Norkart - KartAI project

TDT4290 - Customer-driven development

Group 7

Introduction

- KartAI
- Norkart
  - Alexander Salveson Nossum
  - Håvard Watland
- Kristiansand Municipality
  - Eva Høksaas
  - Dagfinn Øksendal

Team

Group contract

- Align expectations
- Establish group norms

Group dynamics

- Inclusiveness, cohesion & collaboration

Teambuilding

- Initial group dinner
- AI workshop
- Trip to Kristiansand

Team Structure

Johanne Eide Omland
Team Leader

Magnus Andreas Giverin
Deputy Leader

Sverre Nystad
AI Lead

Andreas Lilleby Hjulstad
System Architect

Artemis Kjøllmoen Aarø
Team Product Owner

Maurice Wegerif
Report Lead

Project proposal

"The project goal is to **develop a full stack web based system** that **integrates** and **further develops the different "AI assistants"** in a **user friendly** application tailored to the work process and technical knowhow of both **case workers, citizens and companies** involved in the building permit application process."

- Alexander Salveson Nossum, Norkart Project Proposal document for TDT4290

Project motivation

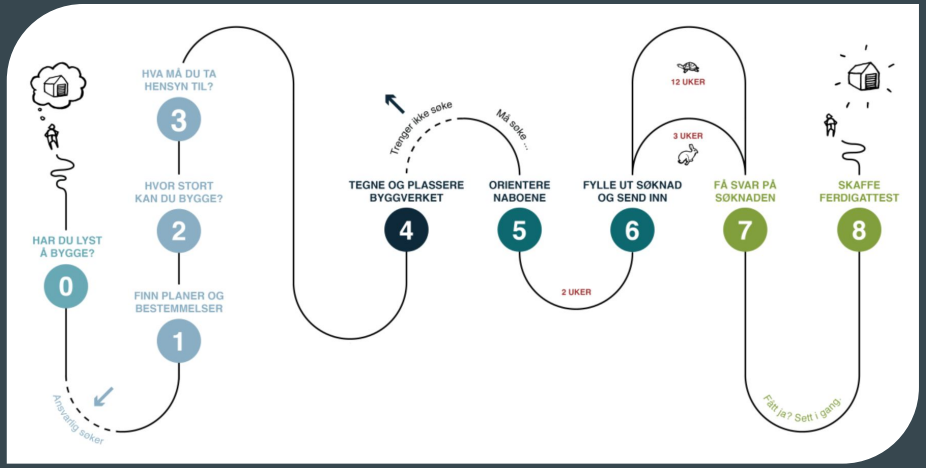
1. Reduce number of errors in building permit applications

2. Make the building permit application handling more efficient with digital tools

3. Bridge and test AI technologies in a human centric software system

Problem Space

- Preliminary Research
- 6 interviews with municipal case workers
- Cataloguing **needs** vs. **wants**
  - Tools for applicants
  - Tools for municipal case workers
- **Before**, **during**, and **after** application



Solution Space

Explorative design

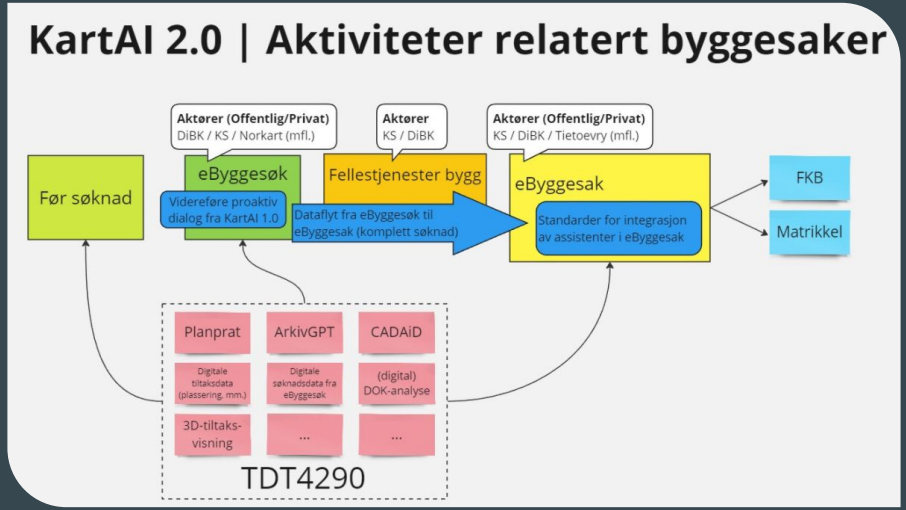
First proposal:

- Integrate into existing **eByggesøk** solution

Final scope:

