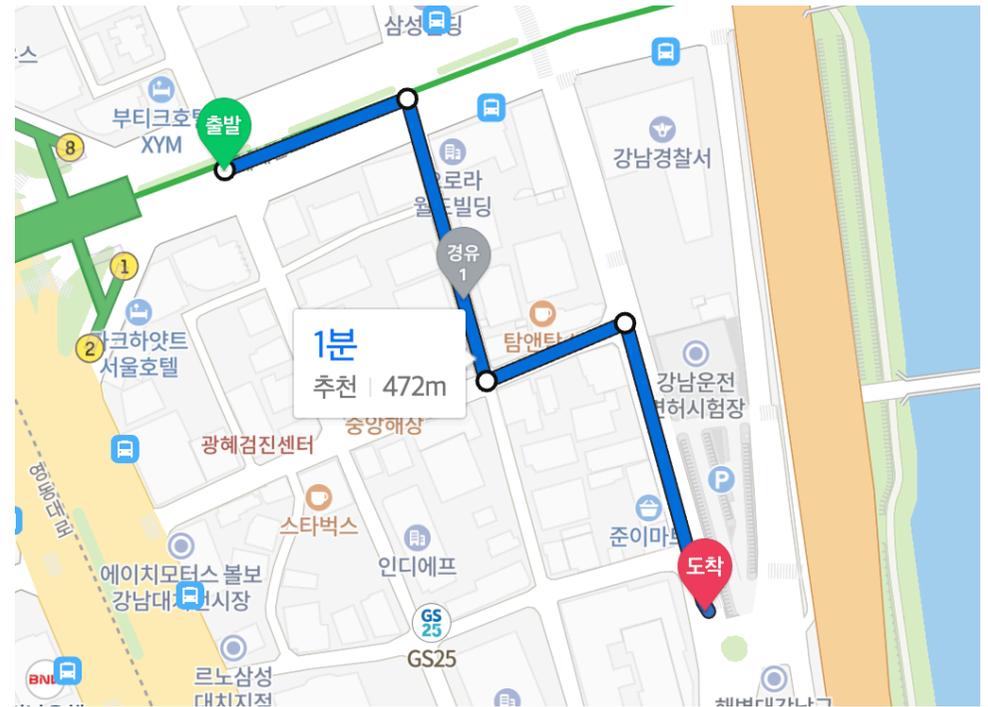
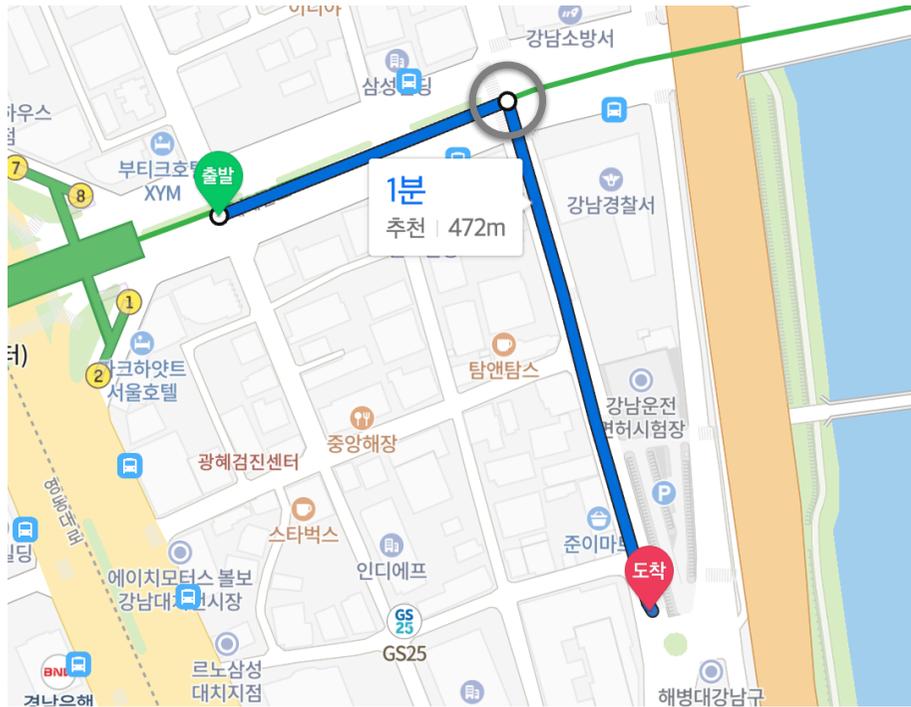
약 2000줄
수정 예상

2~3주
소요 예상

오픈 소스
버전업?

