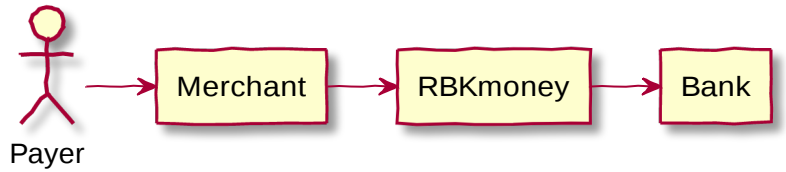
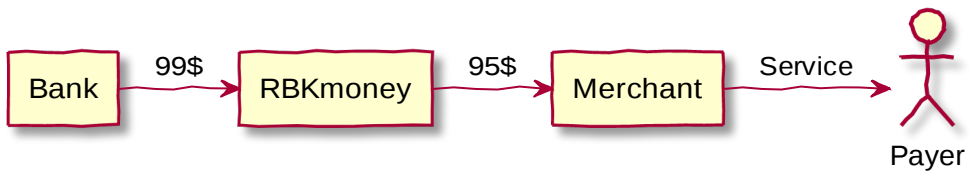
Автоматный подход при разработке бизнес-логики платёжного процессинга

Бизнес логика процессинга RBK Money

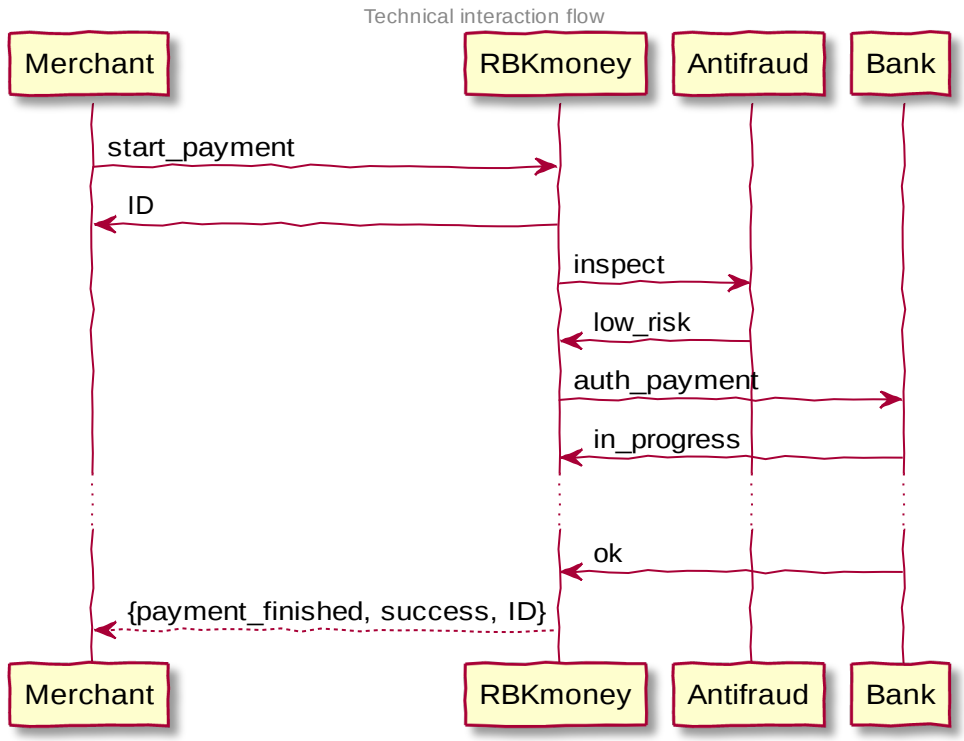
Technical interaction flow



Money and service interaction flow



Бизнес логика процессинга RBK Money



Какие требования стояли перед процессингом

Какие требования стояли перед процессингом

Принимать платежи:

- консистентность

Какие требования стояли перед процессингом

Принимать платежи:

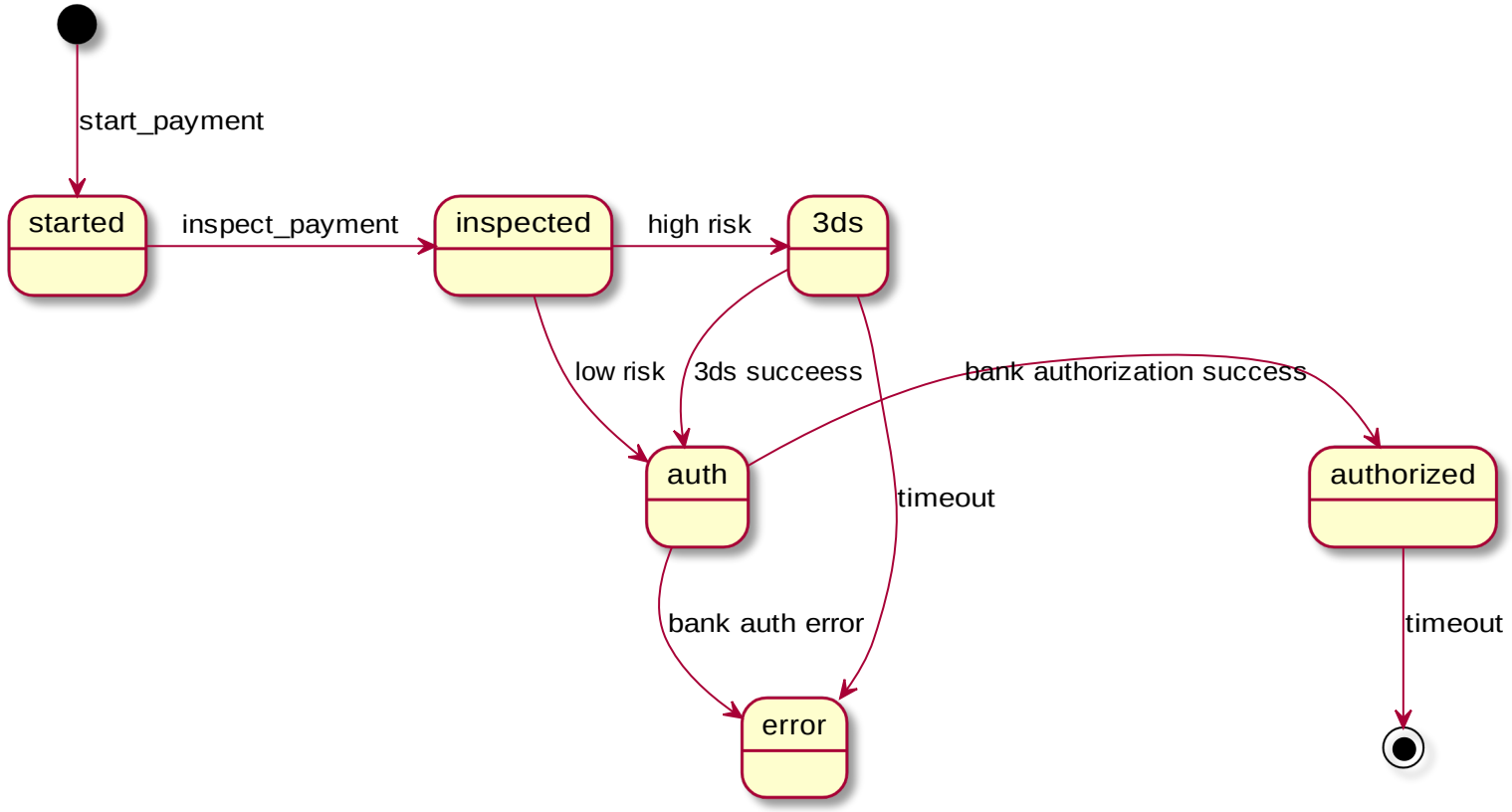
- консистентность
- горизонтальная масштабируемость

Какие требования стояли перед процессингом

Принимать платежи:

- консистентность
- горизонтальная масштабируемость
- отказоустойчивость

Платёж — это конечный автомат



Что решили сделать

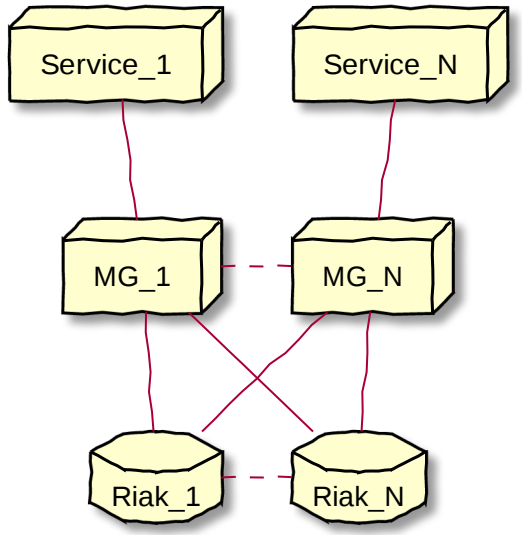
Сервис для обработки конечных автоматов — **machinegun**