- Proof-of-concept
- Three stages
- **Figma** user tests

Expanded scope - AI summary assistant



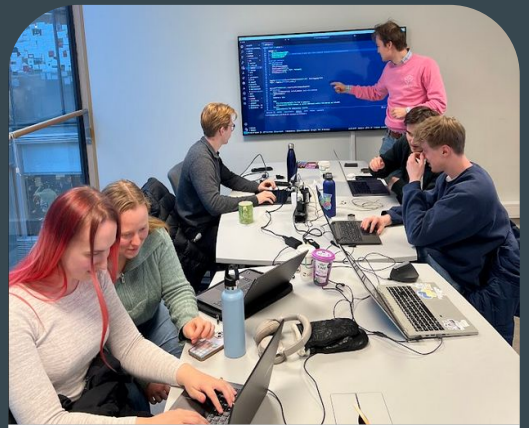
Development Methodology

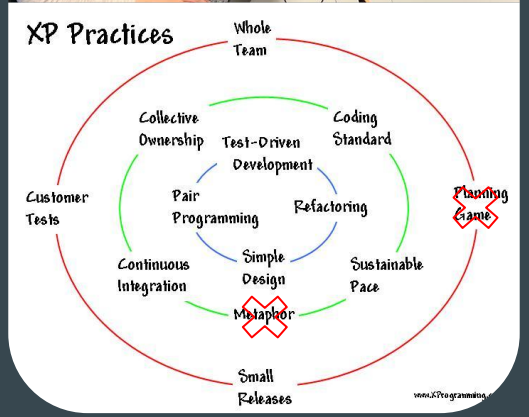
Scrum

- Sprint planning
- Sprint retrospectives
- Daily standup

Extreme Programming (XP)

- Test-driven development
  - Continuous refactoring
- Pair programming





Final Product

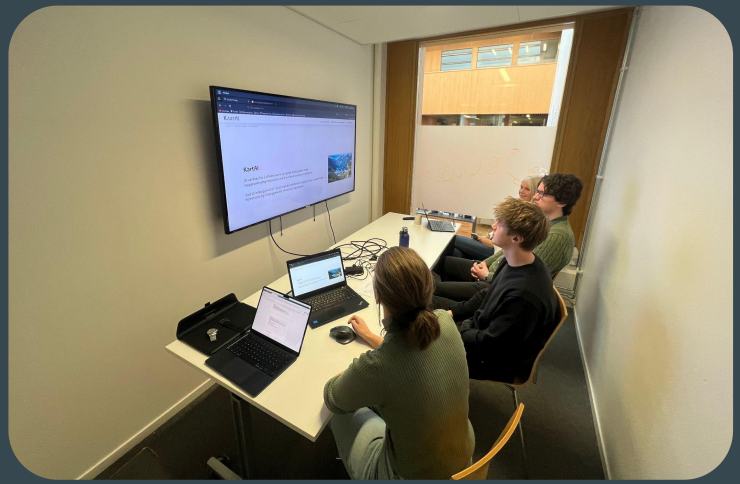
Two parts - **Web application** and **AI assistant API**

1. Web application

- **User tests** on **live demo** to ensure **usability**

2. AI Summary Assistant

- **Summarizes** applications
- **Validating** applications against official checklists, laws, and regulations
- **AI Innovation**: Combining new Agent architectures
    - CRAG and Reflexion



Demo

Self-Evaluation

Solution

- Incomplete requirements
- Security

Sustainability

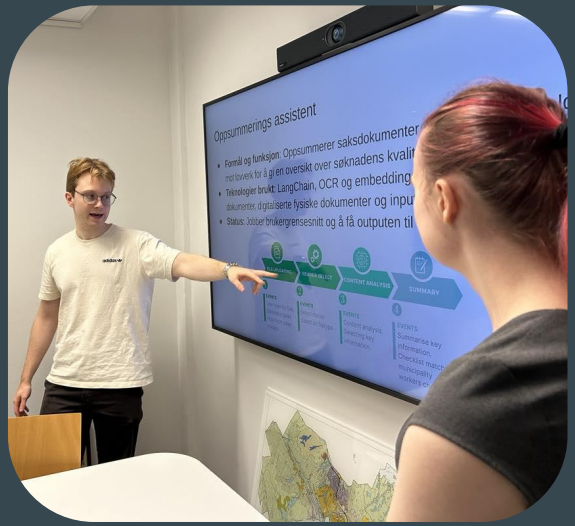
- Technical, social & environmental sustainability

Internal Processes

- Cohesion in the team was essential
- Proactive meeting scheduling needed improvement

User testing

- Last user testing came close to the delivery date
- Clearly specifying that it was a PoC
- Diversity - 5 women and 3 men aged 20-60+

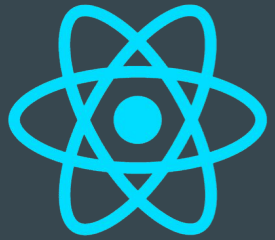


Q&A

- **"Any questions?"**
- Discussion and feedback.

