

LLM을 활용한 코드 자동 생성 및 변환 프로그램 개발

컴퓨터공학부
박현 배준익



Contents

Overview

Goal/Problem & Requirement

Approach

Development Environment

Architecture

Implementation Spec

Results

Demo

Division and Assignment of Work

Overview

LLM을 이용한 코드 자동 변환 프로그램 개발



Proworks

- 자바 백엔드 프레임워크 유틸리티를 제공하는 라이브러리

Proworks 4: Map 기반 구현

- 데이터베이스 쿼리 및 함수 파라미터로 Map 자료구조를 이용
- 동적 자료형: Architecture 복잡도 및 유지보수 수고 증가

Proworks 5: DataClass (Value Object) 기반의 구현

- DB 및 함수 파라미터로 데이터클래스를 활용
- 정적 자료형: 모던한 디자인 패턴 및 Proworks 5의 기능을 최대한 활용 가능
- 추가로, 여러 유틸리티 API의 업데이트 존재

Legacy code

```
public Map selectEmp(int empNo) {...};  
public Map updateEmp(Map data) {...};  
public Map selectUpdateEmp(int empNo) {...};  
...
```

Converted code

```
public EmpVO selectUpdateEmp(int EmpNo);  
public EmpVO selectEmp(int empNo);  
public EmpVO updateEmp(EmpVO data);
```

Goal & Requirements

Goal & Requirements

ProWorks4 → ProWorks5 코드 변환의 완전 자동화

(VO-based architecture, API 변경에 대응 가능한 코드 변환기)

- 다양한 케이스의 ProWorks 4 코드를 변환 가능한 **방법론 구상**
- 방법론 구현 및 평가

Constraints

- Local LLM만 활용 가능
- LLM 학습 없이 진행

Legacy code

```
public Map selectEmp(int empNo) {...};  
public Map updateEmp(Map data) {...};  
public Map selectUpdateEmp(int empNo) {...};  
...
```

Converted code

```
public EmpVO selectUpdateEmp(int EmpNo);  
public EmpVO selectEmp(int empNo);  
public EmpVO updateEmp(EmpVO data);
```

Approach

1. VO 생성

레거시 코드를 분석하여 ProWorks 5 코드 작성에 필요한 VO 클래스 생성

2. 코드 변환

생성된 VO 및 추가적인 context 활용
Self-feedback & Refinement

3. RAG

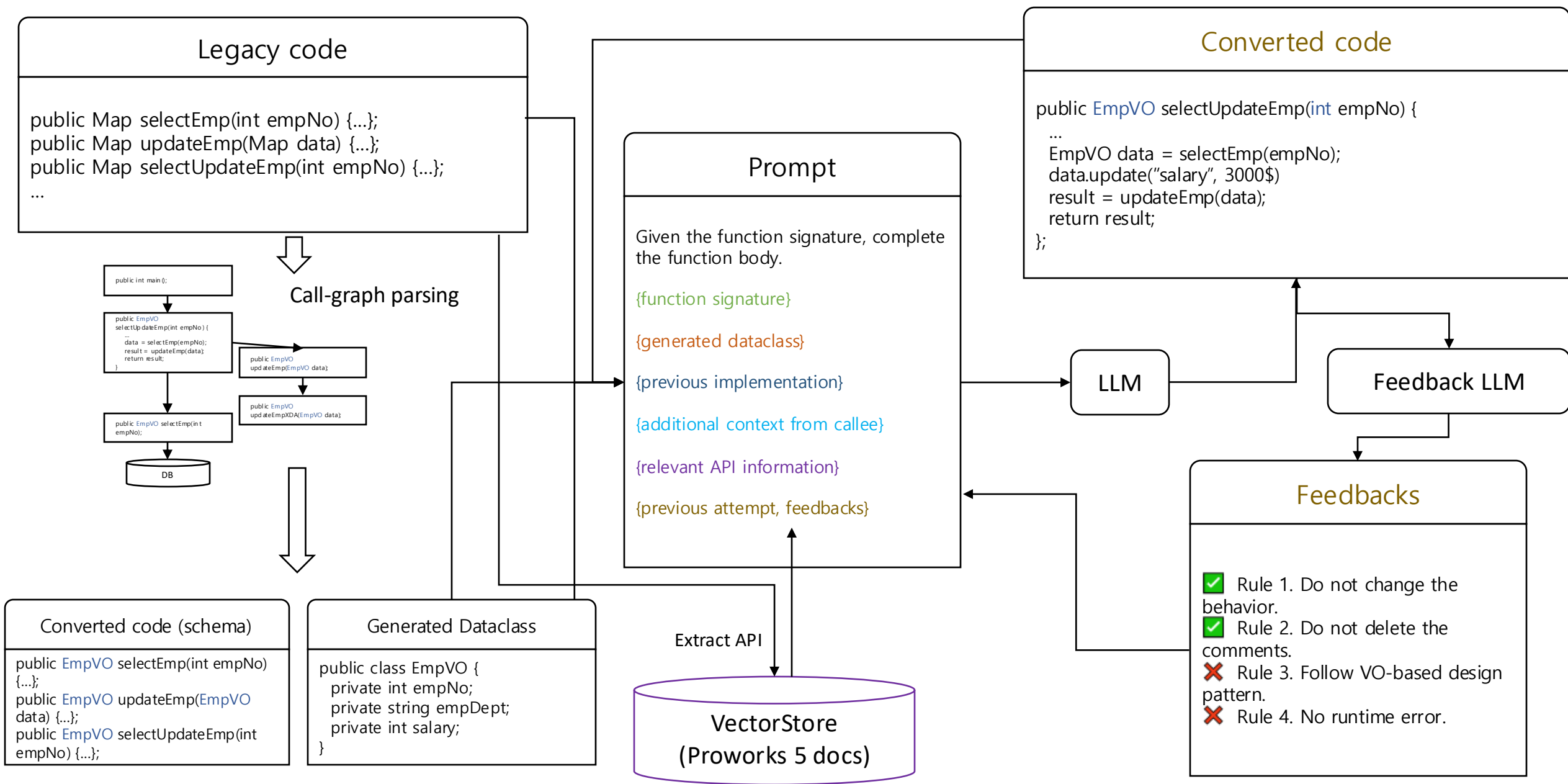
Proworks 5 API Documentation의 정보와 예시를 검색