Идея в том, чтобы собрать в одном месте основную сложность

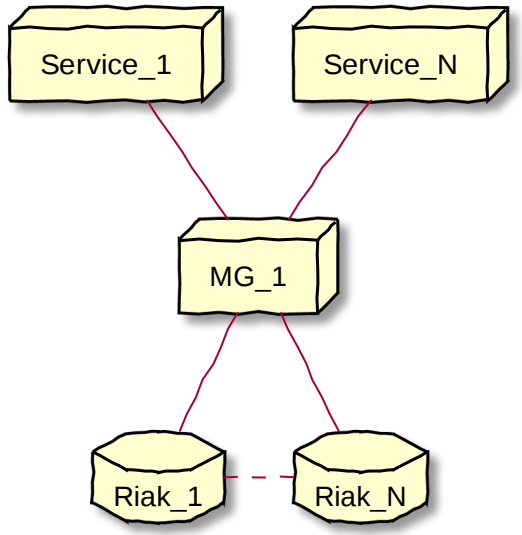
Какие у него задачи:

- хранение стейта
- взаимодействие со stateless процессорами
- распределённая отказоустойчивая обработка
- обработка таймеров

Общая архитектура



Общая архитектура (что уже сделали)

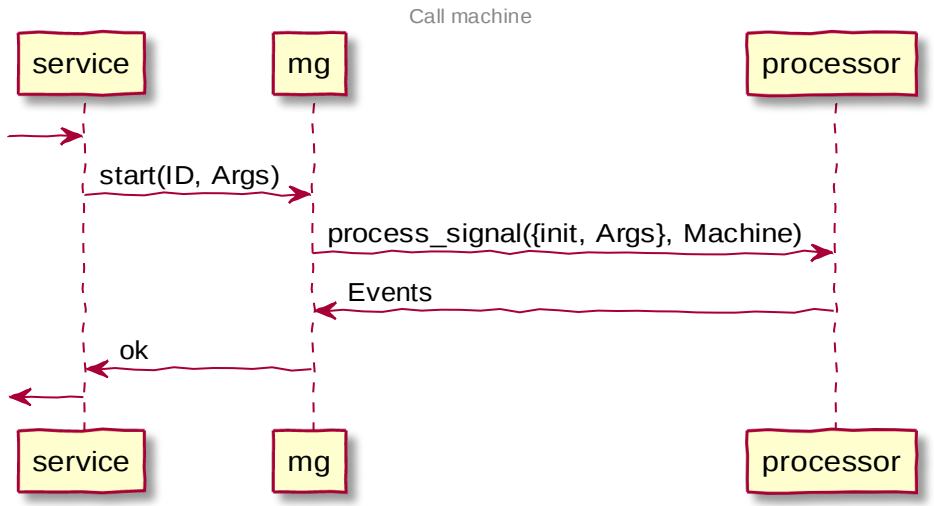


Что такое автомат (machine) в представление MG

```
-type machine() ::
  {
    History  :: [event()],
    AuxState :: term(),
    Timer    :: timestamp()
  }
```