Technologies

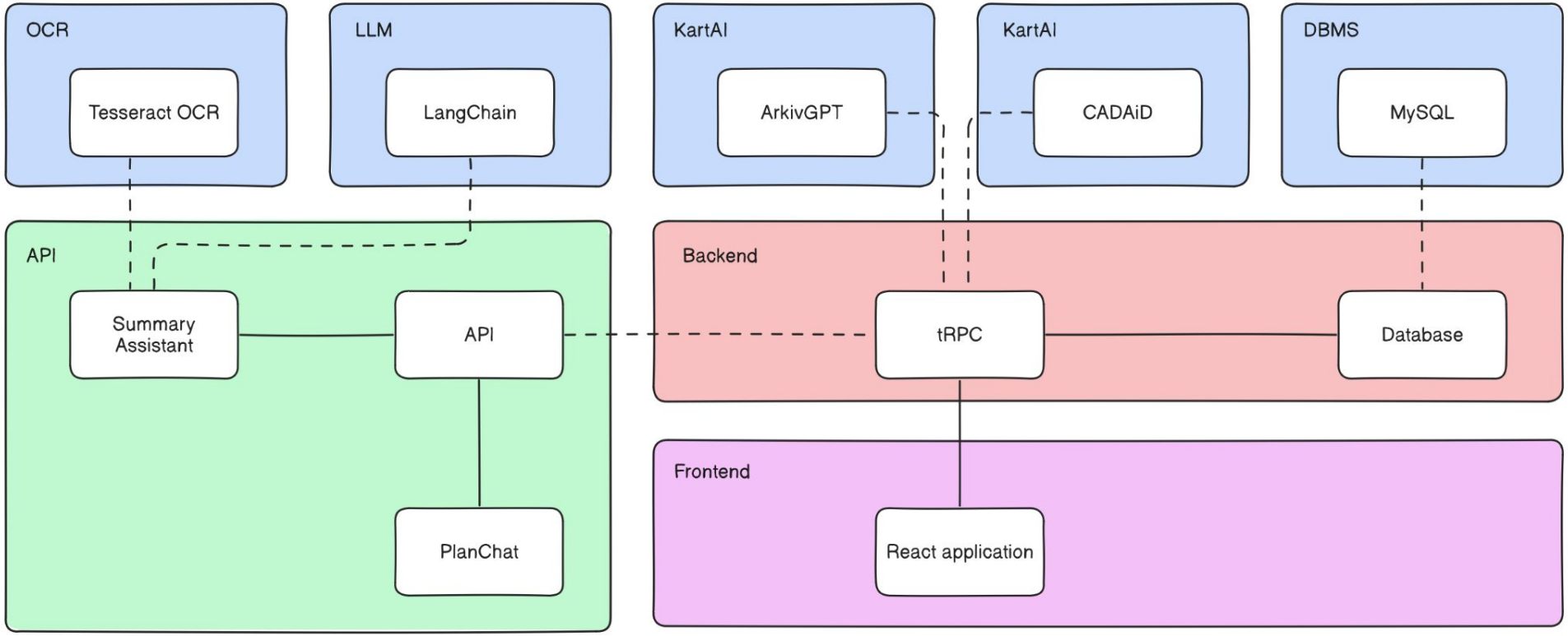
Web application:

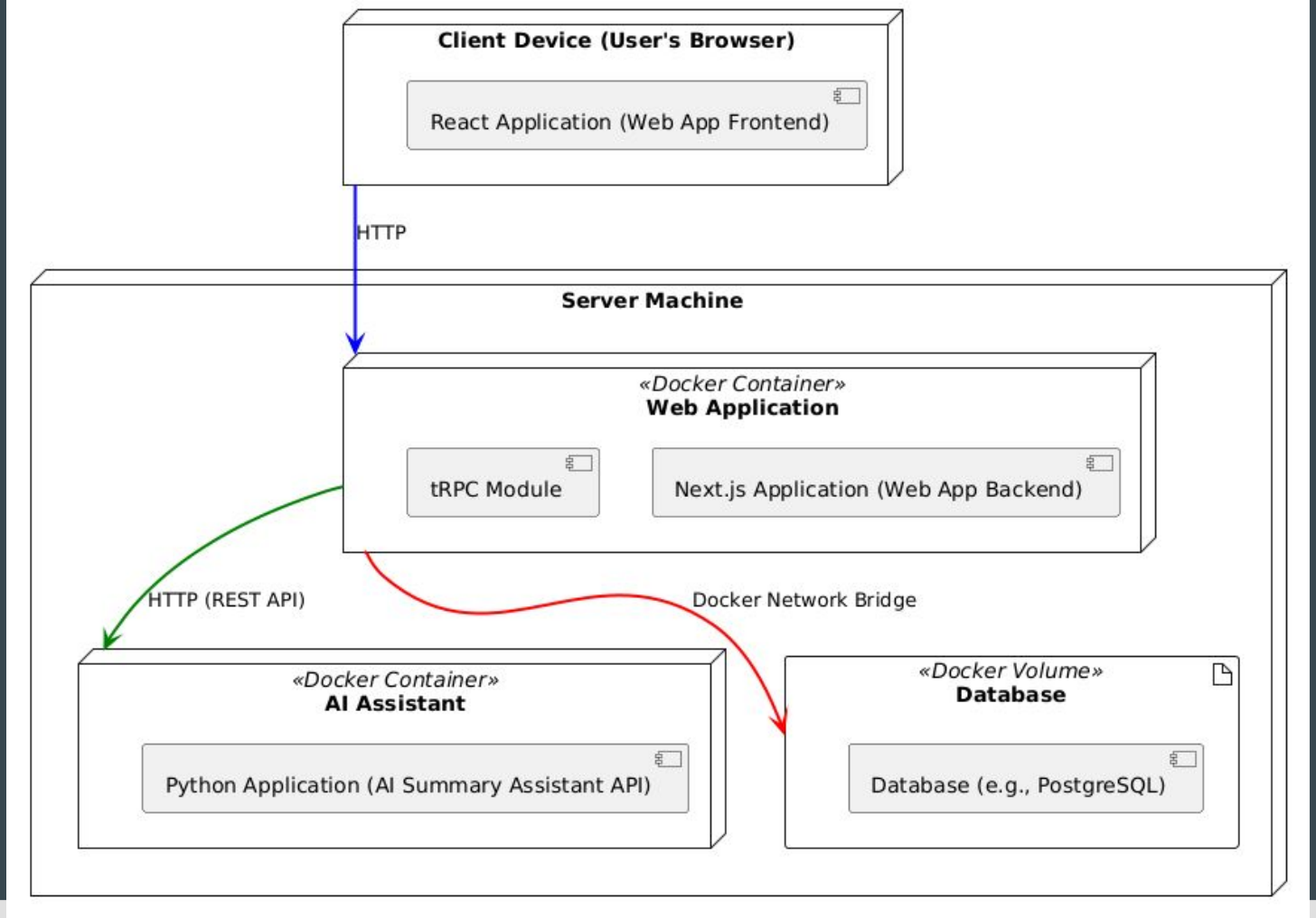



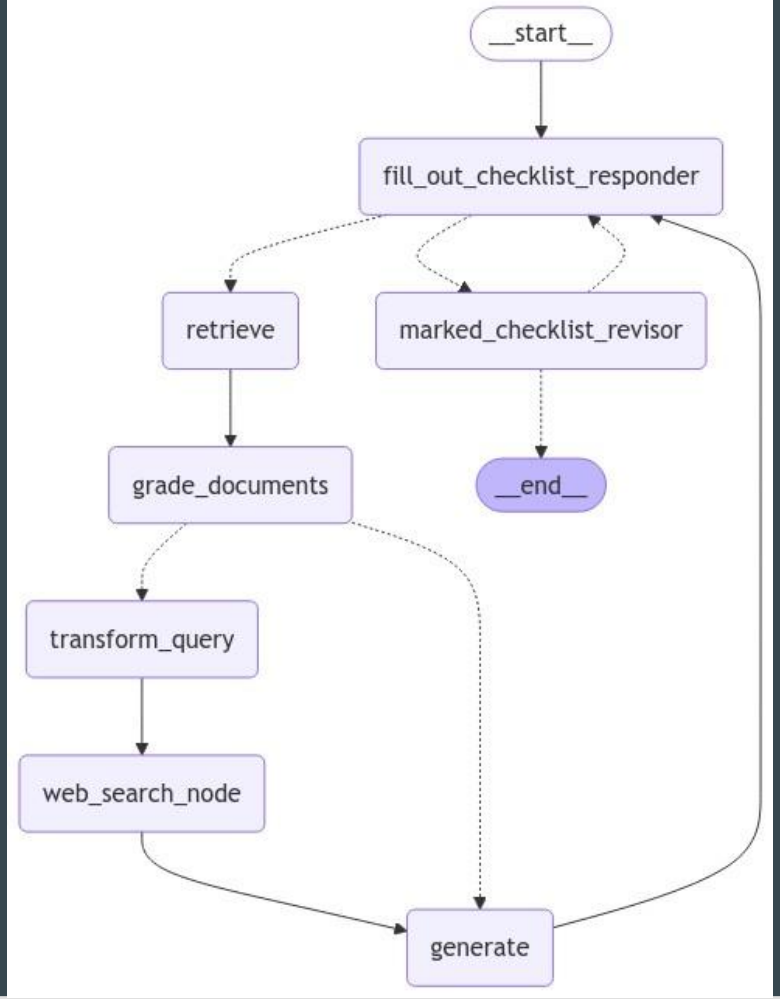

AI Summary Assistant:

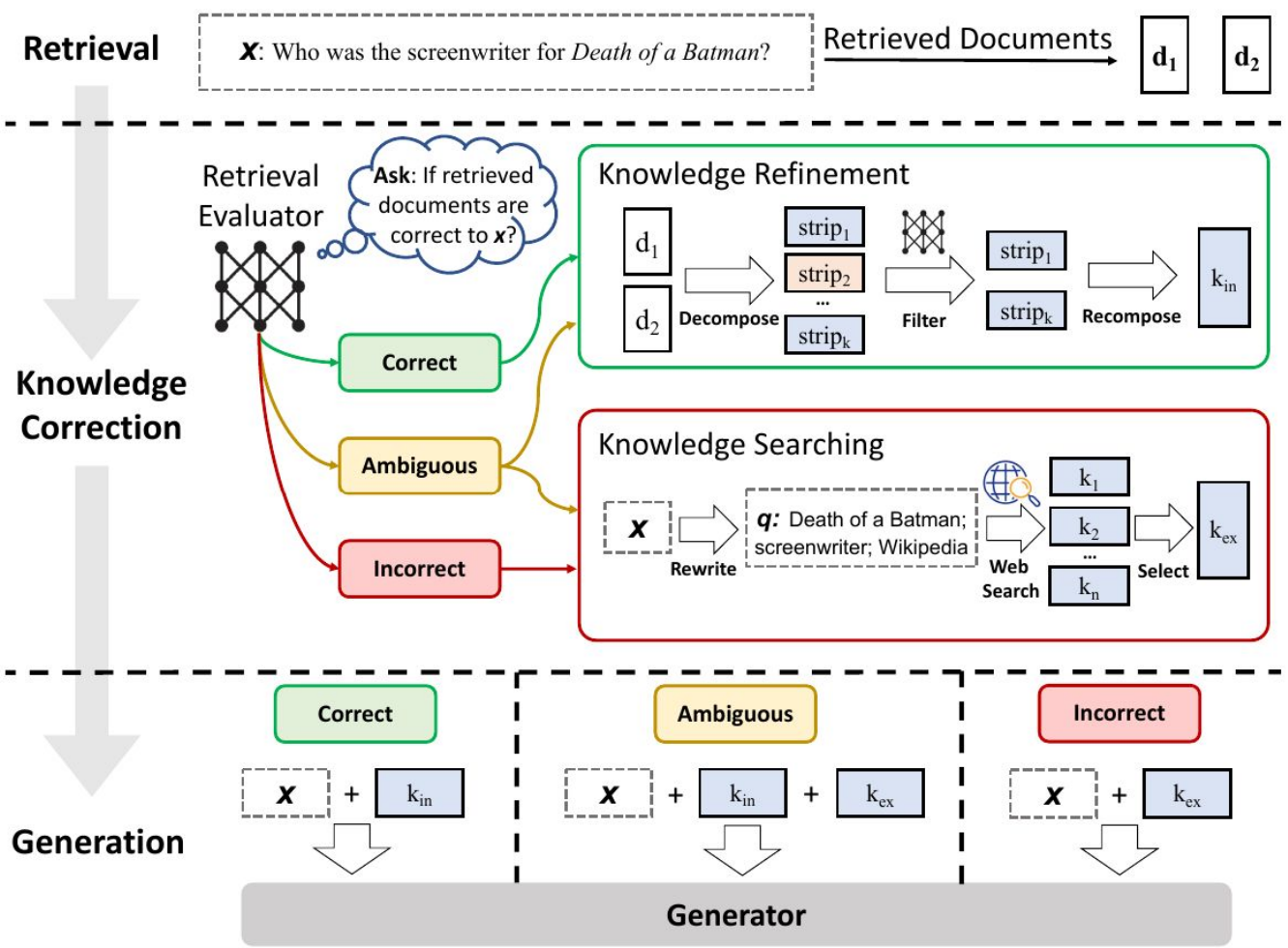


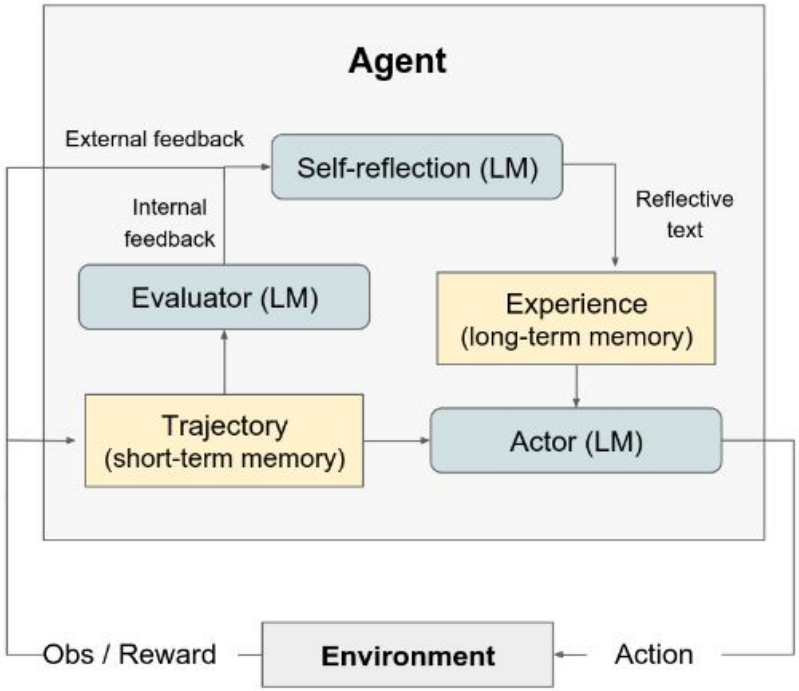



OCR
Tesseract OCR
LLM
LangChain
KartAI
ArkivGPT
KartAI
CADAiD
DBMS
MySQL
API
Summary Assistant
API
PlanChat
Backend
tRPC
Database
Frontend
React application


Client Device (User's Browser)
React Application (Web App Frontend)
HTTP
Server Machine
«Docker Container»
Web Application
tRPC Module
Next.js Application (Web App Backend)
HTTP (REST API)
Docker Network Bridge
«Docker Container»
AI Assistant
Python Application (AI Summary Assistant API)
«Docker Volume»
Database
Database (e.g., PostgreSQL)