4. (Programmable Converter - tried)

Development Environment

- Java 1.8, Spring-based backend framework (Proworks 4 / 5)
- 1 unit of A100 GPU (VRAM: 40GB)
- Python 3.10 / ollama

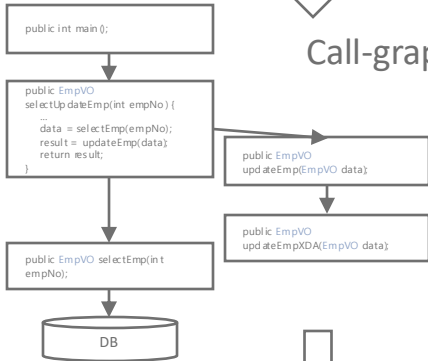
Architecture



Legacy code

```
public Map selectEmp(int empNo) {...};
public Map updateEmp(Map data) {...};
public Map selectUpdateEmp(int empNo) {...};
...
```

Call-graph parsing



VO Generation

Converted code (schema)

```
public EmpVO selectEmp(int empNo) {...};
public EmpVO updateEmp(EmpVO data) {...};
public EmpVO selectUpdateEmp(int empNo) {...};
```

Generated Dataclass

```
public class EmpVO {
    private int empNo;
    private string empDept;
    private int salary;
}
```

Prompt

Given the function signature, complete the function body.

{function signature}
{generated dataclass}
{previous implementation}
{additional context from callee}
{relevant API information}
{previous attempt, feedbacks}

Extract API

RAG

VectorStore
(Proworks 5 docs)

Converted code

```
public EmpVO selectUpdateEmp(int empNo) {
    ...
    EmpVO data = selectEmp(empNo);
    data.update("salary", 3000$);
    result = updateEmp(data);
    return result;
};
```

