МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ
МЭДЭЭЛЛИЙН ТЕХНОЛОГИ, ЭЛЕКТРОНИКИЙН СУРГУУЛЬ
МЭДЭЭЛЭЛ, КОМПЬЮТЕРЫН УХААНЫ ТЭНХИМ

Бадарчийн Бат-Энх

Үйлдвэрлэлийн дадлага (INTE301)

(Internship report)

Компьютерын Ухаан (D061301)
Үйлдвэрлэлийн дадлагын тайлан

Улаанбаатар

2026 оны 2 дугаар сар

МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ
МЭДЭЭЛЛИЙН ТЕХНОЛОГИ, ЭЛЕКТРОНИКИЙН СУРГУУЛЬ
МЭДЭЭЛЭЛ, КОМПЬЮТЕРЫН УХААНЫ ТЭНХИМ

Үйлдвэрлэлийн дадлага (INTE301)

(Internship report)

Компьютерын Ухаан (D061301)
Үйлдвэрлэлийн дадлагын тайлан

Удирдагч: _____ У. Нямбаяр

Хамтран удирдагч: _____

Гүйцэтгэсэн: _____ Б.Бат-Энх (22B1NUM7226)

Улаанбаатар

2026 оны 2 дугаар сар

Зохиогчийн баталгаа

Миний бие Бадарчийн Бат-Энх ”Үйлдвэрлэлийн дадлага (INTE301)” сэдэвтэй судалгааны ажлыг гүйцэтгэсэн болохыг зарлаж дараах зүйлсийг баталж байна:

- Ажил нь бүхэлдээ Монгол Улсын Их Сургуульд дээд боловсролын зэрэг горилохоор дэвшүүлсэн болно.
- Энэ ажлын аль нэг хэсгийг эсвэл бүхлээр нь ямар нэг их, дээд сургуулийн зэрэг горилохоор оруулж байгаагүй болно.
- Бусдын хийсэн ажлаас хуулбарлаагүй, эшлэл, зүүлтийг зохистой хийсэн болно.
- Ажлыг зохиогч би хийсэн ба миний хийсэн ажил, бусдын дэмжлэгийг дипломын ажилд тодорхой тусгасан болно.
- Ажилд тусалсан бүх эх сурвалжид талархаж байна.

Гарын үсэг: _____

Огноо: _____

ГАРЧИГ

ЗУРГИЙН ЖАГСААЛТ

ХҮСНЭГТИЙН ЖАГСААЛТ

Кодын жагсаалт

УДИРТГАЛ

Энэхүү дадлагын тайлангийн явцад миний бие Бадарчийн Бат-Энх Unitel компанийн Дижитал газрын Дата сайнсын хэлтэст дадлагын төлөвлөгөөнд хамаарах хугацаанд гүйцэтгэсэн судалгаа, хэрэгжилт болон тэдгээрийн үр дүнг танилцуулав. Дадлагын ажлын хүрээнд байгууллагын дотоод процессуудад хиймэл оюун ухаанд суурилсан шийдлүүдийг нэвтрүүлэх, шинэ архитектур болон техник ашиглан комплекс системүүдийн автоматжуулалт, өгөгдөлд суурилсан шийдвэр гаргалтыг оновчлох ажлууд хийгдсэн.

Судалгааны ажлын хүрээнд Agentic AI, Model Context Protocol, Huawei Cloud Stack, Langfuse, Gemini болон AWS Cohere зэрэг орчин үеийн технологи, архитектуруудын онолын үндэс болон практик хэрэглээг судалсан. Харин хэрэгжилтийн хэсэгт ERP системтэй холбогдож буй MCP сайжруулах, Information Retrieval MCP-д вектор өгөгдлийн санг шилжүүлэх, embedding моделиудыг сольж доголдлын үед ажиллагааг бүрэн хангах логик зэргийг хөгжүүлсэн аргачлалыг оруулав. Үүнээс гадна Huawei Cloud Stack дээр development болон production орчныг төлөвлөн байгуулах, CI/CD болон deployment infrastructure-ийг сайжруулах ажлуудыг хийж гүйцэтгэсэн.

Гүйцэтгэсэн ажлуудын үр дүнд LLM-д суурилсан системүүдийн гүйцэтгэл, найдвартай байдал болон ажиглалт, хяналтын түвшин сайжирч, байгууллагын дотоод процессын автоматжуулалт, өгөгдлийн ашиглалтын үр ашиг нэмэгдсэн гэж дүгнэж байна.

Unitel компанийн дотоод бодлогын хүрээнд системийн нарийн зохион байгуулалт болон түүнд ашиглагдаж буй техникийн аргачлалын дэлгэрэнгүй тайлан болон дотоод судалгаа, туршилтын үр дүн болон тэдгээрт хамаарах өгөгдлийг тайланд оруулаагүй болно.

1. БАЙГУУЛЛАГЫН ТАНИЛЦУУЛГА

1.1 Unitel Group

Юнител групп нь Монгол Улсад үйл ажиллагаа явуулдаг харилцаа холбоо, мэдээллийн технологийн компани юм. Тус компани нь 2005 онд үүсгэн байгуулагдаж, 2006 оноос GSM үүрэн холбооны үйлчилгээ үзүүлж эхэлсэн. Үүсгэн байгуулагдсанаасаа хойш Юнител групп нь Монгол Улсын харилцаа холбооны салбарын хөгжилд чухал үүрэг гүйцэтгэж, дэвшилтэт технологийг тасралтгүй нэвтрүүлж ирсэн.