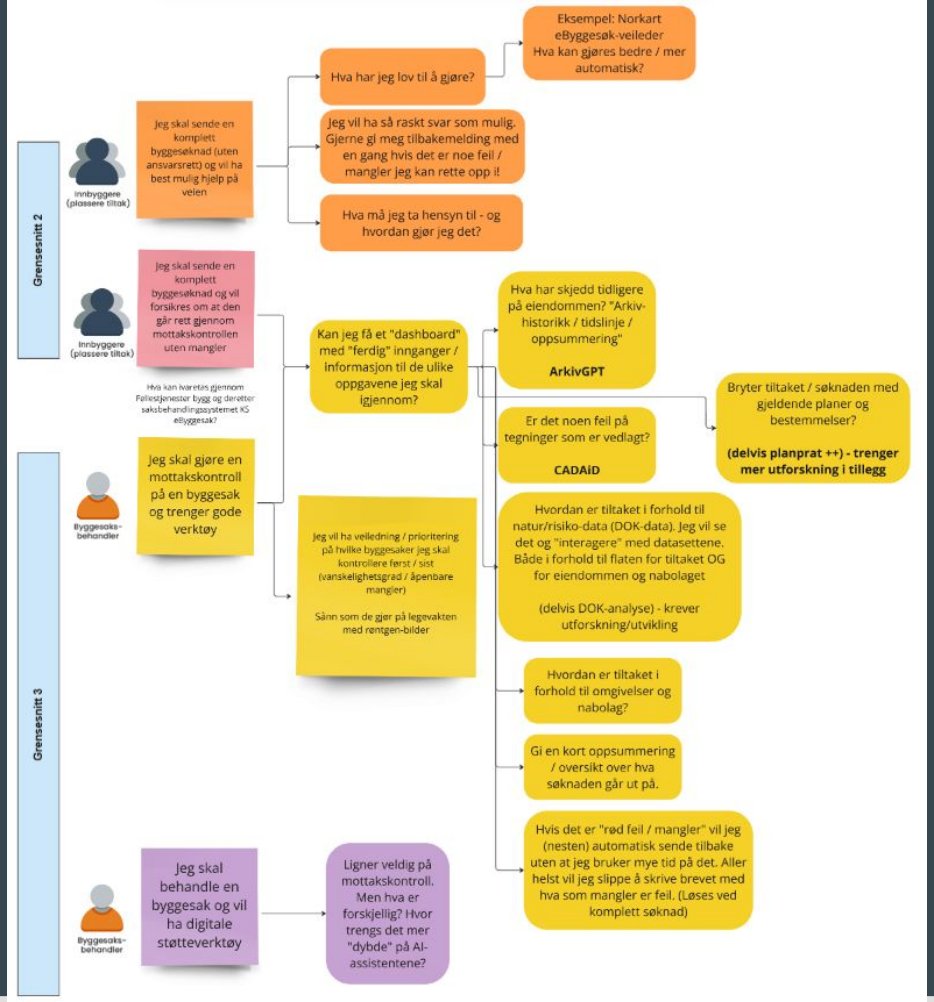

__start__
fill_out_checklist_responder
retrieve
marked_checklist_revisor
grade_documents
__end__
transform_query
web_search_node
generate