Self-Feedback

Feedback LLM

Feedbacks

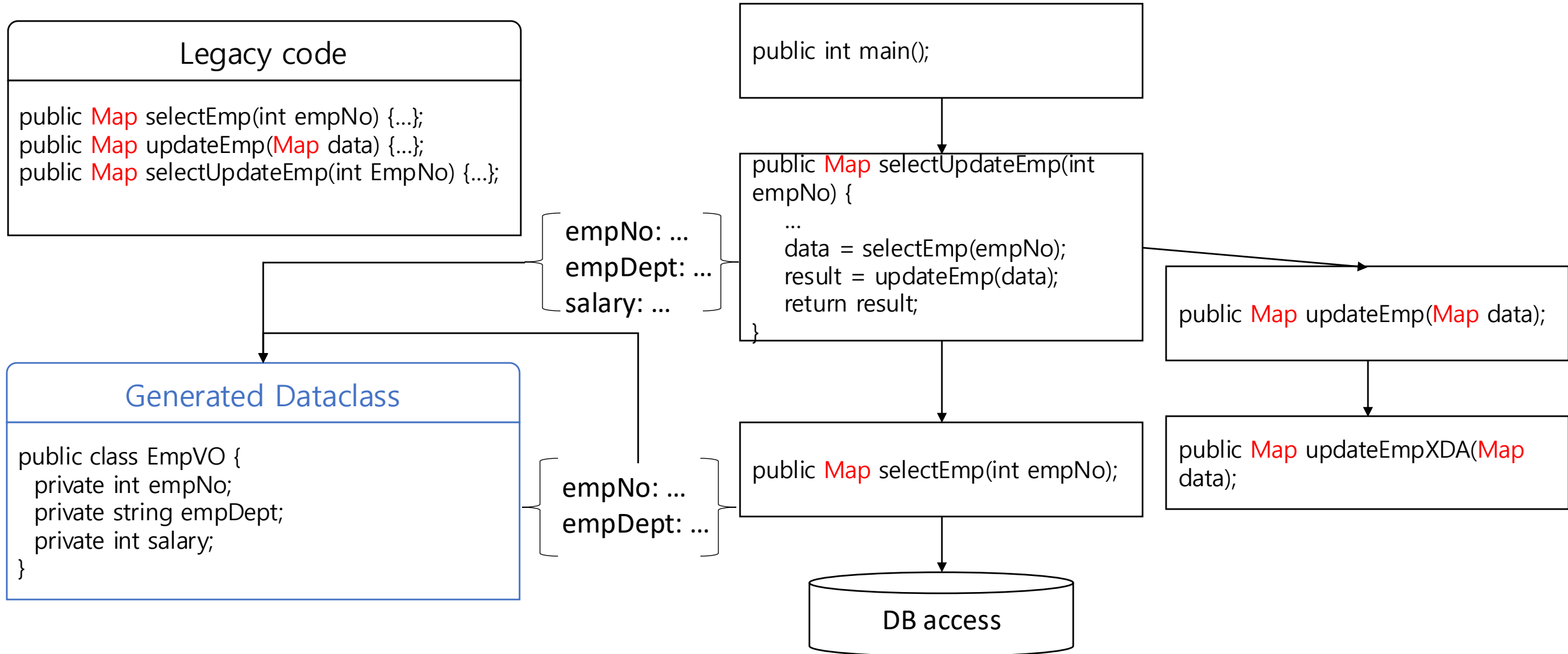
- ✅ Rule 1. Do not change the behavior.
- ✅ Rule 2. Do not delete the comments.
- ❌ Rule 3. Follow VO-based design pattern.
- ❌ Rule 4. No runtime error.

Implementation Spec

VO Generation

Goal: Create VO (DTO) Class

1. Map을 파싱하여 vo 클래스 생성



VO Generation

Goal: Create VO (DTO) Class

1. Map을 파싱하여 vo 클래스 생성
2. VO 클래스를 이용하도록 함수 시그니처 변환

Legacy code

```
public Map selectEmp(int empNo) {...};  
public Map updateEmp(Map data) {...};  
public Map selectUpdateEmp(int EmpNo) {...};
```

empNo: ...
empDept: ...
salary: ...

Generated Dataclass

```
public class EmpVO {  
    private int empNo;  
    private string empDept;  
    private int salary;  
}
```

empNo: ...
empDept: ...

```
public int main();
```

```
public EmpVO selectUpdateEmp(int empNo) {  
    ...  
    data = selectEmp(empNo);  
    result = updateEmp(data);  
    return result;  
}
```

```
public EmpVO selectEmp(int empNo);
```

