Prometheus Design and Documentation

Prometheus follows a four step process, very similar to Tropos.

1. Goals: goals and subgoals are defined.
2. Scenario: construct scenarios, break into steps and derive roles from it
3. Architectural Design: from roles, define what type of data needs to be stored as beliefs and which agents are required
4. Detailed Design: details agents and their capabilities

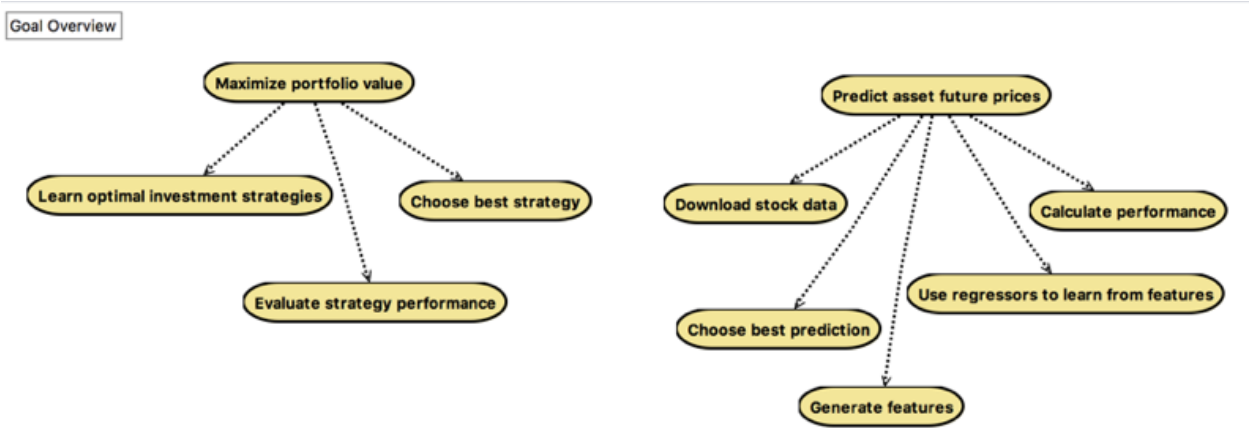
The following process is described below for the multiagent system being designed:

Goals

The following are the goals and subgoals of the MAS:

Maximize portfolio's value
- Learn optimal investment strategies
- Evaluate performance
- Choose best strategy

Predict asset prices
- Download and wrangle stock data
- Generate features
- Create a regressor and learn from features
- Calculate performance with backtesting
- Choose best prediction



Scenarios

There are two scenarios:

- User inquiries best strategy to maximize capital value of a given portfolio (pick best strategy)
- User inquiries future asset prices for a given stock and period (predict future stock prices)

They can be divided into the following steps:

User inquiries best strategy to maximize capital value of a given portfolio

1. Strategies Learning
2. Choose best strategy
3. Present strategy to user

User inquiries future stock prices for a given stock and period

1. Data collected
2. Regressors learning
3. Choose best regressor
4. Regressor predicts
5. Present future prices to user

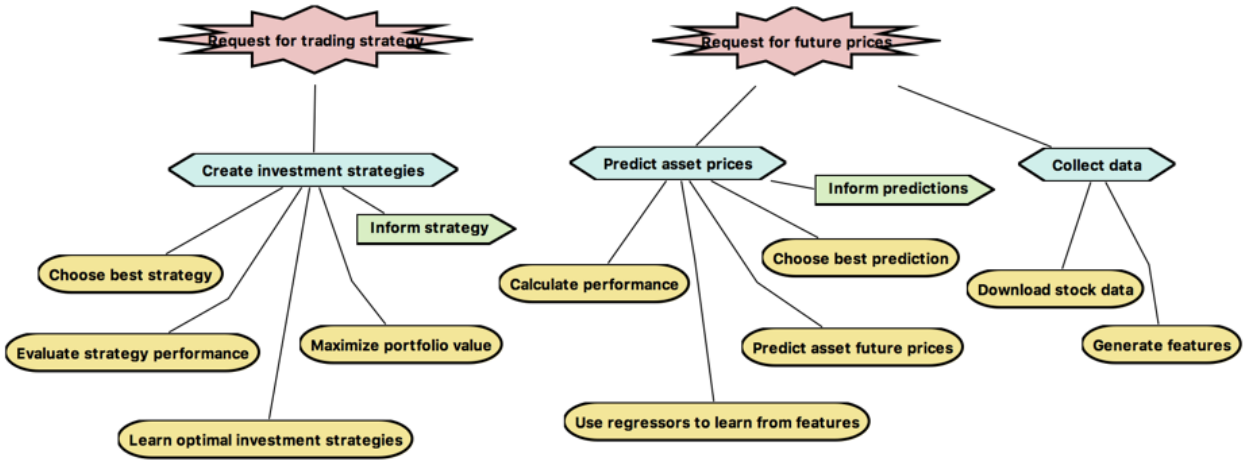
Percepts:

- Request for trading strategy
- Request for future prices

Roles identified from the system functionalities:

- Predict asset prices
- Create investment strategies
- Collect data

System Role Overview



Architectural design

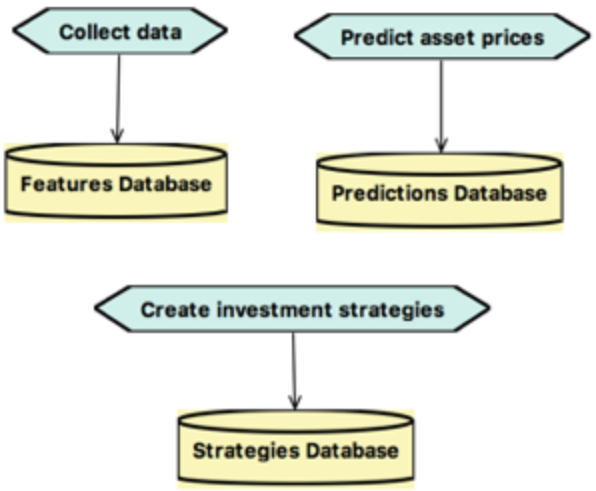
Identify types of data that needs to be stored in the system as beliefs:

Role: Predict stock prices
- Predictions Database

Role: Create investment strategies
- Strategies Database

Role: Collect data
- Features Database

Data Coupling Overview

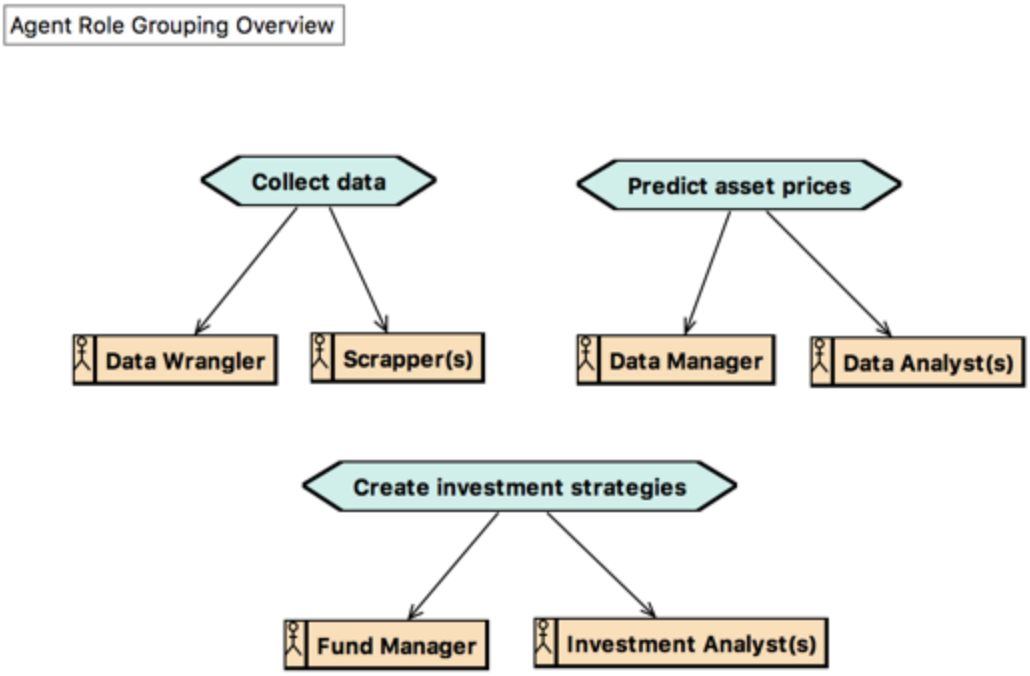


