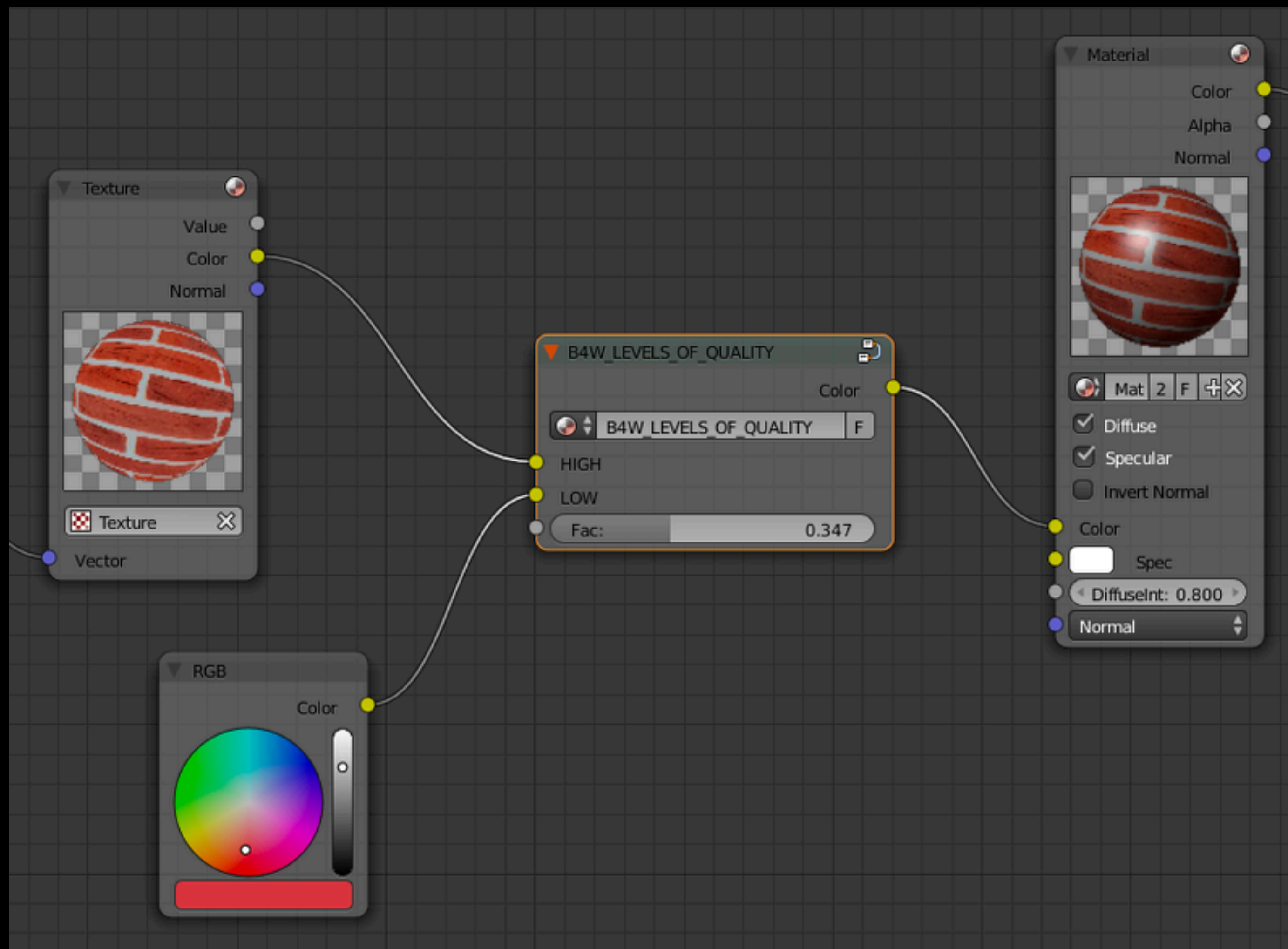


# Описание конструктора стратегии

