Vega - AI Agent Engineer Take-Home Exercise

Introduction

Welcome! Thank you for your interest in the AI Agent Engineer position at Vega. Our platform is dedicated to helping Private Equity (PE) firms streamline their complex operations. A significant opportunity for innovation lies in automating and assisting with the review process for investor subscription documents, specifically the questionnaires submitted by potential investors (subscribers).

This take-home exercise is designed to simulate a core task you might encounter in this role: building an AI agent to assist with the review of these subscription questionnaires. It's an opportunity for you to showcase your problem-solving abilities, engineering practices, and your approach to building practical AI solutions that can learn and adapt.

We estimate this exercise should take approximately 2-4 hours of focused work. You are free to use any programming languages, frameworks, tools, or libraries you are most comfortable and productive with, including AI code generation such as Chat GPT or similar.

The Task: Build a Questionnaire Review Agent

Your primary goal is to design and build a prototype AI agent system capable of processing PE fund subscription questionnaires. Based on the content and completeness of a given questionnaire (provided as structured data), the agent must decide on one of the following three actions:

1. **Approve:** The questionnaire appears complete, consistent, and meets all predefined basic requirements.
2. **Return to Subscriber:** The questionnaire is missing specific, required information. The agent must clearly identify exactly which pieces of information are missing or incomplete.
3. **Escalate to Human Review:** The questionnaire contains ambiguous information, potential inconsistencies, fails certain checks (e.g., accreditation status), or presents details requiring nuanced judgment that goes beyond simple rule-based checks. The agent should ideally provide a concise reason for the escalation.

The goal is to be able to automate as many decisions as possible, while not increasing the number of forms that need to be returned manually.

The Learning Aspect: A crucial aspect of building agents at Vega Investments is designing them to improve over time with human oversight. Your solution should consider, and ideally demonstrate a basic mechanism for, how the agent could learn from feedback provided by human reviewers. For example, if a human corrects an agent's decision (e.g., escalates a document the agent approved), how could this feedback be incorporated to make the agent more accurate in the future?

Provided Resources

1. **Sample Questionnaire Data:** You will be provided with a set of sample subscription questionnaires in JSON format. These are simplified and anonymized representations derived from more complex internal structures. Each record represents one questionnaire. (See Appendix A for the simplified structure and sample records used for this exercise).
2. **Basic Review Criteria (Simplified):** For this exercise, assume the following criteria:
    - **Required Fields:** `investor_name`, `investor_address`, `investment_amount`, `is_accredited_investor`, `signature_present`, `tax_id_provided`. All must have non-null/non-empty values.
    - **Investment Amount Check:** Must be a positive number.
    - **Accreditation Check:** If `is_accredited_investor` is `false`, the questionnaire should generally be escalated. If `true`, it's usually acceptable unless the `accreditation_details` field contains ambiguous or concerning language (which might also trigger escalation).
    - **Signature Check:** `signature_present` must be `true`.
    - **Tax ID Check:** `tax_id_provided` must be `true`.
    - **Ambiguity:** Fields like `source_of_funds_description` or `accreditation_details` might contain text requiring interpretation. Vague or potentially problematic descriptions should lead to escalation e.g. "Funds come from selling football tickets on the black market".

Suggested Pattern / Architecture (Consider this a guideline, not a strict requirement)

While you have the freedom to design your solution, here’s a potential workflow to

consider:

1. **Input Processing:** Load and parse the simplified JSON questionnaire data. Handle potential data format issues gracefully.
2. **Agent implementation**
    - **Rule-Based Validation:** Implement logic to perform deterministic checks based on the `Basic Review Criteria`. This layer can quickly identify missing fields (for "Return") or clear violations (e.g., `is_accredited_investor: false` leading to "Escalate", `tax_id_provided: false` leading to "Return").
    - **NLP/ML Analysis:** For interpreting free-text fields (`source_of_funds_description`, `accreditation_details`), simple rule checks might be insufficient. Consider techniques such as:
        i. **Keyword/Pattern Matching:** Look for specific terms or patterns indicating completeness, ambiguity (e.g., "various", "TBD"), or potential issues.
        ii. **Basic Text Classification:** Use a simple model (if feasible within the scope) to classify text snippets as 'clear', 'ambiguous', 'needs review'.
        iii. **Entity Recognition:** Identify key entities if necessary (though perhaps less critical for this specific simplified task).
    - **Decision Engine:** Combine the outputs from the validation and analysis steps. Implement clear logic that maps the findings to one of the three final actions (Approve, Return, Escalate).
        i. If Returning: Ensure the output clearly lists all missing/invalid fields.
        ii. If Escalating: Provide a concise reason (e.g., "Ambiguous source of funds", "Non-accredited investor", "Incomplete accreditation details").
    - **Feedback Loop / Learning Mechanism:** Design and document (or implement a basic version of) how human feedback on the agent's decisions could be captured and utilized for improvement. Examples:
        i. A mechanism to log agent decisions alongside human corrections.
        ii. A process description for how this logged data could be used to refine rules (e.g., add new keywords that trigger escalation).
        iii. A plan for periodically retraining any ML models using the corrected data as labels.
3. **Output:** The agent should output its decision (Approve/Return/Escalate) and any associated reasoning or list of missing fields for each input questionnaire

processed. A structured output format (like JSON or CSV) is preferred.

Expected Deliverables

1. **A project overview:** An overview of the solution you have built, the assumptions and considerations you have taken into account and any other important information that you have considered.
2. **Recording of your agent:** Provide a brief video of your agent processing the sample data.
3. **Source Code:** Your complete, runnable code implementing the agent system. Please ensure it is well-structured and commented. Include clear instructions on how to set up any dependencies and execute your solution. Providing a link to a Git repository (e.g., GitHub, GitLab) is the preferred method.
4. **README.md File:** It should include:
    - Clear, step-by-step instructions to set up the environment (dependencies) and run your code.
    - A detailed explanation of your design choices, the overall architecture, and any libraries/tools used.
    - A description of the logic your agent uses to arrive at each decision type (Approve, Return, Escalate).
    - An explanation of how you approached handling potential ambiguity in the data.
    - If you considered multiple approaches (e.g., rule-based vs. ML), describe why you chose the one you implemented.
    - A detailed description of your proposed (or implemented) learning mechanism and how it would enable the agent to improve over time based on feedback.
    - Any assumptions you made and any known limitations of your prototype.
5. **Output File(s):** The output generated by your agent when run against the provided sample data set.

Submission

Please package your source code, README file, and output file(s) into a single archive (e.g., `.zip` or `.tar.gz`) or provide a link to a version control repository (e.g., GitHub, GitLab) containing these items. We will be running your solution against a set of completed questionnaires. It **must** parse the provided input and provide the output exactly. We look forward to seeing your solution!

Appendix A: Simplified Sample Questionnaire Data Structure and Examples

The complex questionnaire data has been **simplified** for this exercise into the following flat JSON structure. You will receive a file containing a list of objects like these.

Example Input JSON Object Structure:

```
Unset
{

  "questionnaire_id": "string", // Unique identifier

  "investor_name": "string | null", // Full name of the investor
or entity name

  "investor_type": "string | null", // e.g., "Natural Person",
"Entity", "Joint Tenants"

  "investor_address": "string | null", // Primary address

  "investment_amount": "number | null", // Amount intended to
invest

  "is_accredited_investor": "boolean | null", // Derived status
based on questionnaire answers

  "accreditation_details": "string | null", // Text description
summarizing accreditation basis

  "source_of_funds_description": "string | null", // Text
description of fund origins (may require interpretation)

  "tax_id_provided": "boolean | null", // Whether a valid Tax ID
(e.g., SSN, TIN, EIN) was provided

  "signature_present": "boolean | null", // Whether the signature
task/section is completed

  "submission_date": "string" // Date of submission}

}
```