4.2 오픈 소스와의 만남

특정 도로에서 회전이 특히 더 어렵다면?



4.2 오픈 소스와의 만남

```
$ cat enter_penalty.csv
# from_vertex_id,via_vertex_id,to_vertex_id,penalty_in_secs

15002,15003,14002,30      # 대기시간 30초
21003,22001,22002,60    # 대기시간 60초
(...)
```

범용적으로
사용 가능

약 100줄
구현

작업에
하루 소요

4.2 오픈 소스와의 만남

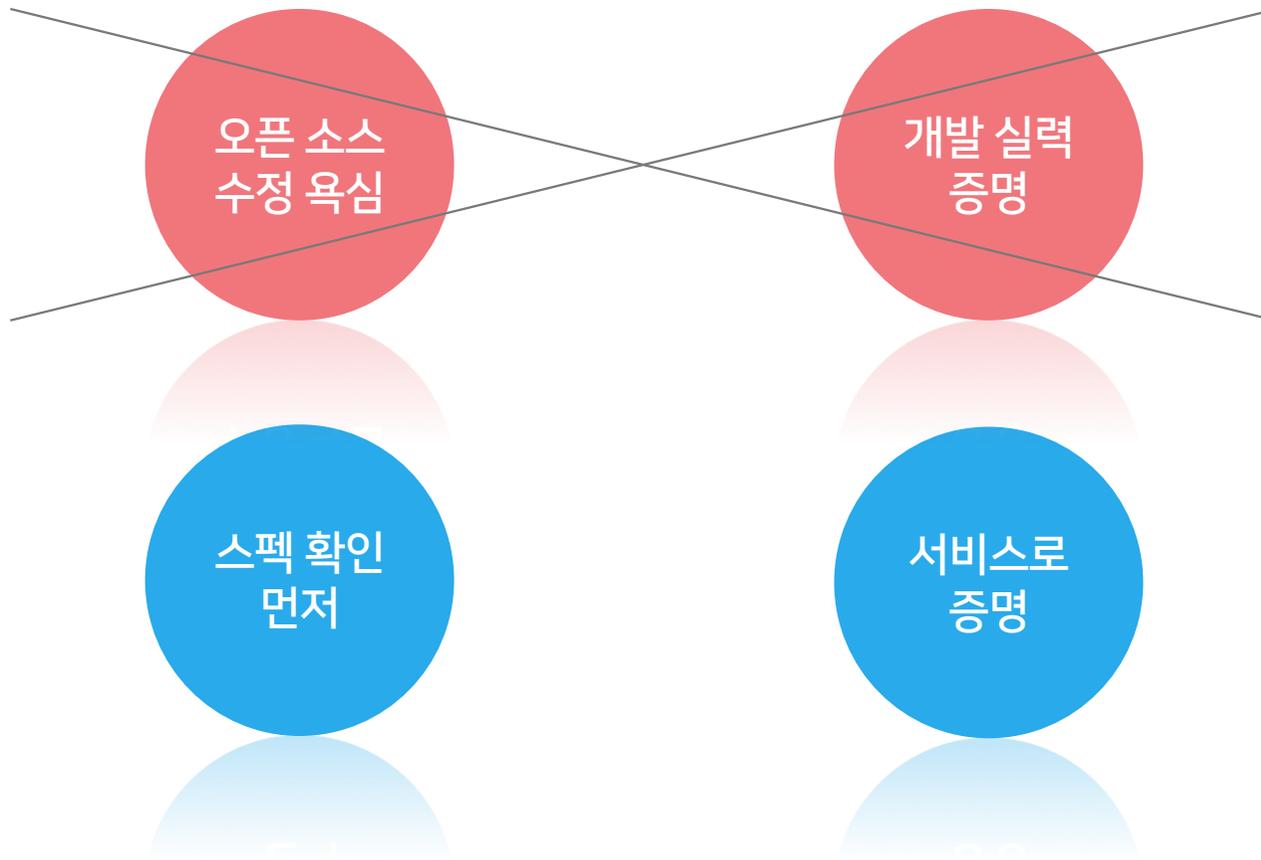


오픈 소스
수정 욕심



개발 실력
증명

4.2 오픈 소스와의 만남



4.3 오픈 소스 코드 뜯어보기

OSRM



경로 탐색 모듈

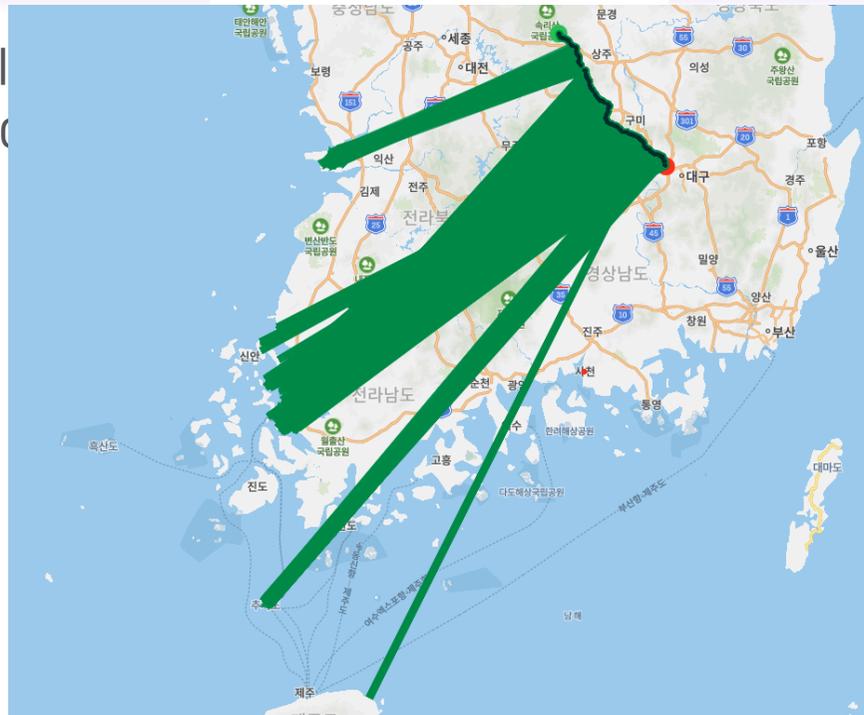
보간점 리스트
(ex. 151002)

보간점 id 변환
(ex. 151번째 링크 2번째 보간점)

4.3 오픈 소스 코드 뜯어보기



보간점 리
(ex. 1510



점 id 변환
(링크 2번째 보간점)

4.3 오픈 소스 코드 뜯어보기

스펙을 준수했는가?

- 변환용 파일, 함수 입출력 스펙 준수

에러에 공통점이 있는가?

- 보간점 id가 90억 이상에서 발생

내 코드에 문제가 있나?

- id의 값에 의존하는 로직 없음

4.3 오픈 소스 코드 뜯어보기

결론부터 말하자면,

```

  2 include/extractor/packed_osm_ids.hpp
  @@ -11,7 +11,7 @@ namespace extractor
  11 namespace detail
  12 {
  13 template <storage::Ownership Ownership>
  14 - using PackedOSMIDs = util::detail::PackedVector<OSMNodeID, 33, Ownership>;
  15 }
  16
  17 using PackedOSMIDsView = detail::PackedOSMIDs<storage::Ownership::View>;

  11 namespace detail
  12 {
  13 template <storage::Ownership Ownership>
  14 + using PackedOSMIDs = util::detail::PackedVector<OSMNodeID, 36, Ownership>;
  15 }
  16
  17 using PackedOSMIDsView = detail::PackedOSMIDs<storage::Ownership::View>;
  
```

4.3 오픈 소스 코드 뜯어보기

무엇을 준비할까?

어떻게 볼까?

4.3 오픈 소스 코드 뜯어보기



4.3 오픈 소스 코드 뜯어보기

무엇을 준비할까?

어떻게 볼까?