Юнител нь Монгол Улсад 4G LTE болон 5G технологийг нэвтрүүлсэн анхдагч компаниудын нэг бөгөөд хот, хөдөөгийн харилцаа холбооны дэд бүтцийг өргөжүүлэх, өндөр хурдны интернэтийн хүртээмжийг нэмэгдүүлэхээр ажиллаж байна.

1.2 Unitel Group-ийн үйлчилгээ

Юнител групп нь хэрэглэгчдэд дараах үндсэн үйлчилгээнүүдийг үзүүлдэг. Үүнд:

- Үүрэн холбооны үйлчилгээ (дуу, мессеж, дата)
- Өргөн зурвасын интернэт үйлчилгээ
- Суурин утасны үйлчилгээ
- IPTV болон дижитал телевизийн үйлчилгээ
- Streaming болон контент үйлчилгээ
- Хиймэл дагуулын холбооны үйлчилгээ

Unitel компани дээрх үйлчилгээнүүдийг хувь хэрэглэгч болон байгууллагын хэрэгцээнд нийцсэн байдлаар санал болгогддог бөгөөд Монгол Улсын мэдээллийн технологи, дижитал шилжилтийг дэмжихэд чиглэсэн байдаг.

1.3 Дадлагын албан тушаал

Дадлагын хүрээнд миний бие Дата Сайнсын хэлтсийн Дата сайнтистын ажлын байранд ажилласан. Дата сайнсын хэлтсийн хувьд компани дахь бусад баг, хэлтэстэй хамтран өгөгдлийн чанар болон олборлолтыг сайжруулах, AI системүүдийн нэвтрүүлэлт, автоматжуулалтын ажлуудад түлхүү чиглэсэн ажил гүйцэтгэдэг.

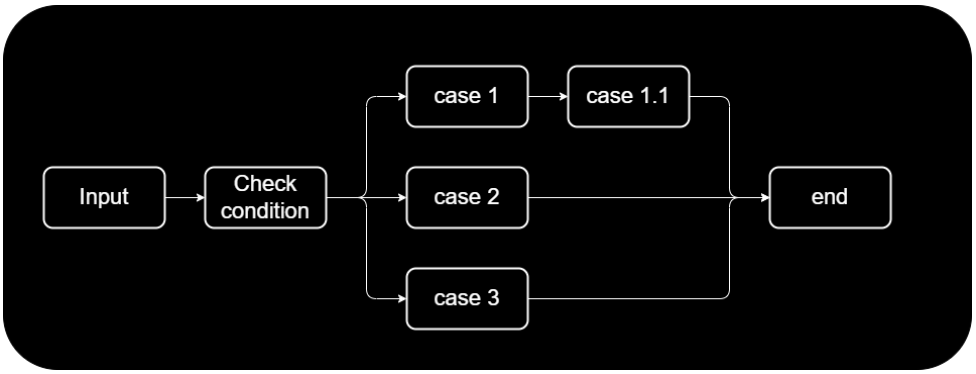
Дадлагын ажлын хугацаанд миний бие ажилтанд зориулсан AI туслах, Call center-ийн ажилтны AI туслах, TV content автомат дата оруулагч болон хэлтсийн Development ба Production орчны зохион байгуулалтуудыг хариуцан ажилласан.

2. СУДАЛГАА

Unitel Group нь дэлхийн хиймэл оюуны ухааны хувьсгалтай ижил түвшинд хөгжих зорилгын дагуу шинэ технологи болон архитектуруудыг компаны хэрэгцээнд зориулан байнга нэвтрүүлэн ашигладаг.

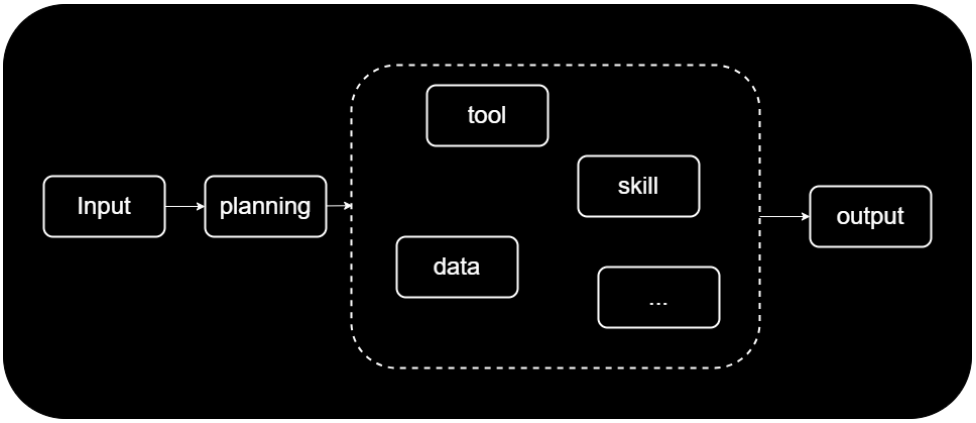
2.1 Agentic AI

Agentic архитектур нь нэг ёсондоо өөрийгөө тодорхойлдог workflow юм [7] [5]. Large Language Model (LLM)-ийн огцом хөгжил нь энэ архитектурын динамик байдлыг хангаж комплекс ажлуудыг програмчлалын аргаар шийдэх боломжийг олгосон.



Зураг 2.1: Workflow

Workflow нь статик програмчлагдсан бөгөөд зорилго болон зорилго биелүүлэхэд хэрэгцээтэй зүйлсийг тодорхойлох механизм байхгүй.