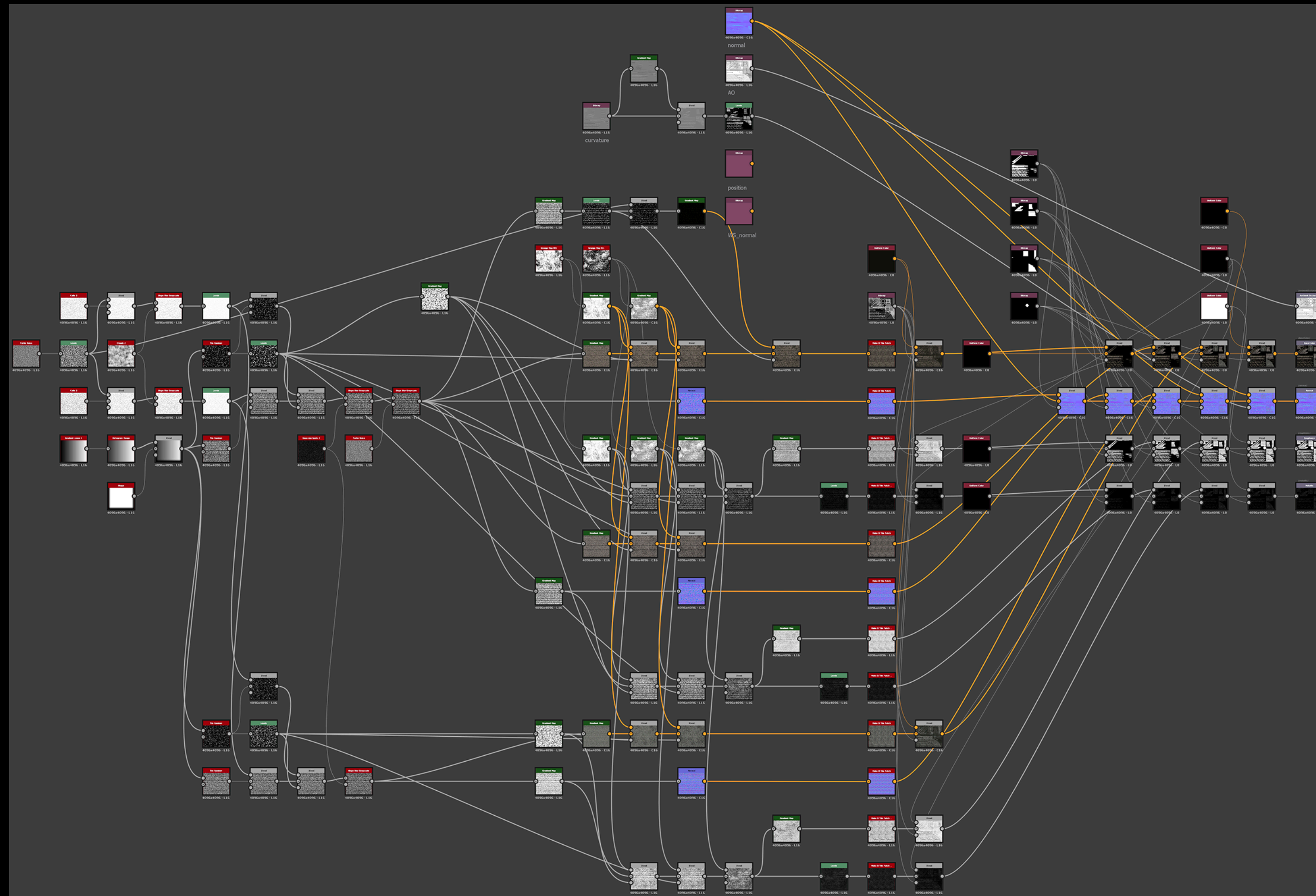
# Нодовая архитектура(Blueprint)