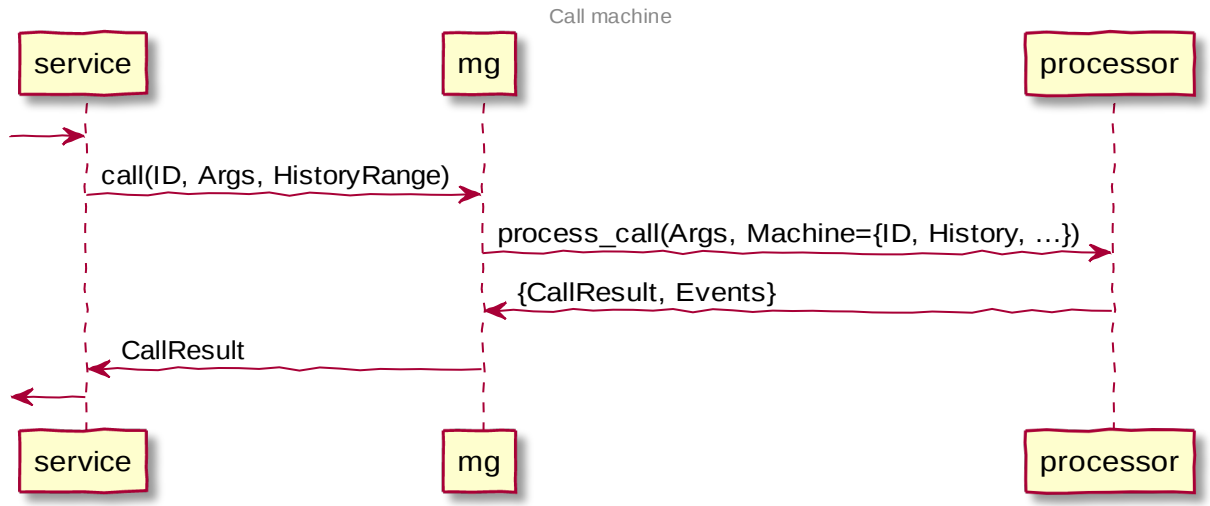
Взаимодействие с процессором

Запуск машины



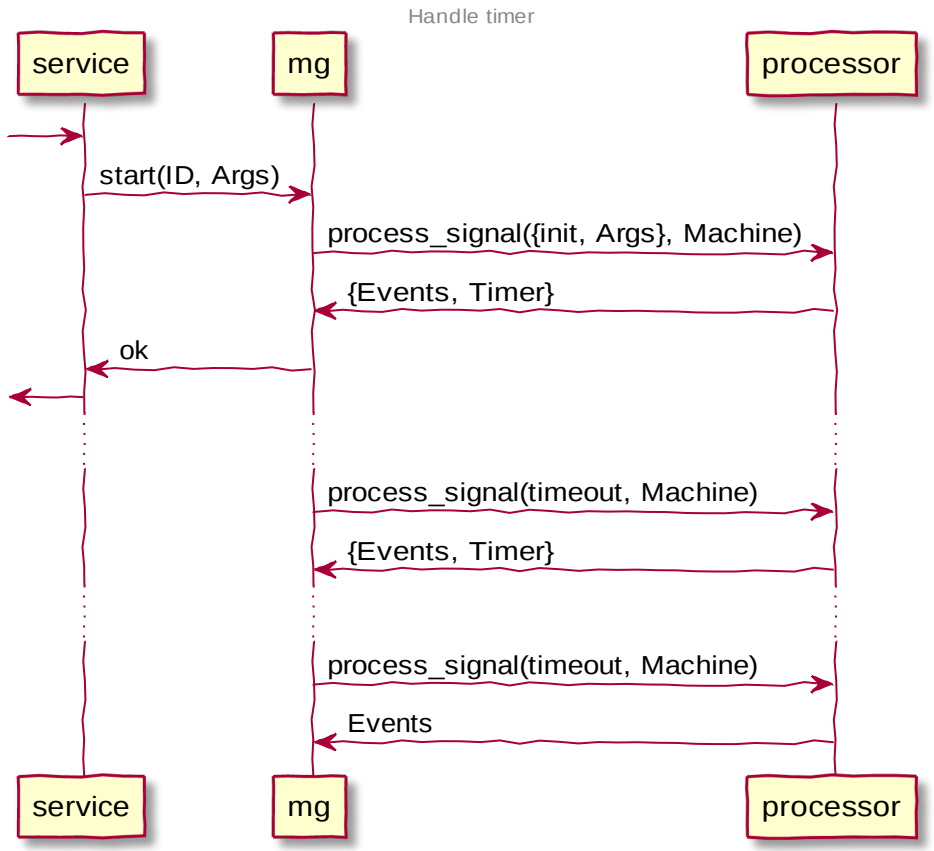
Взаимодействие с процессором

Взаимодействие при обработке запроса