Зураг 2.2: Agentic

2.2 Model Context Protocol

Antropic компанийн хөгжүүлсэн тус протокол нь LLM-д tool call хийх боломж олгох хамгийн уян хатан технологи юм. Model Context Protocol ашигласнаар LLM нь гадны өгөгдөлтэй харьцах стандарт харилцааны протоколтой болсон. Үүний үр дүнд LLM-ийн модель болгон өөрийн гэсэн tool-тэй байхын оронд нэг tool-ийг олон модель зэрэг ашиглах, өөр хүмүүсийн бэлтгэсэн tool-ийг ашиглах зэрэг боломжууд үүссэн [10].

2.3 Huawei Cloud Stack

Huawei Cloud Stack (HCS) нь Data Center удирдлагын систем бөгөөд Unitel компани өөрийн Data Center-ээ тус системээр удирддаг бөгөөд нэвтрүүлэлтийн болон хөгжүүлэлт ажлуудыг удирдахад тус системийн тухай мэдлэгтэй байх шаардлагатай байдаг [6].

2.3.1 Elastic Compute Service

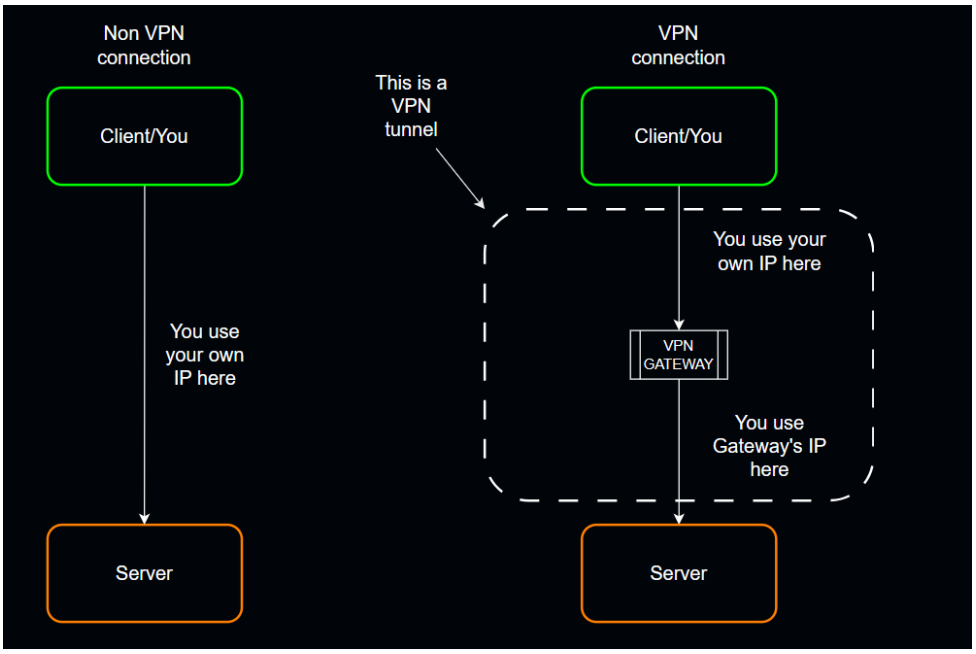
HCS-ийн Elastic Compute Service (ECS) нь AWS-ийн EC2-тэй дүйх бөгөөд аливаа сүлжээний subnet дээр орших виртуал машин юм [12][6].

2.3.2 Virtual Private Cloud

HCS дотор орших тусгаарлагдсан Data Center-ийн дотоод сүлжээ юм. VPC-г зөв зохион байгуулах нь цаашид хандалтыг хязгаарлах, бусад гадны сүлжээтэй харилцах гэх мэт боломжуудыг олгоно [6].

2.3.3 Virtual Private Network

VPC-ийн дотоод сүлжээ нь гадны дотоод сүлжээтэй холбогдохын тулд VPN холболт үүсгэх шаардлагатай байдаг [6].



Зураг 2.3: VPN холболтын диаграм

2.4 Langfuse

Langfuse нь LLM дээр суурилсан системүүдийн ажиглалт, хяналт болон гүйцэтгэлийн шинжилгээнд зориулагдсан open-source платформ юм [8]. Agentic AI архитектурын хувьд олон шатлалт reasoning, tool call, decision making процессууд явагддаг тул эдгээрийг бүртгэн хадгалах шаардлага үүсдэг. Langfuse нь prompt, response, latency, token usage зэрэг мэдээллүүдийг цуглуулж, LLM-ийн үйл ажиллагааг бодит орчинд хянах боломжийг олгодог.

Unitel Group нь Langfuse-г LLM-д суурилсан системүүдийг Production ба Development орчинд ашиглах үед алдааг илрүүлэх, гүйцэтгэлийг оновчлох мөн LLM-ийн Prompt mangement хийх зорилгоор ашигладаг.

2.5 Gemini

Gemini нь Google компанийн хөгжүүлсэн Large Language Model юм [4]. Тус модельд embeddings үүсгэх чадвар 2025 онд шинээр нэмэгдсэн.

Gemini embedding нь текстийг семантик хайлтад ашиглах боломжтой 3072 хэмжээст

вектор болгон хувиргадаг . Agentic AI архитектурын хүрээнд embeddings-ийг ашигласнаар систем нь хадгалагдсан мэдлэгтэй уялдуулан шийдвэр гаргах боломж бүрддэг.