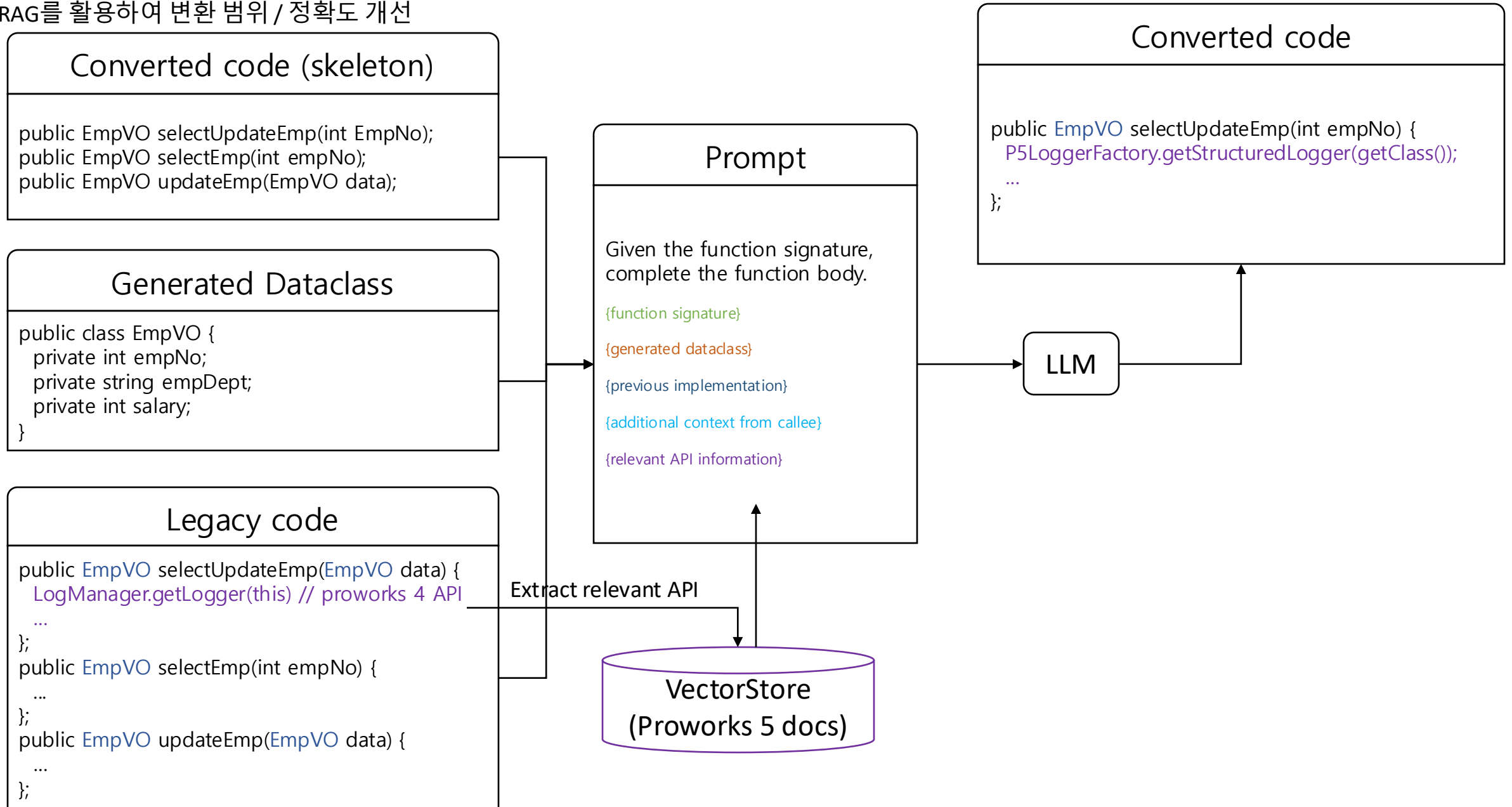
DB access

```
public EmpVO updateEmp(EmpVO data);
```

```
public EmpVO  
updateEmpXDA(EmpVO data);
```