Я знаю только примеры из ПО для моделирования, создания эффектов



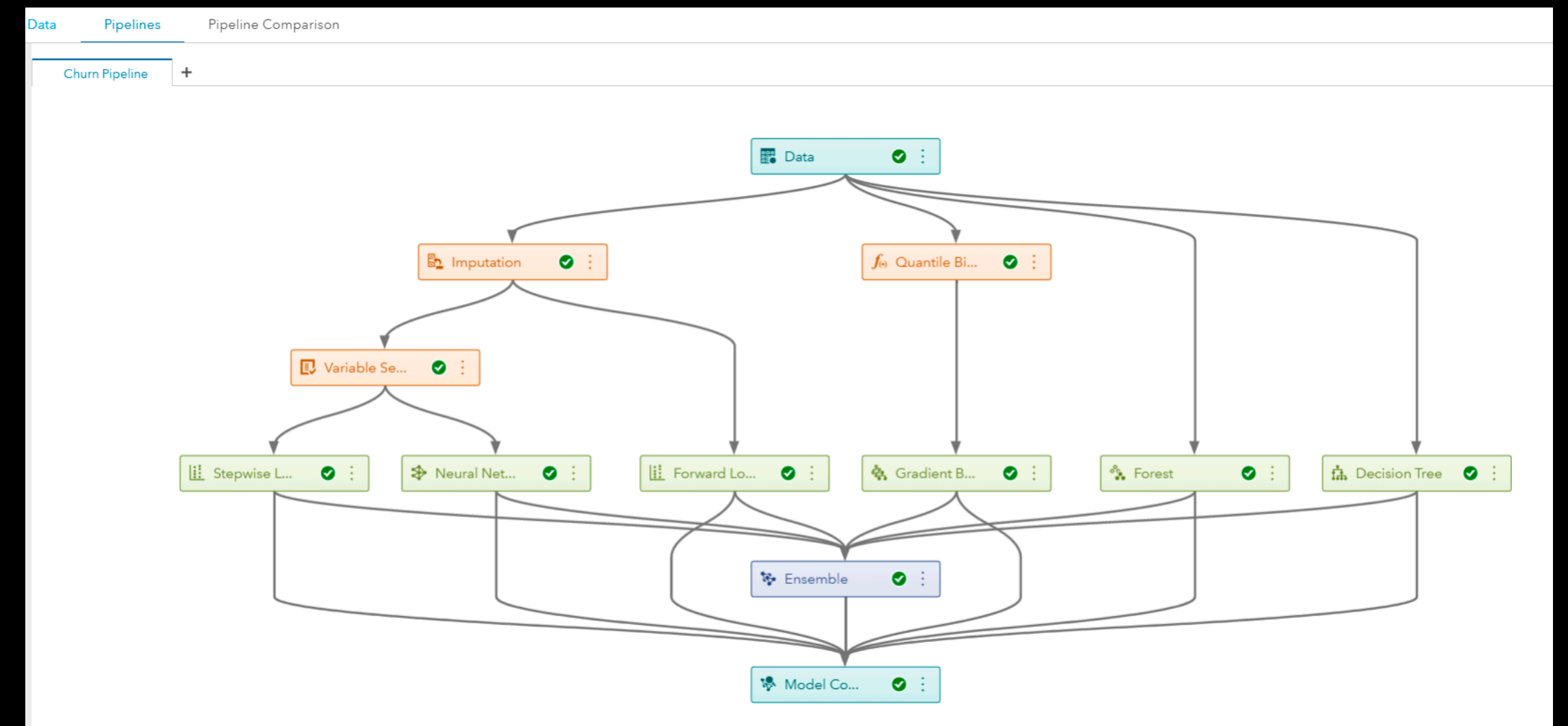
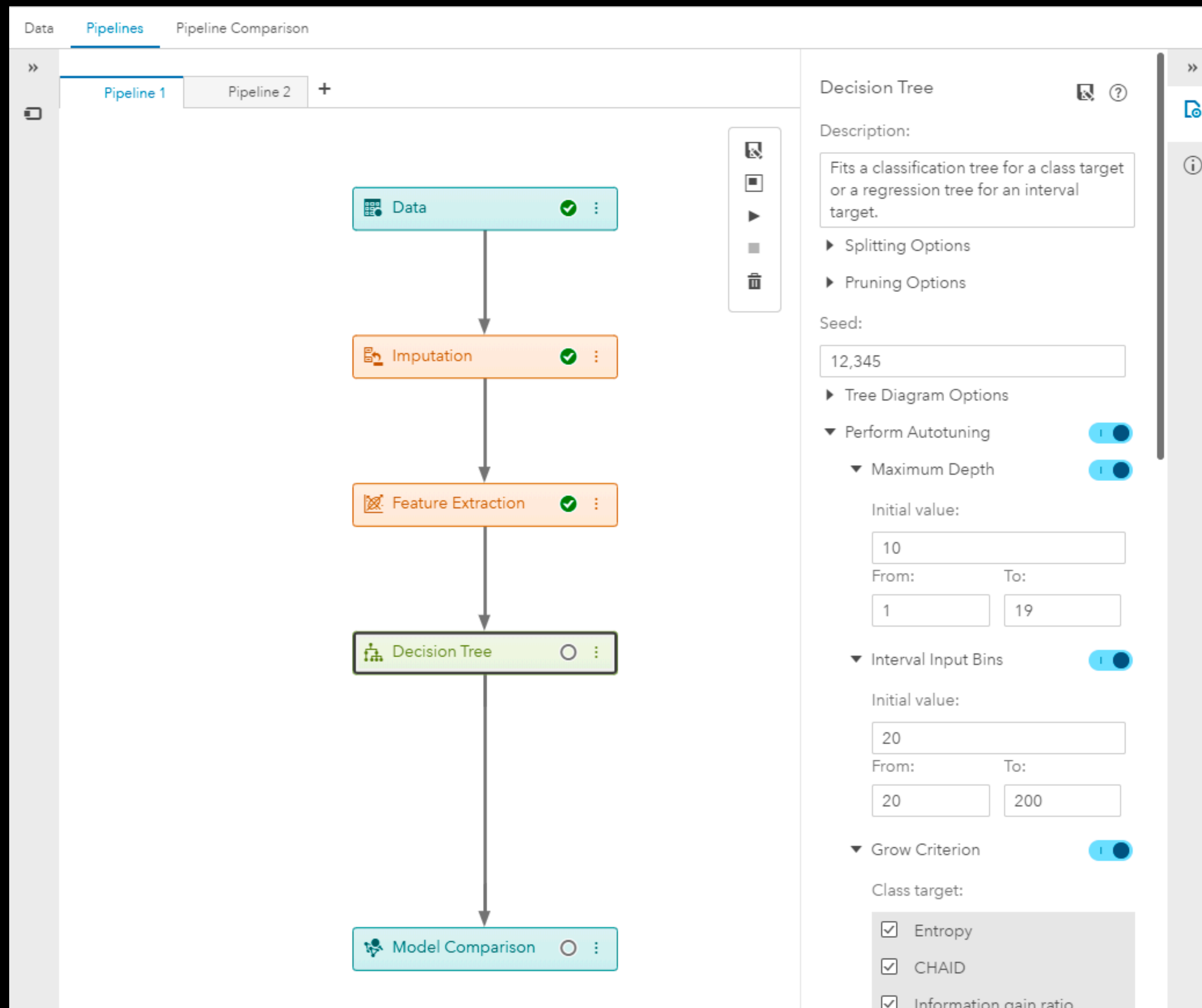
# Основное преимущество