2.6 AWS Cohere

Amazon Web Service-ийн санал болгодог service-үүдийн нэг болох Cohere нь Large Language Model болон embeddings үйлчилгээ үзүүлэгч юм[1].

Unitel Group саяхныг хүртэл Cohere Multilingual v3 embedding моделийг ашигладаг байсан бөгөөд 2026 оны 1 сард хийсэн туршилт дээр үндэслэн Cohere-ийг fallback модель, харин gemini-г үндсэн embedding модель болгон шилжүүлсэн.

2.7 Github

Github нь Git reposistory host service үзүүлэгч юм [9]. Мөн үүнтэй уялдуулан Job pipeline (Github Actions), discoverability болон эх код дээр хамтран ажиллах гэх мэт нэмэлт үйлчилгээ үзүүлдэг.

2.7.1 Github self-hosted runner

Github self-hosted runner нь Github Actions workflow-г тухайн хэрэглэгчийн өөрийн дэд бүтцэд байршуулсан server дээр гүйцэтгэх боломж олгодог механизм юм [9]. Cloud-based runner-ээс ялгаатай нь self-hosted runner нь дотоод сүлжээ, Data Center болон тусгаарлагдсан орчинд байрших системүүдтэй шууд харилцах боломжтой.

Дата Сайнсын хэлтсийн хувьд Github ашиглан CI/CD болон эх кодоо удирддаг учраас Github self-hosted runner ашиглан CI/CD процессыг дотоод орчинд гүйцэтгэдэг.

2.7.2 Github Actions

Github Actions нь программ хангамжийн хөгжүүлэлт, тест, нэвтрүүлэлтийн (CI/CD) процессыг автоматжуулах зориулалттай workflow orchestration систем юм [9]. Workflow нь event-д суурилан ажилладаг бөгөөд кодын өөрчлөлт, pull request, эсвэл гар ажиллагаатай trigger-ээр эхлэн олон шатлалт процессыг гүйцэтгэх боломжтой.

2.8 Google Cloud Platform

Google Cloud Platform (GCP) нь Google компанийн хөгжүүлсэн cloud computing үйлчилгээний цогц платформ бөгөөд compute, storage, networking, data analytics болон хиймэл оюун ухаанд суурилсан олон төрлийн үйлчилгээг санал болгодог. GCP-гийн өндөр найдвартай байдал, аюулгүй байдлын механизм, автоматжуулалтын боломжуудтай гэдгээрээ Дата Сайнсын хэлтсийн хэрэглээнд тохиромжтой тул өөрсдийн систем болон pipeline-ууддаа ашигладаг.

2.8.1 Cloud Build

Cloud Build нь Google Cloud Platform-ийн Continuous Integration / Continuous Deployment (CI/CD) үйлчилгээнүүдийн нэг бөгөөд автоматаар build, test болон deploy хийх боломж олгодог [2]. Cloud Build нь event-д суурилан ажилладаг бөгөөд GitHub зэрэг source control системүүдтэй шууд уялдан ажиллах чадвартай гэдгээрээ давуу талтай.

Энэхүү сервисийг ашигласнаар аливаа server дээр тусгай build орчин урьдчилан бэлдэх шаардлагагүй бөгөөд build процесс бүр тусгаарлагдсан, аюулгүй орчинд гүйцэтгэгддэг. Энэ нь deployment процессийн тогтвортой байдал, давтагдах чанар болон аюулгүй байдлыг хангахад чухал үүрэгтэй.

2.8.2 Workload Identity Federation

Workload Identity Federation (WIF) нь гадны орчинд ажиллаж буй workload-уудаас (GitHub Actions) ирж буй хүсэлтүүдийг баталгаажуулахад ашиглагддаг Google Cloud Platform-ийн authentication механизм юм.

WIF ашигласнаар workload нь өөрийн identity (OIDC token)-г ашиглан Google Cloud-ийн service account-т түр хугацааны authenthication ашиглан event trigger хийх зарчмаар ажилладаг.Иңгэснээр static key хадгалах шаардлагагүй болж байгаа учир Service Account Key алдагдах эрсдэлийг бууруулдаг.

Цаашлаад хандалтыг илүү нарийн, policy-д суурилан удирдах боломжтой болно

Энэхүү механизм нь enterprise орчинд CI/CD процессыг Google Cloud-той аюулгүй,

автомат байдлаар холбох хамгийн зөв шийдэлд тооцогддог.

2.8.3 Artifact Registry

Artifact registry нь Google Cloud platform дээр build хадгалах storage систем юм. Тус service-ийг Cloud Build дээр хийсэн build-үүдээ Production server-т дамжуулахын тулд ашиглана.

2.9 Qdrant

Qdrant нь өндөр гүйцэтгэлтэй, Open-Source вектор өгөгдлийн сан бөгөөд их хэмжээний embedding векторууд дээр ойролцоо хайлт (Approximate Nearest Neighbor) гүйцэтгэхэд зориулагдсан систем юм [11]. Тус систем нь cosine similarity, dot product болон Euclidean distance зэрэг олон төрлийн similarity хэмжүүрийг дэмждэг бөгөөд Large Language Model (LLM)-д суурилсан Retrieval-Augmented Generation (RAG), semantic search болон recommendation системүүдэд өргөн ашиглагддаг.

Qdrant нь metadata filtering, tenant-based separation болон access control зэрэг enterprise түвшний боломжуудыг дэмждэгээрээ давуу талтай бөгөөд self-hosted орчинд ашиглах боломжтой.