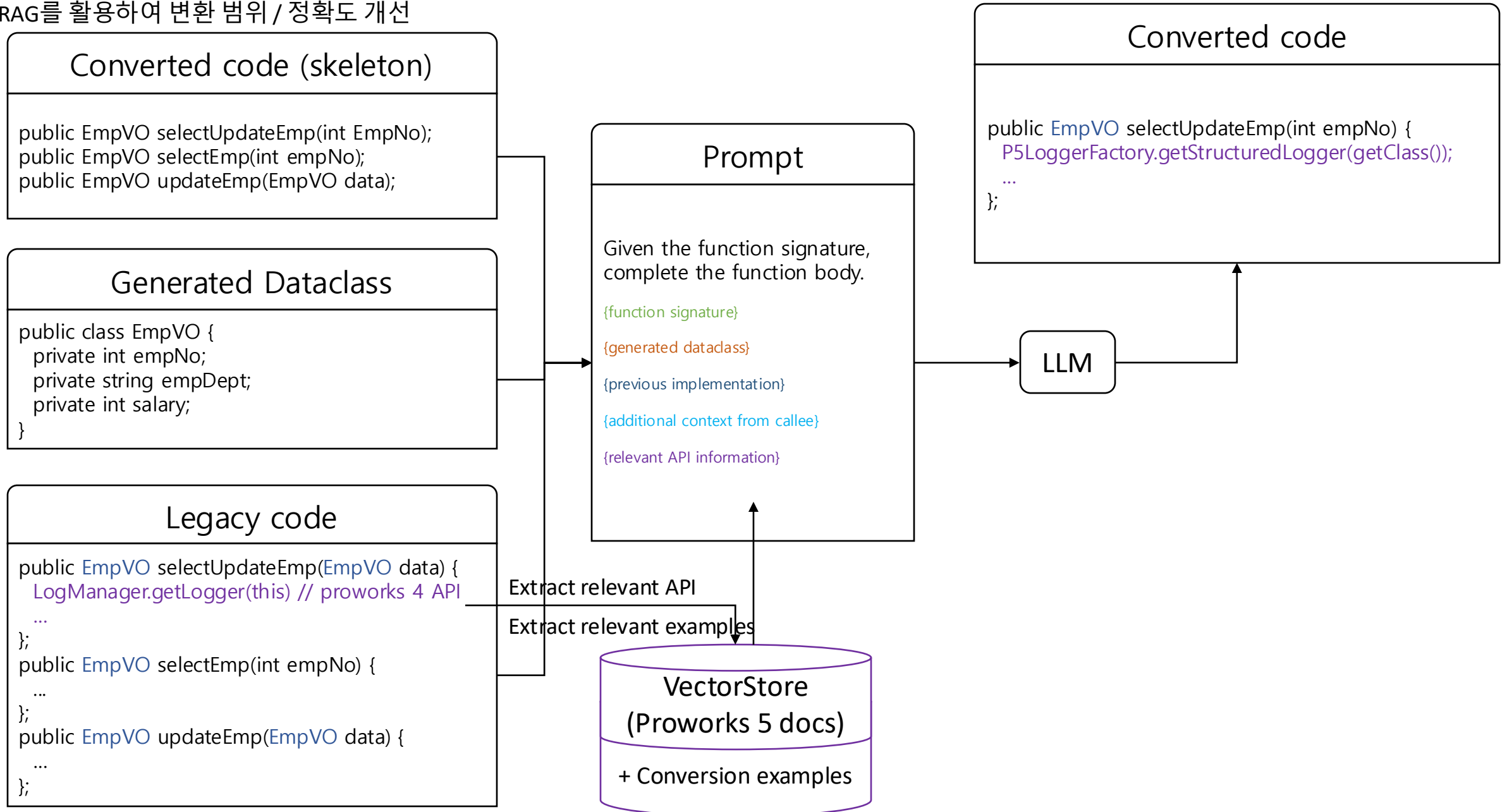
LLM Code Conversion with RAG

RAG를 활용하여 변환 범위 / 정확도 개선



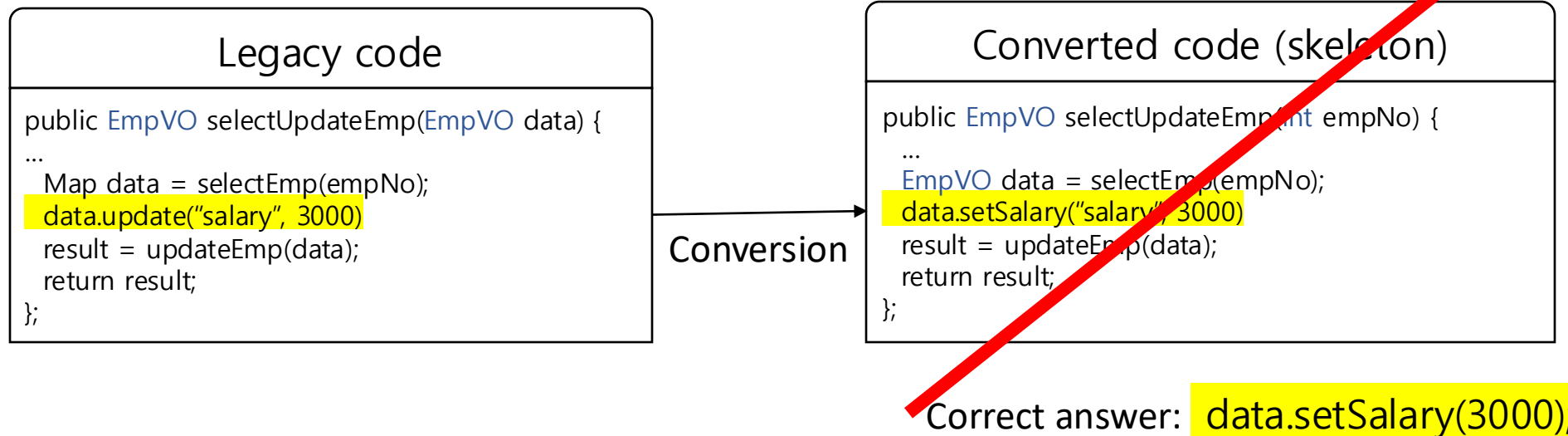
LLM Code Conversion with RAG

RAG를 활용하여 변환 범위 / 정확도 개선



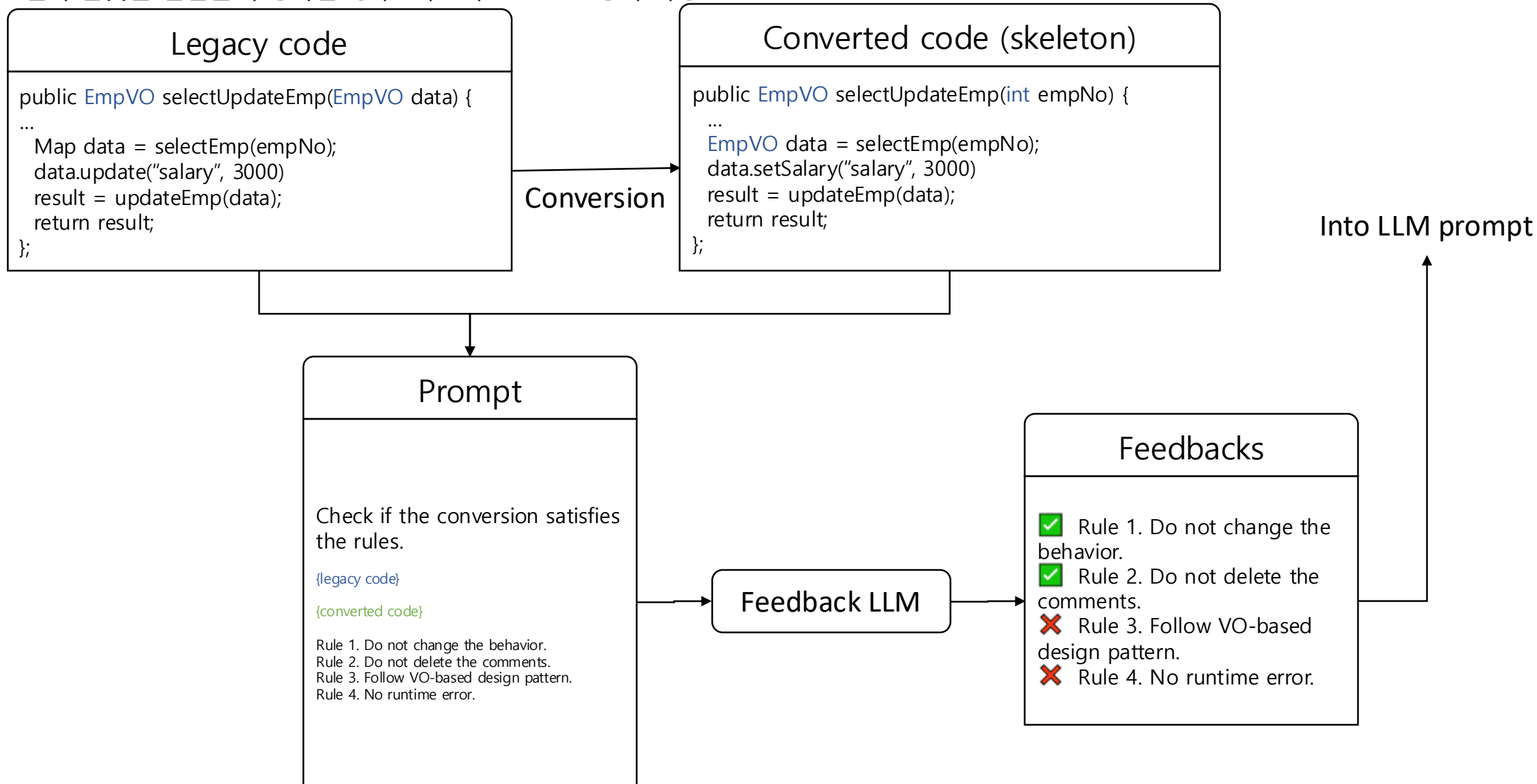