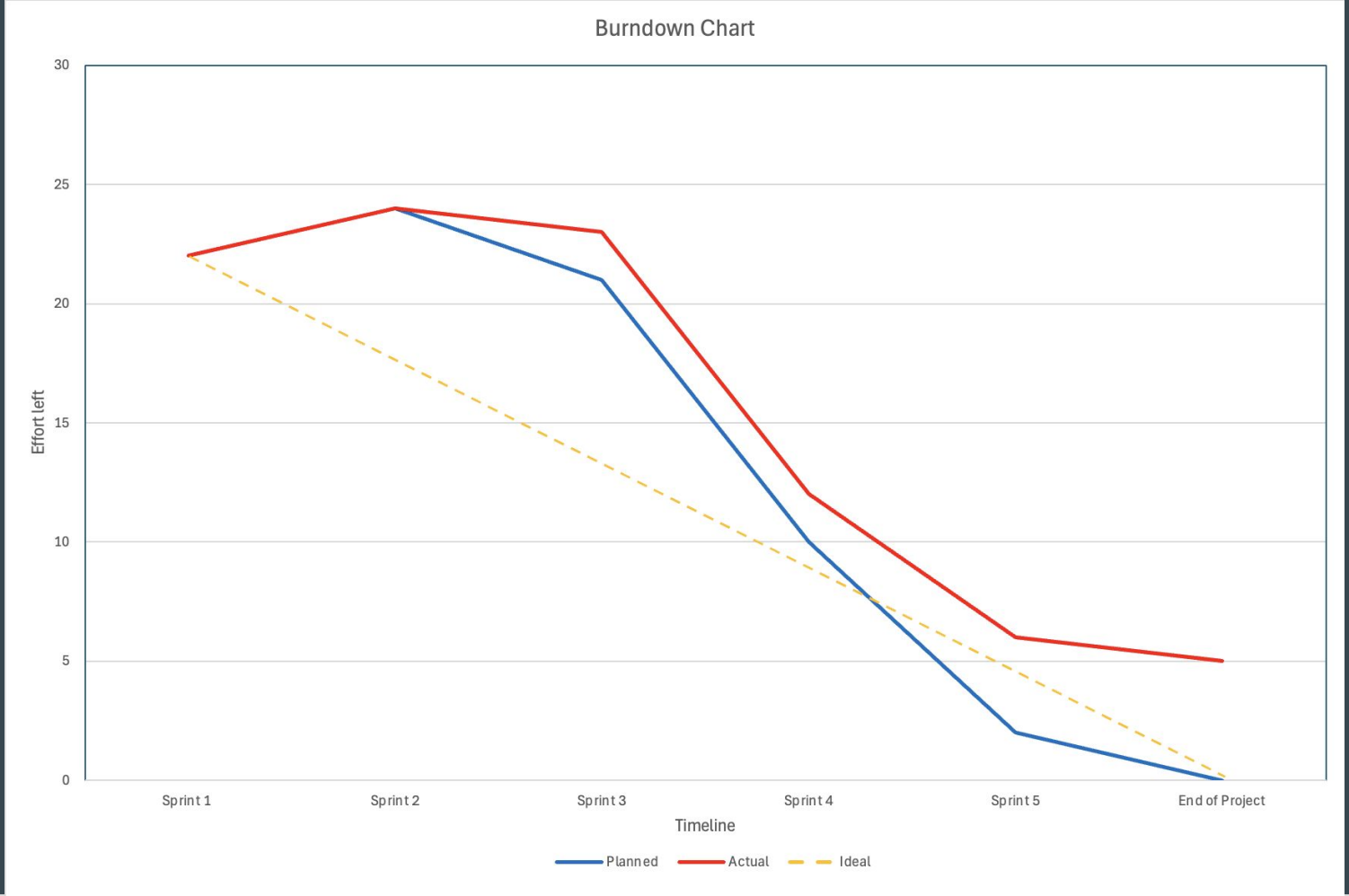

Retrieval
x: Who was the screenwriter for Death of a Batman?
Retrieved Documents
d1
d2
Knowledge Correction
Retrieval Evaluator
Ask: If retrieved documents are correct to x?
Correct
Ambiguous
Incorrect
Knowledge Refinement
d1
d2
Decompose
strip1
strip2
...
stripk
Filter
strip1
stripk
Recompose
kin
Knowledge Searching
x
Rewrite
q: Death of a Batman; screenwriter; Wikipedia
Web Search
k1
k2
...
kn
Select
kex
Generation
Correct
x + kin
Ambiguous
x + kin + kex
Incorrect
x + kex
Generator



Algorithm 1 Reinforcement via self-reflection

Initialize Actor, Evaluator, Self-Reflection: M_a, M_e, M_{sr}
Initialize policy $\pi_\theta(a_i|s_i), \theta = \{M_a, mem\}$
Generate initial trajectory using π_θ
Evaluate τ_0 using M_e
Generate initial self-reflection sr_0 using M_{sr}
Set $mem \leftarrow [sr_0]$
Set $t = 0$
while M_e not pass or $t <$ max trials **do**
    Generate $\tau_t = [a_0, o_0, \ldots a_i, o_i]$ using π_θ
    Evaluate τ_t using M_e
    Generate self-reflection sr_t using M_{sr}
    Append sr_t to mem
    Increment t
end while
return


Grensesnitt 2
Grensesnitt 3
Innbyggere (plassere tiltak)
Jeg skal sende en komplett byggesøknad (uten ansvarsrett) og vil ha best mulig hjelp på veien
Hva har jeg lov til å gjøre?
Eksempel: Norkart eByggesøk-veileder
Hva kan gjøres bedre / mer automatisk?
Jeg vil ha så raskt svar som mulig. Gjerne gi meg tilbakemelding med en gang hvis det er noe feil / mangler jeg kan rette opp i!
Hva må jeg ta hensyn til - og hvordan gjør jeg det?
Innbyggere (plassere tiltak)
Jeg skal sende en komplett byggesøknad og vil forsikres om at den går rett gjennom mottakskontrollen uten mangler