3. ХЭРЭГЖҮҮЛЭЛТ

3.1 Enterprise Resource Planning MCP

Unitel компанийн ажилтны AI туслах нь Enterprise Resource Planning (ERP) системээс мэдээлэл татах чадвартай. Энэ нь AI туслах ажилтанд ээлжийн амралтын үлдсэн хоног, хэзээ амралтаа авч болох, илүү цагийн эсвэл хоцролтын мэдээллийг хүргэх гэх мэт боломжуудыг бүрдүүлж өгдөг.

Миний бие дадлагын хугацаанд ERP MCP server-ийг хариуцан ажилласан бөгөөд хийгдсэн ажлуудыг дурдвал:

- Шинэ API call хийх tool хэрэгжүүлэх.
- Шаардлагатай сайжруулалтуудыг цаг тухайд нь хийж гүйцэтгэх.

```
def get_user_info(
        self, employee_domain: str = None, first_name: str = None,
           last_name: str = None
    ) -> str:

  # Preprocessing checks
  if not employee_domain and not (first_name and last_name):
  ...
  # criteria building logic
  criteria = {}
  if employee_domain:
    ...
  payload = {
    ...
    "criteria": criteria
  }
```

```
16
17    # response parsing logic
18    try:
19      response = self.send_request(payload, 10)
20    except requests.RequestException as e:
21      ...
22    try:
23      ...
24    except (KeyError, TypeError):
25      ...
26
27    result = result["0"]
28    json_string = json.dumps(result, ensure_ascii=False, indent=2)
29
30    return json_string
```

Код 3.1: Бусад ажилчдын мэдээллийг хайх tool-ийн хэрэгжүүлэлт

ERP системтэй холбогдох холболтыг session байдлаар шийдсэн бөгөөд ингэснээр TLS handshake болон TCP холболтууд хүсэлт болгон дээр үүсэх шаардлагагүй болсон.

```
1  def __init__(self):
2    ...
3    self.session = self._create_session()
4  def _create_session(self):
5    session = requests.Session()
6    retries = requests.adapters.Retry(
7      total=2,
8      backoff_factor=0.5,
9      status_forcelist=[500, 502, 503, 504],)
10   adapter = requests.adapters.HTTPAdapter(max_retries=retries)
11   session.mount("http://", adapter)
```

```
  session.mount("https://", adapter)
  return session
```

Код 3.2: Request session үүсгэж буй логик

```
def send_request(self, payload: dict, timeout: int):
  try:
    return self.session.post(
      self.ERP_URL,
      headers=self.ERP_HEADERS,
      data=json.dumps(payload),
      verify=False,
      timeout=timeout,
    )
  except requests.RequestException as e:
    logger.error(f"[send_request] ERP request failed after retries: {
      e}")
    raise
```

Код 3.3: Session ашиглаж буй POST request-ийн wrapper

3.2 Information Retrieval MCP

Дадлагын хугацаанд Information Retrieval MCP tool-ийг хариуцсан ажилтны хувьд дараах ажлуудыг хийсэн.

3.2.1 Вектор өгөгдлийн сан солих

Unitel Chatbot болон бусад RAG технологи ашигладаг системүүдэд AstraDB-ийн Cloud вектор өгөгдлийн санг ашигладаг байсан бол 2026 оноос эхлэн Self-Hosted хувилбар луу шилжих ажил хийгдэж хэд хэдэн вектор өгөгдлийн санг харьцуулснаас Qdrant хамгийн тохиромжтой гэх дүгнэлтэд хүрсэн. Үүнд харгалзаж үзсэн үзүүлэлтүүдийг дурдвал:

- Retrieval latency

- Self-Host үнийн санал
- Өөрийн гэсэн User Interface
- Вектор өгөгдлийн сан дээрх хийгдэж болох үйлдлүүд:
  - Metadata filtering
  - Tenant based separation
  - Access control

Иймд Information Retrieval MCP tool-ийн ашигладаг вектор өгөгдлийн санг солих ажлыг дараах байдлаар сольсон.

```
1
2  class Qdrant:
3    def __init__(self):
4      settings = get_settings()
5      self.gemini_embedder = GoogleEmbedder().google_embeddings
6      self.cohere_embedder = AWSEmbedder().aws_embeddings
7      self.client = QdrantClient(
8        url=settings.QDRANT_CLIENT_URL,
9        api_key=settings.QDRANT_API_KEY,
10       timeout=20.0  # 60 second timeout for Qdrant operations
11     )
12     self.QDRANT_PROD_MODE=settings.QDRANT_PROD_MODE
```

Код 3.4: Qdrant client initialization

```python
    async def vector_search(self, query: str, filter=None, tenant_id=
        None, embedding_name=None, top_k: int=10):
        collection_name = self._qdrant_collection_name(tenant_id=
            tenant_id, embedding_name=embedding_name)
        #embded query
        if embedding_name == "gemini":
            vec = await self.gemini_embedder.aembed_query(query)
        else:
            vec = await self.cohere_embedder.aembed_query(query)
        hits = self.client.query_points(
            collection_name=collection_name,
            query=vec,
            query_filter=filter,
            limit=top_k,
        )
        return hits
    def __call__(self, query, filter, top_k=10):
        self.vector_search(query=query, filter=filter, top_k=top_k)
```

Код 3.5: Qdrant client вектор хайлт

3.2.2 Embedding модель солих

Cohere

2026 онд хийгдэж эхэлсэн ажлуудад вектор өгөгдлийн санг шинэчлэхээс гадна embed хийж буй моделиудыг шинэчлэх мөн fallback модель сонгох ажил багтсан. Өмнө нь AWS-ийн Cohere Multilingual v3 [1] модель Монгол хэл дээр хамгийн сайн үр дүн үзүүлсэн тул сүүлийн 2 жилийн турш сонгон ашигласан.