Example Output JSON Object Structure :

```
Unset
[
  {
    "questionnaire_id": "1a59843c-9ade-4b6d-8961-215c44c9ca6a",
    "decision": "Approve",
    "missing_fields": null,
    "escalation_reason": null
  },
  {
    "questionnaire_id": "2b67890d-1bfg-5c7e-9012-326d55d0db7b",
    "decision": "Return",
    "missing_fields": ["investment_amount", "tax_id_provided"],
    "escalation_reason": null
  },
  {
    "questionnaire_id": "3c78912e-2cgh-6d8f-0123-437e66e1ec8c",
    "decision": "Escalate",
    "missing_fields": null,
    "escalation_reason": "Ambiguous source of funds",
  },
...
]
```

Example Records (Derived & Simplified from Full Structure):

Sample 1 (Likely Approve):

```
Unset
{

  "questionnaire_id":
"1a59843c-9ade-4b6d-8961-215c44c9ca6a",

  "investor_name": "Mr and Mrs Simpson",

  "investor_type": "Joint Tenants",

  "investor_address": "25, Springfield, New Jersey, United
States",

  "investment_amount": 250000,

  "is_accredited_investor": true,

  "accreditation_details": "Joint Income over $300k for
past two years with expectation to continue.",

  "source_of_funds_description": "Personal savings and
employment income.",

  "tax_id_provided": true,

  "signature_present": true,

  "submission_date": "2025-04-30"

}
```

Sample 2 (Likely Return - Missing Investment Amount & no Tax ID provided):

```
Unset
{
  "questionnaire_id":
"2b67890d-1bfg-5c7e-9012-326d55d0db7b",
  "investor_name": "Example Corp.",
  "investor_type": "Entity",
  "investor_address": "123 Main St, Anytown, CA 90210,
United States",
  "investment_amount": null,
  "is_accredited_investor": true,
  "accreditation_details": "Entity with total assets in
excess of $5M, not formed for specific purpose.",
  "source_of_funds_description": "Sale of previous
business.",
  "tax_id_provided": false,
  "signature_present": true,
  "submission_date": "2025-05-01"
}
```

Expected Agent Output Hint: Return, Missing: `investment_amount`, `tax_id_provided` Reason: False tax_id_provided

Sample 3 (Likely Escalate - Ambiguous Source of Funds):

```
Unset

{

  "questionnaire_id":
"3c78912e-2cgh-6d8f-0123-437e66e1ec8c",

  "investor_name": "Investor Three",

  "investor_type": "Natural Person",

  "investor_address": "456 Oak Ave, Sometown, TX 75001,
United States",

  "investment_amount": 750000,

  "is_accredited_investor": true,

  "accreditation_details": "Holds Series 7, 65 licenses in
good standing.",

  "source_of_funds_description": "Various sources including
family contributions.",

  "tax_id_provided": true,

  "signature_present": true,

  "submission_date": "2025-05-01"

}
```

Expected Agent Output Hint: Escalate, Reason: Ambiguous source of funds description.

Sample 4 (Likely Return - Missing Signature & Address):

```
Unset

{
  "questionnaire_id": "4d89023f-3dhi-7e9g-1234-548f77f2fd9d",
  "investor_name": "Investor Four Trust",
  "investor_type": "Trust",
  "investor_address": null,
  "investment_amount": 100000,
  "is_accredited_investor": true,
  "accreditation_details": "Trust with total assets over $5M, directed by sophisticated person.",
  "source_of_funds_description": "Investment portfolio.",
  "tax_id_provided": true,
  "signature_present": false,
  "submission_date": "2025-05-02"
}
```

Expected Agent Output Hint: Return, Missing: `investor_address`, `signature_present`

Sample 5 (Likely Escalate - Not Accredited):

```
Unset
```


```
{
  "questionnaire_id": "5e90134g-4eij-8f0h-2345-659g88g3ge0e",
  "investor_name": "Investor Five",
  "investor_type": "Natural Person",
  "investor_address": "789 Pine Rd, Villagetown, FL 33301, United States",
  "investment_amount": 50000,
  "is_accredited_investor": false,
  "accreditation_details": "Does not meet income or net worth requirements.",
  "source_of_funds_description": "Savings.",
  "tax_id_provided": true,
  "signature_present": true,
  "submission_date": "2025-05-02"
}
```


Expected Agent Output Hint: Escalate, Reason: Investor is not accredited.