작업 절차 파악

클래스 내의/간의
구조 파악

4.3 오픈 소스 코드 뜯어보기

```
std::tie( &: turn_lane_map,  
         &: turn_restrictions,  
         &: conditional_turn_restrictions,  
         &: unresolved_maneuver_overrides) =  
    ParseOSMData( &: scripting_environment, number_of_threads);  
  
// Create a node-based graph from the OSRM file  
NodeBasedGraphFactory node_based_graph_factory( input_file: config.GetPath( fileName: ".osrm"),  
                                                &: scripting_environment,  
                                                &: turn_restrictions,  
                                                &: conditional_turn_restrictions,  
                                                &: unresolved_maneuver_overrides);  
  
NameTable name_table;  
files::readNames( path: config.GetPath( fileName: ".osrm.names"), &: name_table);  
  
auto const &coordinates = node_based_graph_factory.GetCoordinates();  
files::writeNodes(  
    path: config.GetPath( fileName: ".osrm.nbg_nodes"), coordinates, node_based_graph_factory.GetOsmNodes());  
node_based_graph_factory.ReleaseOsmNodes();  
  
auto const &node_based_graph = node_based_graph_factory.GetGraph();  
  
files::writeCompressedNodeBasedGraph( path: config.GetPath( fileName: ".osrm.cnbg").string(),  
                                     toEdgeList(node_based_graph));
```

<https://github.com/Project-OSRM/osrm-backend/blob/5.21/src/extractor/extractor.cpp> 일부 수정

4.3 오픈 소스 코드 뜯어보기

```

class NodeBasedGraphFactory
{
public:
    NodeBasedGraphFactory(const boost::filesystem::path &input_file,
                          ScriptingEnvironment &scripting_environment,
                          std::vector<TurnRestriction> &turn_restrictions,
                          std::vector<ConditionalTurnRestriction> &conditional_turn_restrictions,
                          std::vector<UnresolvedManeuverOverride> &maneuver_overrides);

    auto const &GetGraph() const { return compressed_output_graph; }
    auto const &GetBarriers() const { return barriers; }
    auto const &GetTrafficSignals() const { return traffic_signals; }
    auto const &GetCompressedEdges() const { return compressed_edge_container; }
    auto const &GetCoordinates() const { return coordinates; }
    auto &GetAnnotationData() { return annotation_data; }
    auto &GetOsmNodes() { return osm_node_ids; }
    // 일부 생략

private:
    void LoadDataFromFile(const boost::filesystem::path &input_file);

    void Compress(ScriptingEnvironment &scripting_environment,
                 std::vector<TurnRestriction> &turn_restrictions,
                 std::vector<ConditionalTurnRestriction> &conditional_turn_restrictions,
                 std::vector<UnresolvedManeuverOverride> &maneuver_overrides);

    void CompressGeometry();
    void CompressAnnotationData();

    // 일부 생략
}

```

https://github.com/Project-OSRM/osrm-backend/blob/5.21/include/extractor/node_based_graph_factory.hpp 일부 수정

4.3 오픈 소스 코드 뜯어보기

```
NodeBasedGraphFactory::NodeBasedGraphFactory(  
    const boost::filesystem::path &input_file,  
    ScriptingEnvironment &scripting_environment,  
    std::vector<TurnRestriction> &turn_restrictions,  
    std::vector<ConditionalTurnRestriction> &conditional_turn_restrictions,  
    std::vector<UnresolvedManeuverOverride> &maneuver_overrides)  
{  
    LoadDataFromFile(input_file);  
    Compress( & scripting_environment,  
             & turn_restrictions,  
             & conditional_turn_restrictions,  
             & maneuver_overrides);  
    CompressGeometry();  
    CompressAnnotationData();  
}
```

https://github.com/Project-OSRM/osrm-backend/blob/5.21/src/extractor/node_based_graph_factory.cpp

4.3 오픈 소스 코드 뜯어보기

```
NodeBasedGraphFactory::NodeBasedGraphFactory(  
    const boost::filesystem::path &input_file,  
    ScriptingEnvironment &scripting_environment,  
    std::vector<TurnRestriction> &turn_restrictions,  
    std::vector<ConditionalTurnRestriction> &conditional_turn_restrictions,  
    std::vector<UnresolvedManeuverOverride> &maneuver_overrides)  
{  
    LoadDataFromFile(input_file);  
    Compress( &: scripting_environment,  
             &: turn_restrictions,  
             &: conditional_turn_restrictions,  
             &: maneuver_overrides);  
    CompressGeometry();  
    CompressAnnotationData();  
}
```

https://github.com/Project-OSRM/osrm-backend/blob/5.21/src/extractor/node_based_graph_factory.cpp

전체 흐름 파악?

세세한 로직 파악?

4.3 오픈 소스 코드 뜯어보기

세 사람이 함께 걸어가면, 반드시 나의 스승이 있다.

그 중 나보다 나은 이의 좋은 점을 배우고, 그렇지 않은 이의 좋지 않은 점을 거울삼아 고치도록 한다.