Gemini Embedding

Google-ийн хөгжүүлсэн gemini-embedding-001 модель нь олон нийтэд 2025 оноос эхлэн нээлттэй болсон [3].

Unitel компани дотооддоо нээлттэй embedding моделиуд дээр

- Хайлтын хурд
- Хайлтын оновчтой байдал

гэсэн хоёр үзүүлэлтүүдэд тулгуурласан туршилт хийсэн. Туршилтын үр дүнд Gemini Embedding хамгийн хурдан бөгөөд оновчтой ажиллсан тул үндсэн моделиор сонгогдож, харин Cohere Multilingual v3 fallback моделиор сонгогдсон.

```
class AWSEmbedder:
    def __init__(self):
        settings = get_settings()
        self.aws_embeddings = BedrockEmbeddings(
            client=AWSClient().aws_client_embed,
            region_name=settings.AWS_REGION,
            model_id=settings.AWS_COHERE_MODEL_NAME,
            normalize=False,
            model_kwargs={"input_type": "clustering"},
        )
```

Код 3.6: Cohere Embedding client

```
class GoogleEmbedder:
    def __init__(self):
        settings = get_settings()
        # Create httpx client with timeout
        http_client = httpx.Client(
            timeout=httpx.Timeout(20.0, connect=15.0)
        )
        self.google_embeddings = GoogleGenerativeAIEmbeddings(
            model=settings.GOOGLE_EMBEDDING_MODEL_NAME,
            api_key=settings.GOOGLE_GEMINI_API_KEY,
            client=http_client,
        )
```

Код 3.7: Gemini Embedding client

```
        async def vector_search(self, query: str, filter=None,
            tenant_id=None, embedding_name=None, top_k: int=10):

        collection_name = self._qdrant_collection_name(tenant_id=
            tenant_id, embedding_name=embedding_name)

        #embded query
        if embedding_name == "gemini":
            vec = await self.gemini_embedder.aembed_query(query)
        else:
            vec = await self.cohere_embedder.aembed_query(query)

        hits = self.client.query_points(
            collection_name=collection_name,
            query=vec,
            query_filter=filter,
            limit=top_k,
        )
        return hits
```

Код 3.8: Embedding модель fallback логик

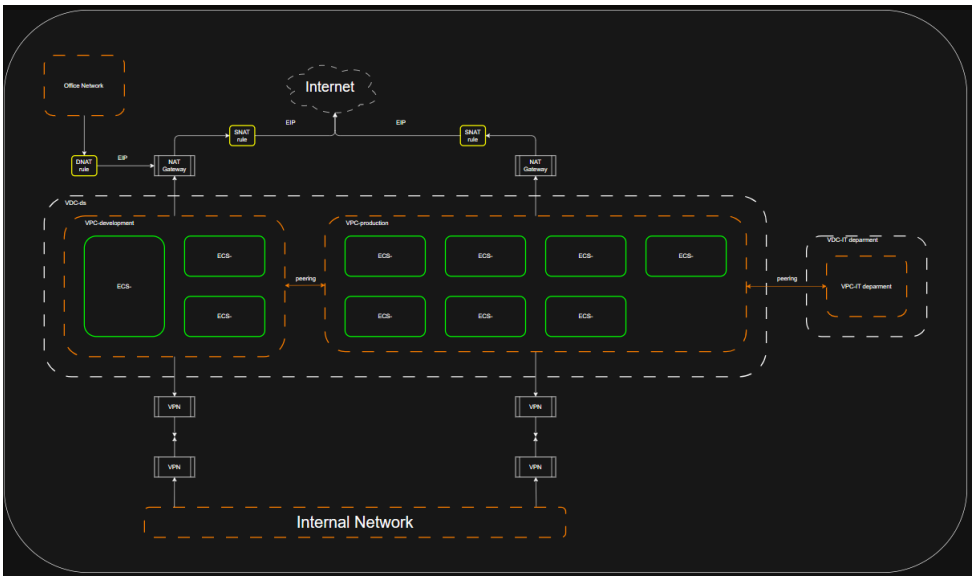
3.3 TV content data entry automation

Unitel компанийн охин компани болох Univision компани нь „„ үйлчилгээ үзүүлдэг. Univision компанитай гэрээтэй сувгуудын хөтөлбөрийг гараар системд бүртгэн, хянаж оруулдаг. Миний бие дадлагын ажлын хугацаанд тус системийн өгөгдөл цэвэрлэх системд шаардлагатай сайжруулалтуудыг цаг тухайд нь гүйцэтгэн ажилласан.

3.4 Huawei Cloud Stack

Unitel компани Data Center-ээ Huawei Cloud Stack (HCS) ашиглан удирддаг. HCS нь функциональ талаасаа AWS-тэй ижил үүрэг гүйцэтгэдэг [6][12].

Дижитал трансформейшны газарт 2026 оноос эхлэн Unitel компанийн Data Center руу өөрсдийн Local Production болон Local Development орчноо шилжүүлэх ажил явагдаж байгаа. Дадлагын ажлын хүрээнд тус системийг төлөвлөгөөний дагуу байгуулах ажлыг гүйцэтгэсэн болно.