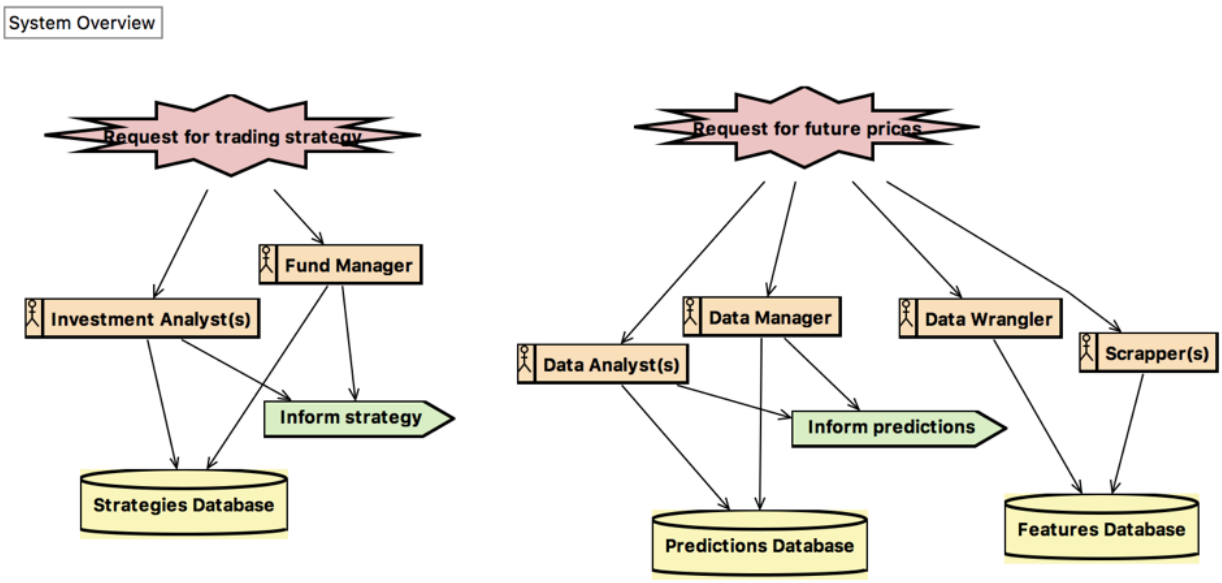
Identify agents:

Role: Predict stock prices
- Data Manager
- Data Analysts

Role: Create investment strategies
- Fund Manager
- Investment Analysts

Role: Collect data
- Data Wrangler (Manager)
- Scrappers



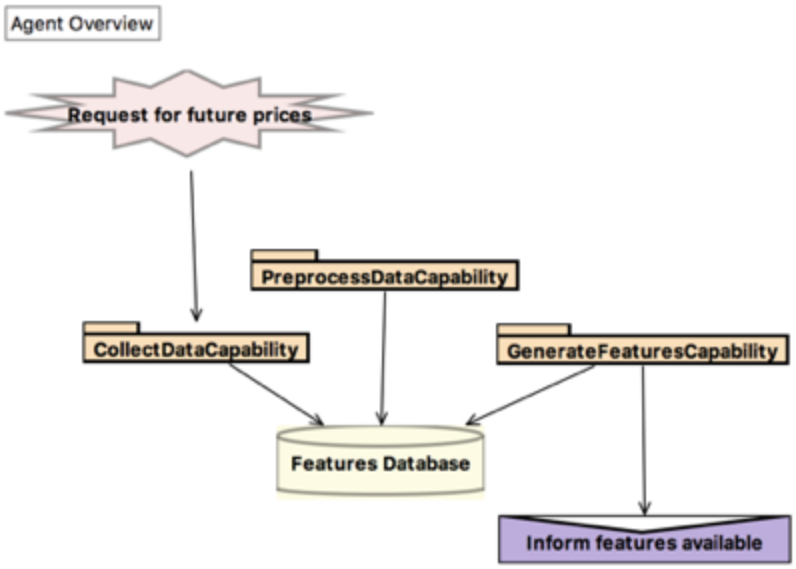


Detailed Design

Detail agents and their capabilities

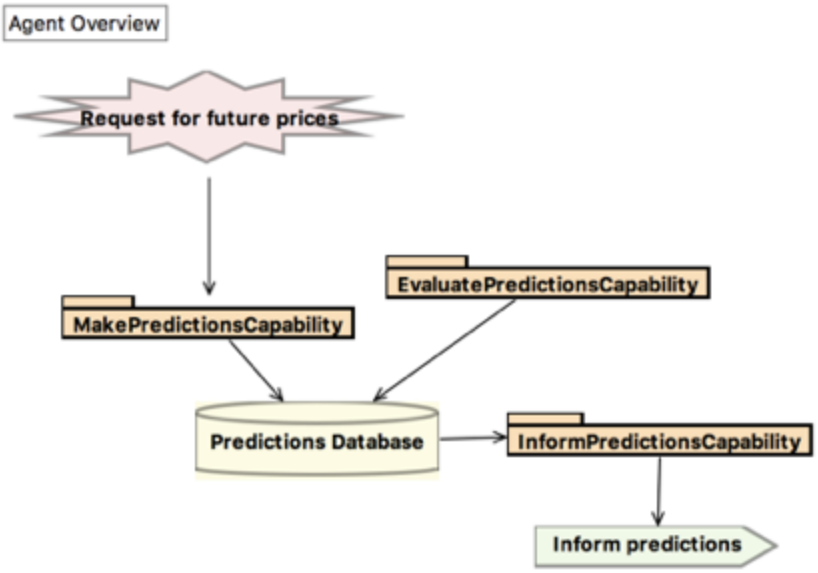
Role: Collect data

- Data Wrangler (Manager)
    - Preprocess data capability
    - Generate features capability
- Scrappers
    - Collect data capability