- 공자

4.3 오픈 소스 코드 뜯어보기

세 사람이 함께 걸어가면, 반드시 나의 스승이 있다.

그 중 나보다 나은 이의 좋은 점을 배우고, 그렇지 않은 이의 좋지 않은 점을 거울삼아 고치도록 한다.

- 공자

오픈 소스에서
배울 점

반면교사로
삼을 점

4.3 오픈 소스 코드 뜯어보기

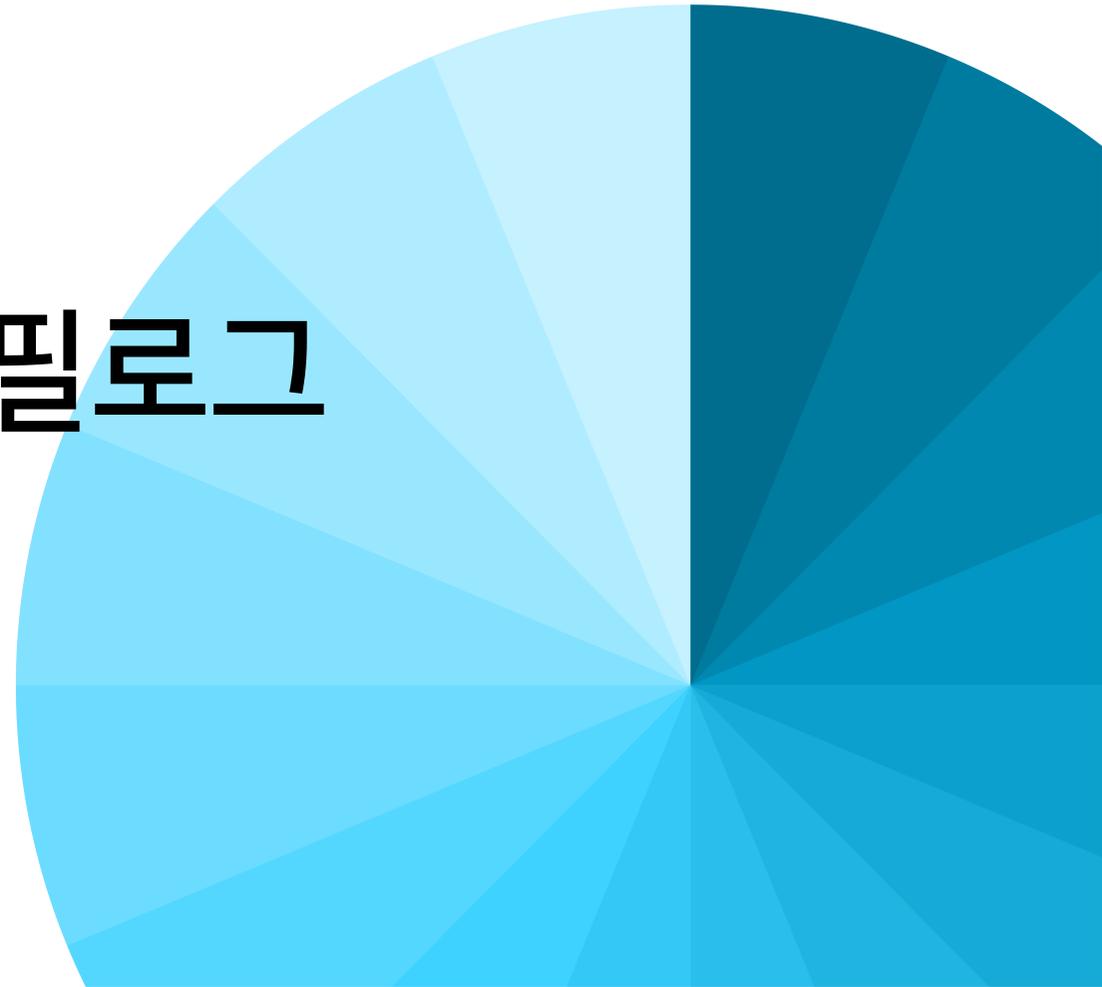
세 사람이 함께 걸어가면, 반드시 나의 스승이 있다.

그 중 나보다 나은 이의 좋은 점을 배우고, 그렇지 않은 이의 좋지 않은 점을 거울삼아 고치도록 한다.