Создав систему взаимодействия можно значительно ускорить процесс написания стратегии путем повторного использования готовых модулей без переписывания коннекторов между ними. Если реализовать систему унифицированной связи блоков, то каждое добавление нового блока отвечающего за независимый функционал будет создавать множество новых комбинаций использований (не всегда рациональных)



# Где есть что-то похожее

SAS VIJA: [https://www.sas.com/ru\\_ru/software/viya.html](https://www.sas.com/ru_ru/software/viya.html)





# Как это должно выглядеть

Я представляю себе реализацию как разбиение на разные смысловые блоки (ноды/Nodes). Примеры таких групп

Стратегия должна собираться из различных элементарных блоков. Каждый блок принадлежит к определенной группе, выполняющей свои функции.

Группа функций  
выбора горизонта  
стратегии

Группа функций  
выполнения  
стратегии

Группа  
вспомогательных  
функций

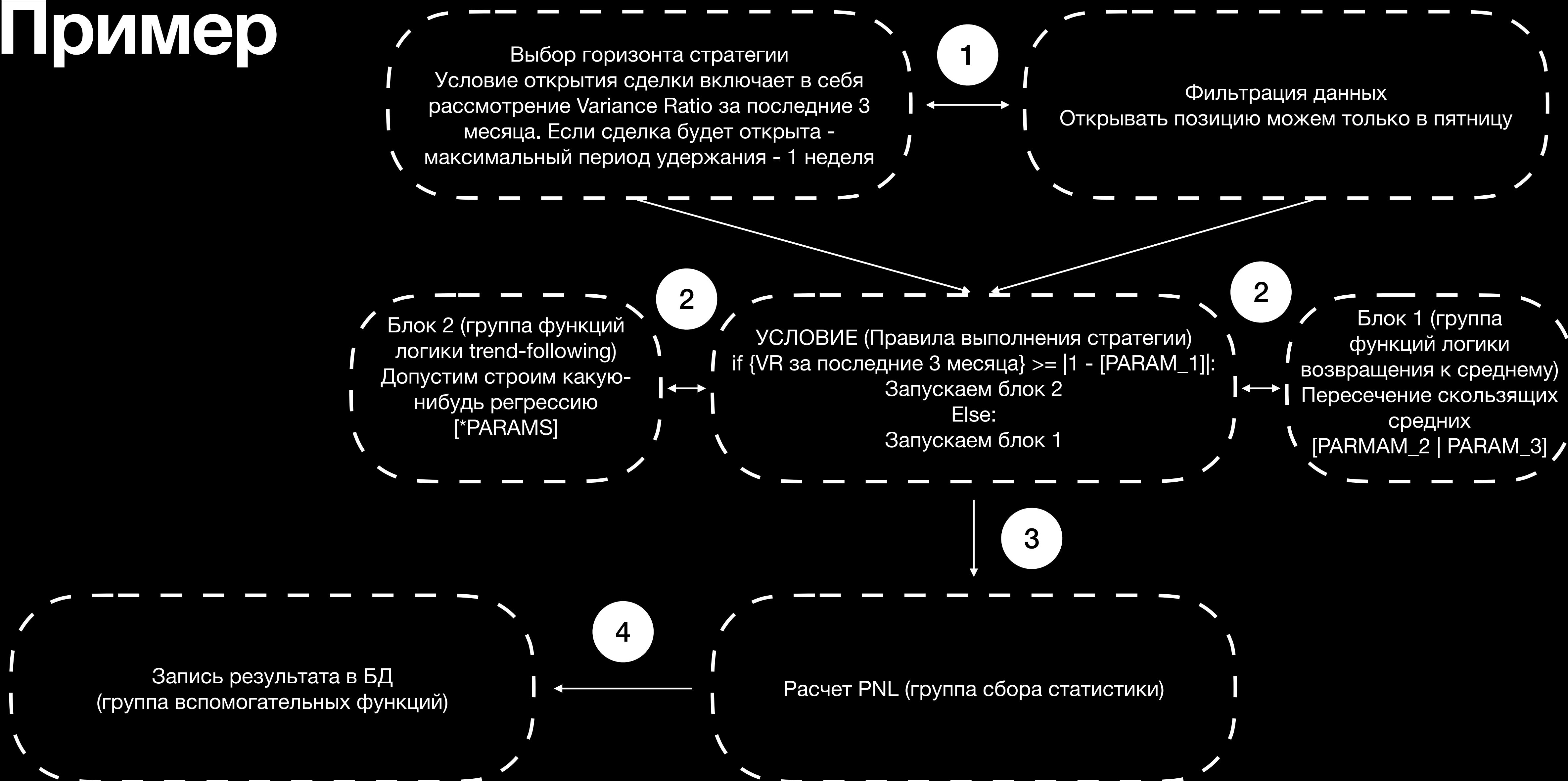
Группа функций  
фильтрации данных

Группа функций  
создания  
статистики

# Пример

Данные итеративно разбиваются на «каретки» длиной 3 месяца + 1 неделя.

Сохранится только те которые проходят условие фильтра



# В чем проблема?

Связи между модулями ① ② ③ ④ должны быть определены

в конструкторе стратегий. Те написав дополнительный Блок-і (связь 2) можно без изменения кода запустить его выполнение в блоке с условием (Правила выполнения стратегий). Возможно добавить условие выполнения Блок\_3; возможно заменить Блок\_1 на Блок\_3. В тоже время без переписывания связи 3 можно изменить отчет в формате PNL к примеру на отчет по markdown. Логично что связь 3 должна уметь передавать данные для создания отчета из композиции требуемых результатов.