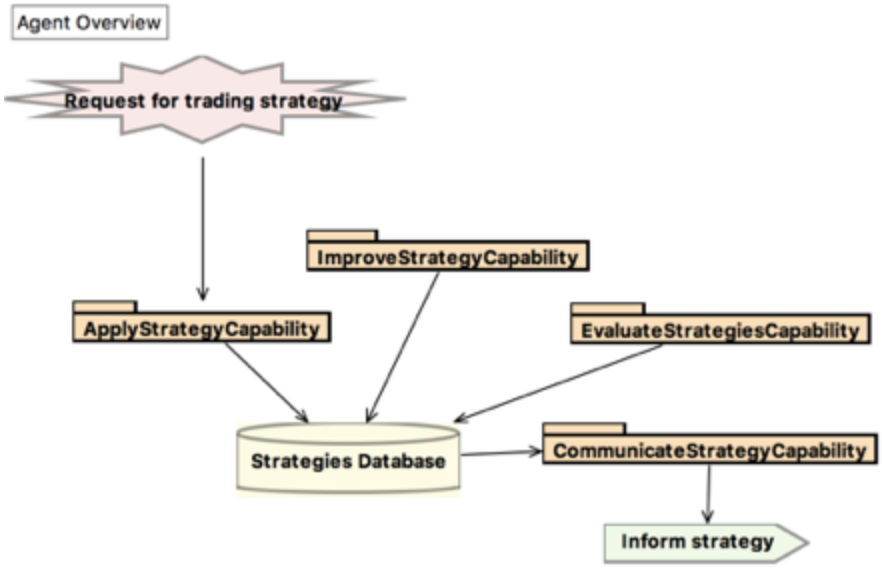
Role: Predict stock prices

- Data Manager
    - Evaluate predictions capability
    - Inform predictions capability
- Data Analysts
    - Make predictions capability



Role: Create investment strategies

- Fund Manager
    - Evaluate strategies capability
    - Communicate strategy capability
- Investment Analysts
    - Apply strategy capability
    - Improve strategy capability



PEAS

The environment of the multiagent system is a virtual container, hosted in a server or in multiple servers, and can be described as follows:

- Multiagent: there are multiple agents interacting in the environment.
- Stochastic or Non Deterministic: The next state can not be determined from the actual state and the choices made by the agent. For example an investment strategy may have a different outcome than predicted, and the data wrangler might fail to retrieve data of an asset due to unexpected changes in the environment.
- Sequential: the actual state of the environment is affected by the previous actions taken by the agent.
- Dynamic: the current state of the environment may change while the agent deliberates the investment plan, which requires use of online learning methods in the reinforcement learning applied at the portfolio optimization layer.
- Continuous: the environment has an infinite number of states, since it is in constant growth.
- Partially known: the agent does not have full knowledge of its action in the environment.

The characteristics of the environment are typical of open world scenarios. Non deterministic and dynamic scenarios pose an extra challenge when implementing the agent's means ends reasoning. More information in the environment specific to each agent is given below, in the agent's PEAS model.

Data Wrangler and Scrapers

- Performance Measure: Performance is measured by the amount of data collected. The more data collected and features derived from data, better the performance of the data wrangler
- Environment: Its environment is electronic only. It interacts with the world wide web through web scraping and API usage.
- Actuators: It does not act on the environment
- Sensors: It can perceive significant changes in the data through continuous monitoring of its sources.

Data Manager and Data Analysts

- Performance Measure: Performance of regressors is measured by supervised learning algorithm metrics, specially R2 score, which is a measure of how much of the variance in the data can be explained by the learning algorithm

- Environment: Its environment is electronic only. It has no direct interaction with the outside world, only with the other agents in the multiagent system
- Actuators: It does not act directly on the environment
- Sensors: It has no interface with the environment. It acts upon request from the Fund Manager only.

Fund Manager and Investment Analysts

- Performance Measure: Performance is measured by the total portfolio value after X period. Different strategies are compared between them and against baseline market parameters, such as IBOVESPA index for Brazilian stock market.
- Environment: Its environment is electronic only. It interacts with users through an input interface, that accepts requests, and output interface that outputs strategies and predictions.
- Actuators: Its only action in the outside world is to output plans, plots and other formats of data.
- Sensors: It does not monitor the environment, as of now. It acts upon request only and its plans and strategies are based from data collected by the Data Wrangler and saved to the beliefs database.