Self-Feedback and Refinement Module

모델이 잘못된 변환을 수행하는 경우

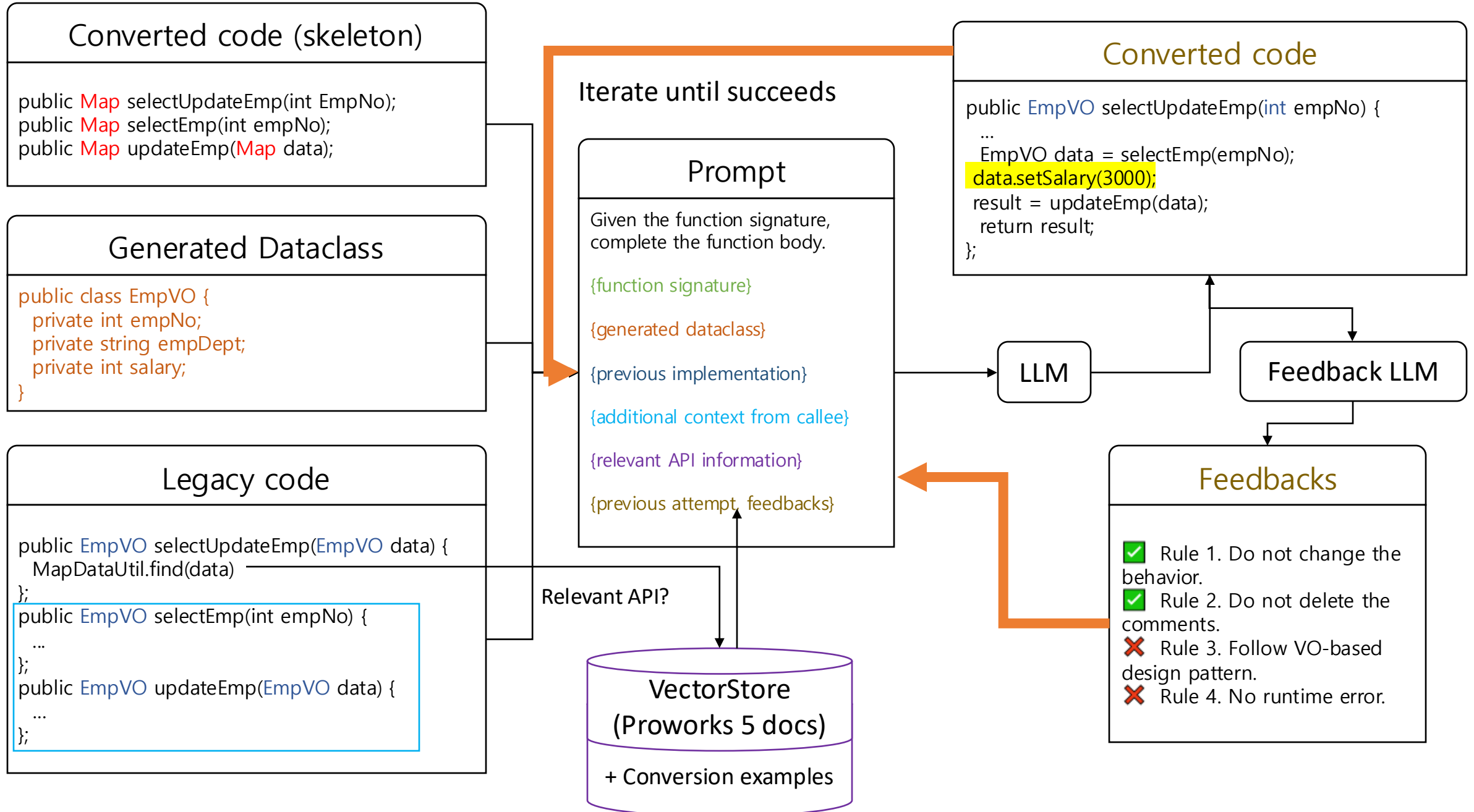


Self-Feedback and Refinement Module

모델이 잘못된 변환을 수행하는 경우: 피드백 LLM으로 평가 가능



Self-Feedback and Refinement Module



Evaluation Data

- Evaluation Dataset
 - 4 Categories from real-use cases
 - VO Generation
 - API Calls
 - VO getter/setter
 - Types: Map to VO
 - Edge cases:
 - nested map conversion
 - Comments
 - bad formatting
 - 10~100 examples for each
- Metric: Unit test correctness