Зураг 3.1: HCS дээрх системийн зохион байгуулалт

3.4.1 Development server дээр хийгдсэн ажлууд

Development server-т шаардлагын дагуу хийгдсэн ажлууд:

- Rootless docker суулгах
- Хэлтсийн гишүүн бүрд user үүсгэж анхны тохиргоо хийх.
- Common user тохируулах.
- System disk хэмжээ ихэсгэх.
- Development server-ээс гарах интернэт холболт үүсгэх.

- Development server-т оффисын сүлжээнээс ирэх хандалтыг нээх.
- Development server-ийг бусад дотоод сүлжээтэй холбох.
- Template devcontainer орчин бэлдэх.
- Development server ашиглах болон дахин тохиргоо хийх талаар documentation бичих.
- Бусад шаардлагатай software суулгах.

3.4.2 Production server-үүд дээр хийгдсэн ажлууд

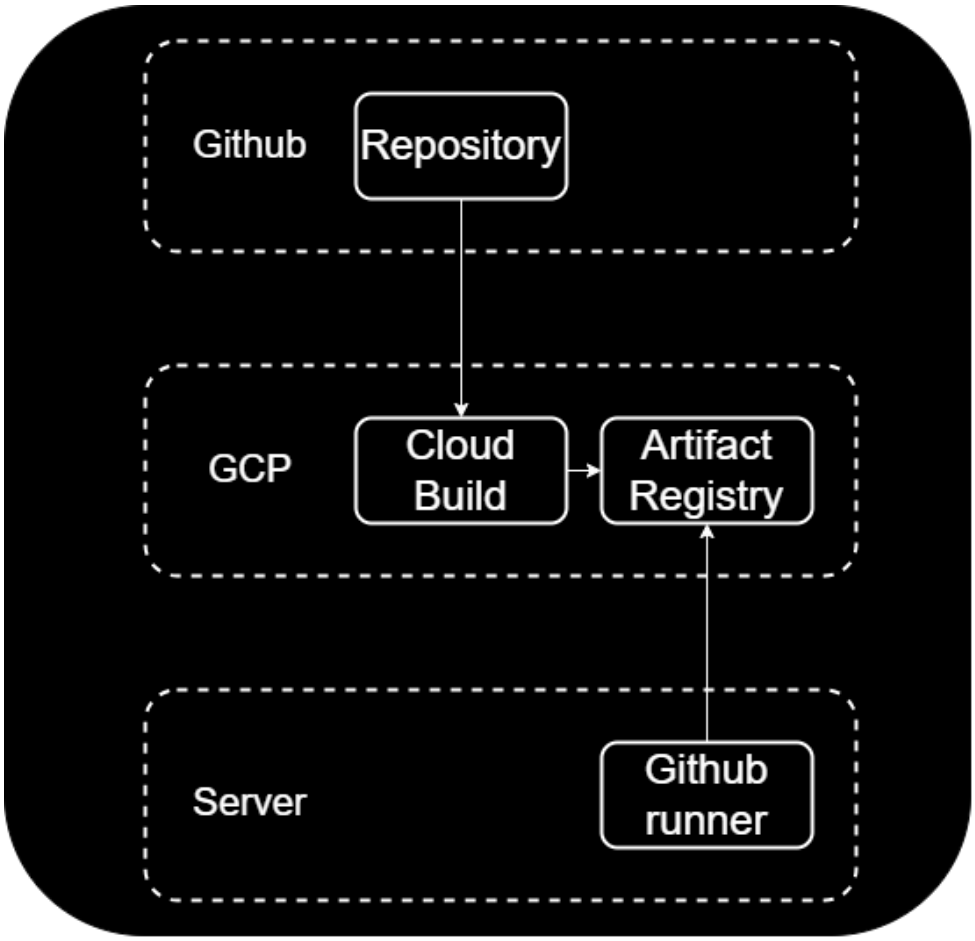
Production server-т шаардлагын дагуу хийгдсэн ажлууд:

- Project бүрд тусгаарлагдсан server үүсгэх.
- Server бүрд Rootful Docker суулгах.
- Server бүрд хандах шаардлагатай дотоод болон гадаад сүлжээ рүү хандалт нээж өгөх.
- System disk хэмжээ ихэсгэх.
- Data disk attach хийх.
- Production server-ээс гарах интернэт холболт үүсгэх.
- Common user тохируулах.
- ШИнэ Production server үүсгэх болон дахин тохируулах талаар documentation бичих

3.5 Deployment infrastructure

Дата Сайнсын хэлтэс дотоод болон бусад баг, хэлтэст зориулан хөгжүүлсэн 20 орчим production repository-г удирддаг. Тэдгээрийн deployment процессыг автоматжуулах буюу workflow definition script-ийг бичих зааварчилгааг боловсруулах ажлыг миний бие хариуцдаг.