- 공자



5. 에필로그



5.1 그동안 한 이야기

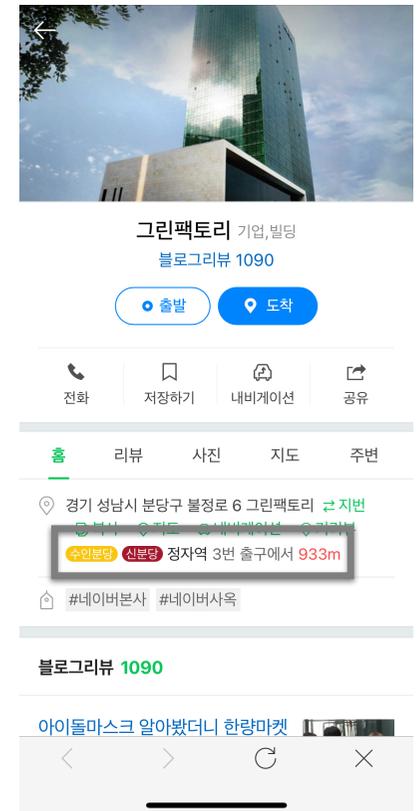
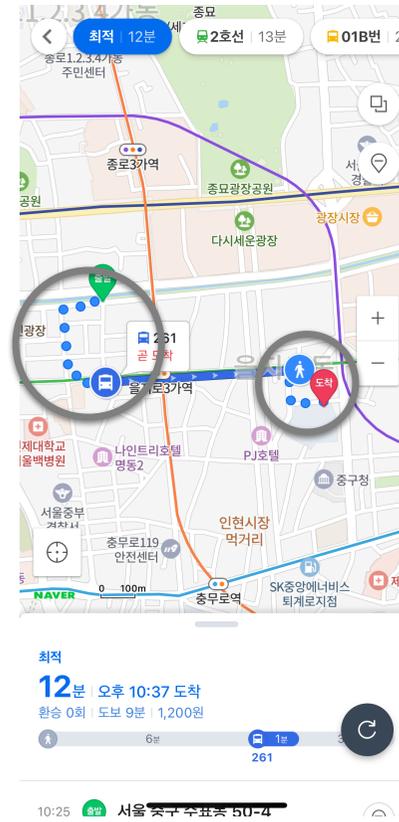
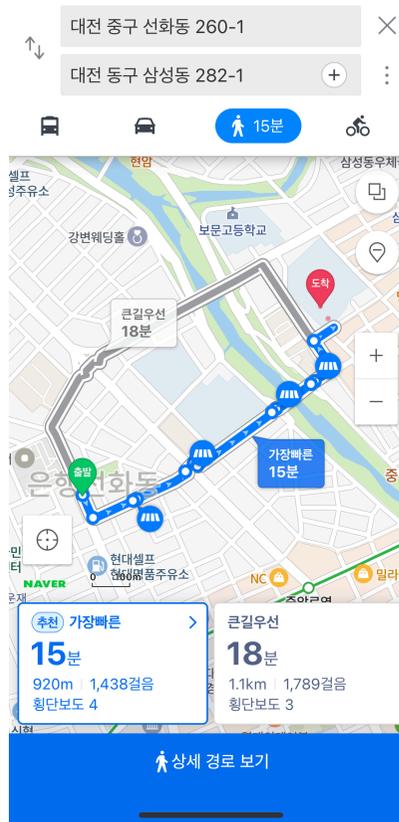


5.2 실제 서비스 적용

도보/대중교통 길찾기

플레이스 정보

일일 평균 1억+ 쿼리



5.3 앞으로의 과제

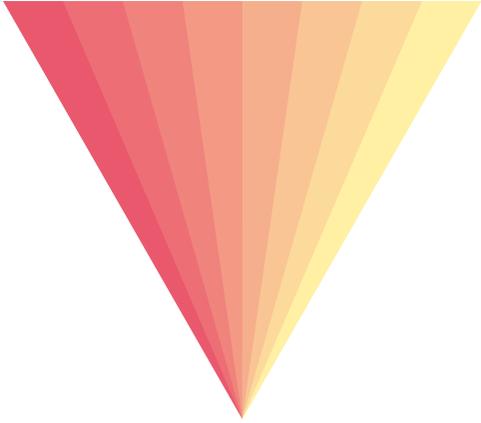
경로 품질
향상

고도 데이터
적용

안내 고도화



Thank You



Q & A