Evaluation (Quantitative)

VO Generation (Acc)	
LLM (GPT-4.1)	0.5
LLM (devstral:24b)	0.0
Ours (Dependency analysis)	1.0

	Avg	API calls	VO get/set	Type Changes	Edge cases (Bad spacing, nested maps, comments etc)
Base	0.54	0.02	0.75	0.78	0.60
Base + RAG	0.91	0.98	0.94	0.97	0.75
Base + Reflexion	0.69	0.09	0.92	0.81	0.94
Base + RAG + Reflexion	0.99	1.00	1.00	1.00	0.98

*Reflexion: Our implementation of (Shinn et al., NIPS 2024).

LLM Model Comparison

Full pipeline
(Base + RAG +
Reflexion)

Avg Score

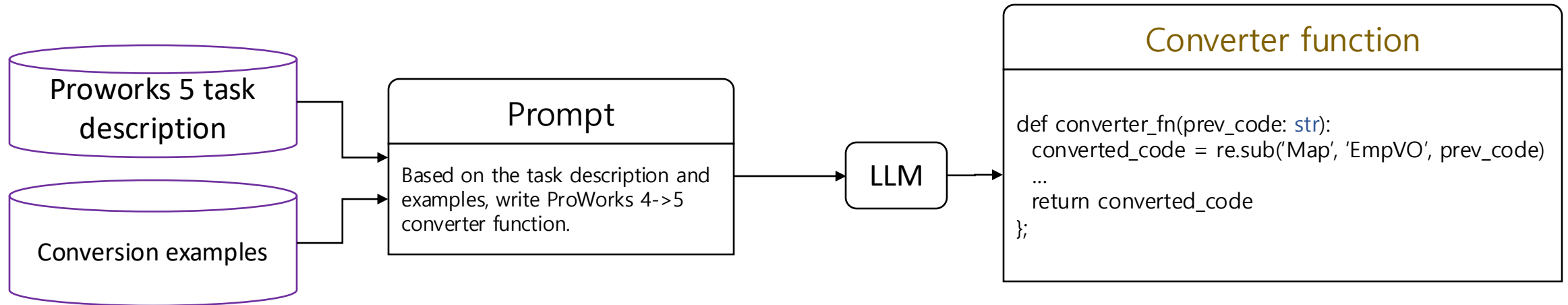
llama3.2	0.67
llama3.1:8b	0.94
phi-4	0.96
qwen2.5-coder:14b	0.96
qwen2.5-coder:7b	0.98
gemma3:12b	0.79
devstral:24b	0.98
mistral:7b	0.96
codellama:7b	0.72
dolphin3:8b	0.89

	Avg	API calls	VO get/set	Type Changes	Edge cases
Inswave-base (Rule-based conversion)	0.79	0.32	0.99	0.98	0.87

Rule-based baseline에 비해 ~20%p의 성능 향상

Tried: Programmable converter

([ICML'25] Franzen et al., Product of Experts with LLMs: Boosting Performance on ARC Is a Matter of Perspective)



	Avg	API calls	VO get/set	Type Changes	Edge cases
Inswave-base (Rule-based conversion)	0.79	0.32	0.99	0.98	0.87
Programmable converter	0.74	0.55	0.94	0.96	0.51

API 변환 예시를 추가로 활용하여 converter 제작 -> 성능 향상

Demo & Analysis

<https://nemodleo.github.io/llm-code-converter>

Division and Assignment of Work

■ 배준익
 ■ 박현
 ■ 공동 작업

내용	3월			4월				5월				6월	
	2 주	3 주	4 주	1 주	2 주	3 주	4 주	1 주	2 주	3 주	4 주	1 주	2 주
회사 컨택 및 스펙 검토													
변환 케이스 분석 및 방법 검토													
Self-Refinement 모듈 구현													
의존성 분석 모듈 구현													
구현 모듈 테스트													
프로젝트 중간 발표													
RAG 구현													
IDE 플러그인 포팅 작업 (배포)													
Programmable Converter 구현													
데모 제작 및 발표													