Deployment infrastructure-ийн ерөнхий зохион байгуулалт:



Зураг 3.2: Deployment infrastructure диаграм

Дадлагын ажлын хүрээнд deployment infrastructure-т хийгдсэн ажлууд:

- Agentic Assistant repository олон deployment file-тай байгааг нэгтгэж parameter-аар шийдэх.
- Шинээр deployment хийгдэх repository-ийг deployment script-ийг үүсгэх documentation бичих.
  - Google Cloud Platform талаас хийгдэх тохиргоо
  - Github талаас хийгдэх тохиргоо
  - Self runner host хийж буй server талаас хийгдэх тохиргоо.

```
on:
workflow_dispatch:
  inputs:
  target:
    description: "Where to deploy"
    required: true
    default: all
    type: choice
    options:
    - all
    - production
    - staging
    ...
```

Код 3.9: Deployment script-ийг input-ээр удирдах script

4. ҮР ДҮН

Дадлагын ажлын хүрээнд Unitel Group компанийн Дата сайнсын хэлтсийн дотоод системүүдэд хиймэл оюун ухаанд суурилсан шийдлүүдийг судалж, бодит орчинд хэрэгжүүлсэн үр дүнгүүд дээр дүгнэлт хийвэл.

Agentic AI архитектур болон Model Context Protocol (MCP)-д суурилсан шийдлүүдийг нэвтрүүлснээр AI туслахуудын гүйцэтгэх боломжит үйлдлүүд өргөжсөн. Үүнд

- Qdrant вектор өгөгдлийн сан болон gemini embedding модель ашигласнаар Retrieval MCP-ний хурд 4 дахин сайжирсан.
- ERP MCP-ний дундаж гүйцэтгэлийн хурд 10 хувь багасаж, гуравдагч системийн доголдлыг зохицуулах логик сайжруулсан.

TV Content дата оруулалтын ажлыг автоматжуулснаар хоёр ажилтны 16 цагийн ажлыг нийт 1 хүн/цаг болгон бууруулж байгаа.

Үүнээс гадна Huawei Cloud Stack орчинд development болон production server-үүдийн архитектурыг зохион байгуулснаар дотоод орчны тусгаарлалт, аюулгүй байдал болон deployment процессыг сайжруулсан нь:

- Server уналтыг багасгах.
- Development server-ийн ачааллыг багасгах.
- Production service-үүдийг тусгаарлагдсан орчинд удирдсанаар бусад service-ээс хамаарсан доголдол гарахгүй байх гэх мэт давуу талуудыг олгож байгаа.

Дүгнэлт

Энэхүү дадлагын ажлын хүрээнд хиймэл оюун ухаанд суурилсан системүүдийг байгууллагын бодит хэрэгцээнд нийцүүлэн судалж, хэрэгжүүлэх практик туршлага олж авлаа. Agentic AI архитектур, Model Context Protocol, вектор өгөгдлийн сан, embedding модель болон cloud дэд бүтцийн зохион байгуулалт зэрэг орчин үеийн технологиуд дээр хийгдсэн онолын ба практикийн цогц ажлууд шинэ технологиудтай ажиллах маш чухал туршлага болсон.

Мөн production орчинд ашиглагдаж буй системүүдэд хяналт тавих, найдвартай байдлыг хангахын ач холбогдол, түүнчлэн дотоод дэд бүтцэд тохирсон технологийн сонголт хийх шаардлагыг практик түвшинд ойлгож чадсан. Үүнээс гадна server-ийн архитектур зохион байгуулах тэдгээрийн гадаад болон дотоод сүлжээний холболт зэргийн хянан ажилласан туршлага нь мэргэжлийн хичээлд төдийлөн тусгагддаггүй компьютер сүлжээний талаарх мэдлэгээ дээшлүүлэх боломж олголоо.

Цаашид Agentic AI системүүдийг нарийвчлалтай ажлуудад ашиглах, хиймэл оюун ухаанд суурилсан автоматжуулалтыг бизнес процессуудад нэвтрүүлэх болон тэдгээрийн оршиж буй техник орчныг бүрдүүлэн хяналт тавих зэрэг дадлагын ажлын явцад олж авсан олон талын мэдлэгээ сайжруулан бусад судалгаа, хэрэгжүүлэлтэд дахин ашиглах бүрэн боломжтой гэж үзэж байна.

Bibliography

[1] Cohere. Cohere documentation. https://docs.cohere.com, 2026. Accessed: 2026-02-03.

[2] Google. Google cloud documentation. https://docs.cloud.google.com/docs, 2026. Accessed: 2026-02-07.

[3] Google AI. Gemini documentation: Embeddings. https://ai.google.dev/gemini-api/docs/embeddings, 2024. Accessed: 2026-02-06.

[4] Google AI. Gemini documentation: Api. https://ai.google.dev/gemini-api/docs, 2026. Accessed: 2026-02-03.

[5] Google Cloud. What is agentic ai. https://cloud.google.com/discover/what-is-agentic-ai, 2026. Accessed: 2026-02-03.

[6] Huawei. Huawei cloud stack documentation. https://support.huawei.com/enterprise/en/cloud-computing/huawei-cloud-stack-pid-23864287, 2026. Accessed: 2026-02-03.

[7] IBM. What is agentic ai? https://www.ibm.com/think/topics/agentic-ai, 2026. Accessed: 2026-02-02.

[8] Langfuse. Langfuse documentation. https://langfuse.com/docs, 2026. Accessed: 2026-02-03.

[9] Microsoft. Github actions documentation. https://docs.github.com/en/actions, 2026. Accessed: 2026-02-07.

[10] Model Context Protocol. Getting started with model context protocol. `https://modelcontextprotocol.io/docs/getting-started/intro`, 2026. Accessed: 2026-01-29.

[11] Qdrant. Qdrant documentation. `https://qdrant.tech/documentation/`, 2026. Accessed: 2026-01-29.

[12] Amazon Web Services. Amazon web service documentation. `https://docs.aws.amazon.com`, 2026. Accessed: 2026-02-06.