Hva kan ivaretas gjennom Fellestjenester bygg og deretter saksbehandlingssystemet KS eByggesak?
Kan jeg få et "dashboard" med "ferdig" innganger / informasjon til de ulike oppgavene jeg skal igjennom?
Hva har skjedd tidligere på eiendommen? "Arkiv-historikk / tidslinje / oppsummering"
ArkivGPT
Er det noen feil på tegninger som er vedlagt?
CADAiD
Bryter tiltaket / søknaden med gjeldende planer og bestemmelser?
(delvis planprat ++) - trenger mer utforskning i tillegg
Byggesaks-behandler
Jeg skal gjøre en mottakskontroll på en byggesak og trenger gode verktøy
Jeg vil ha veiledning / prioritering på hvilke byggesaker jeg skal kontrollere først / sist (vanskelighetsgrad / åpenbare mangler)
Sånn som de gjør på legevakten med røntgen-bilder
Hvordan er tiltaket i forhold til natur/risiko-data (DOK-data). Jeg vil se det og "interagere" med datasettene. Både i forhold til flaten for tiltaket OG for eiendommen og nabolaget
(delvis DOK-analyse) - krever utforskning/utvikling
Hvordan er tiltaket i forhold til omgivelser og nabolag?
Gi en kort oppsummering / oversikt over hva søknaden går ut på.
Hvis det er "rød feil / mangler" vil jeg (nesten) automatisk sende tilbake uten at jeg bruker mye tid på det. Aller helst vil jeg slippe å skrive brevet med hva som mangler er feil. (Løses ved komplett søknad)
Byggesaks-behandler
Jeg skal behandle en byggesak og vil ha digitale støtteverktøy
Ligner veldig på mottakskontroll. Men hva er forskjellig? Hvor trengs det mer "dybde" på AI-assistentene?


Burndown Chart
30
25
20
15
10
5
0
Effort left
Sprint 1
Sprint 2
Sprint 3
Sprint 4
Sprint 5
End of Project
Timeline
Planned
Actual
Ideal