# Очень неправильный пример того как я делал сейчас

## На примере стратегии FX.

1

### SELECTION\_LOGIC MODULE

(создает данные на горизонте которых может существовать стратегия)

2

### FX\_OPENING\_TRADE MODULE

(выполняет функцию обнаружения допустимых для открытия позиции точек)

4

### FX\_HOLD\_LOGIC MODULE

(определяет правила удержания позиции)

5

### CALCULATE\_RETURN MODULE

(определяет return сделанной сделки)

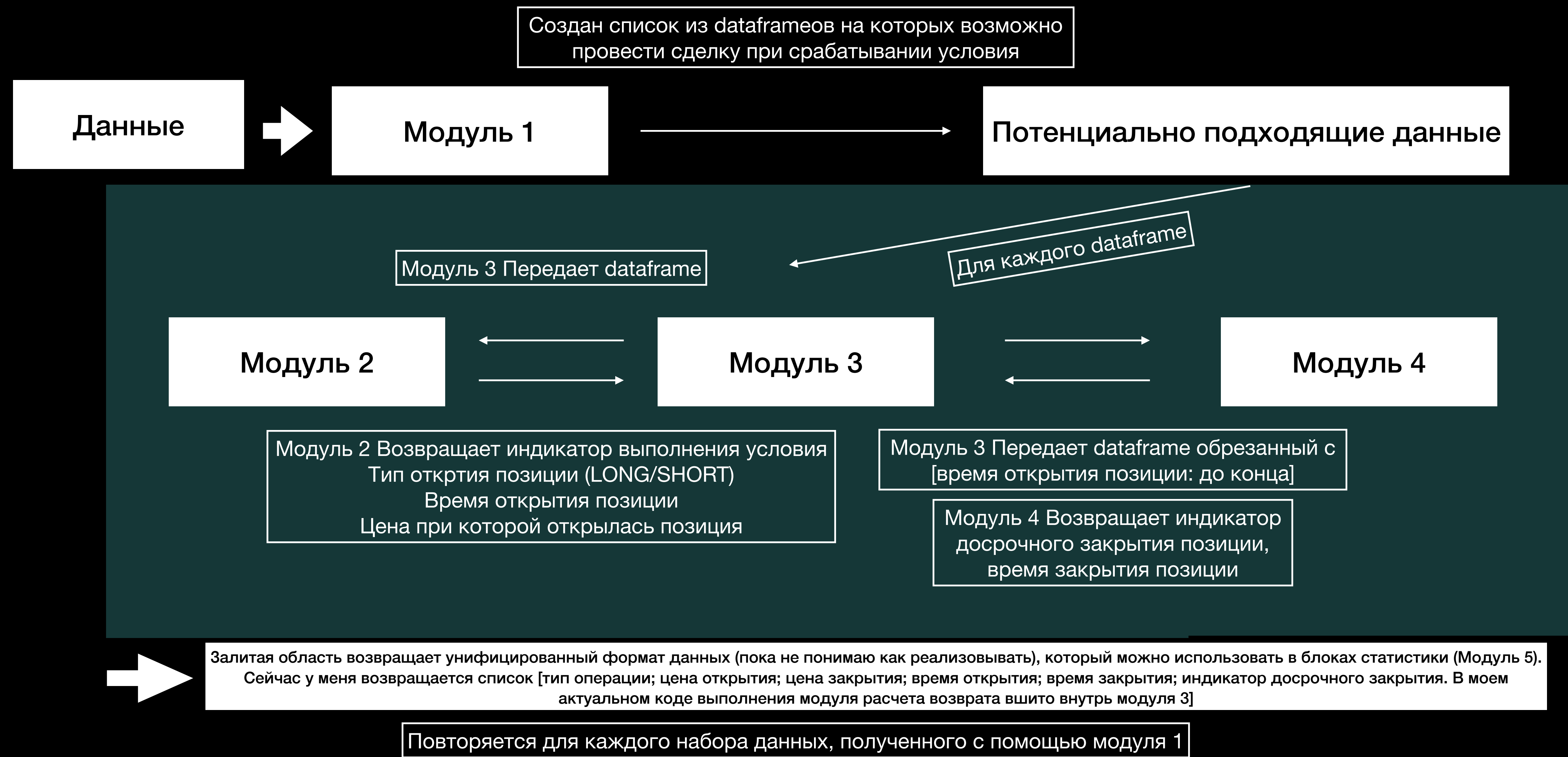
3

### BUYING\_TRIGER MODULE

(является центральным узлом, связывающим модули выполнения между собой. Содержит в себе унифицированный формат данных.)



# Как выглядит последовательность выполнения



# Как бы это должно было выглядеть