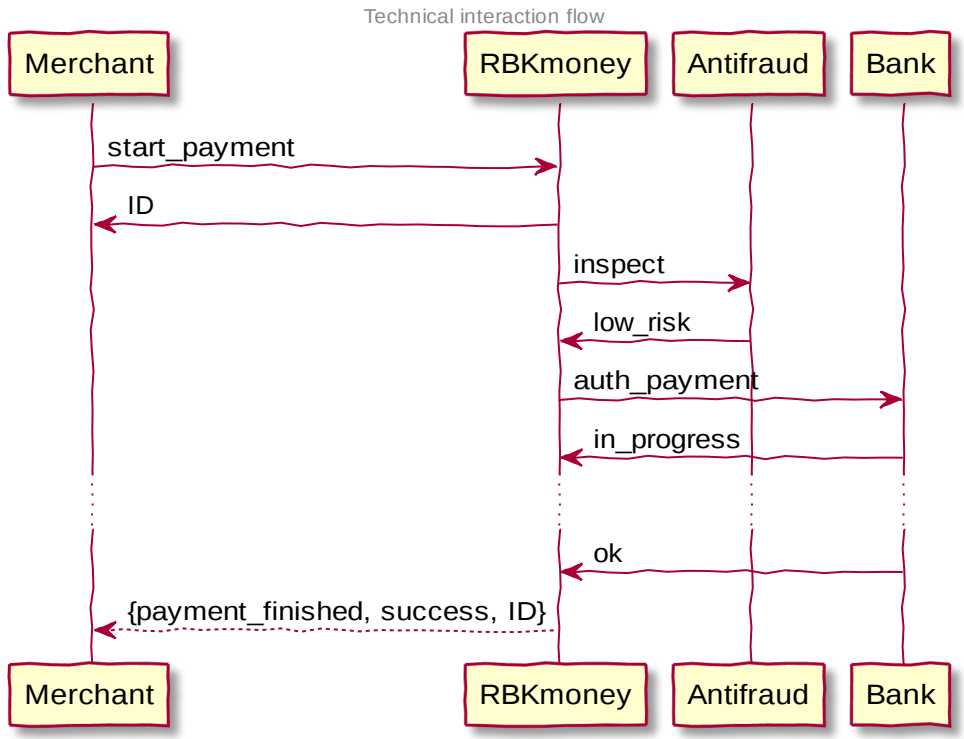
Взаимодействие с процессором

Взаимодействие при обработке таймера



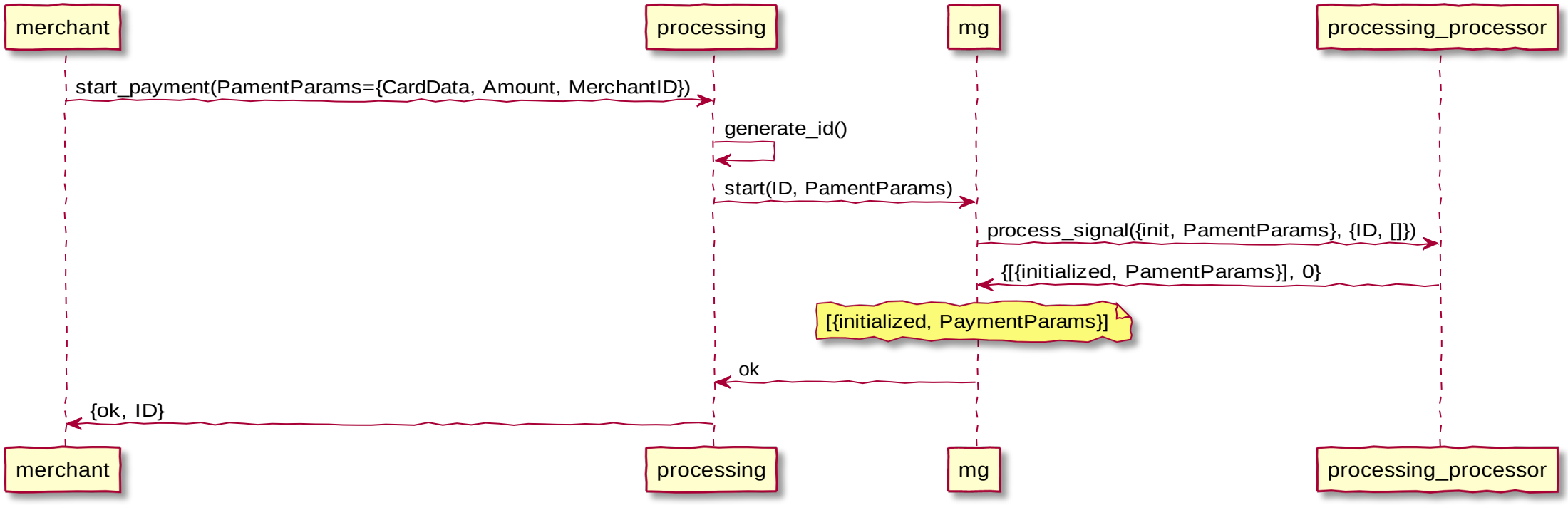
Пример: проведение платежа

Flow платежа



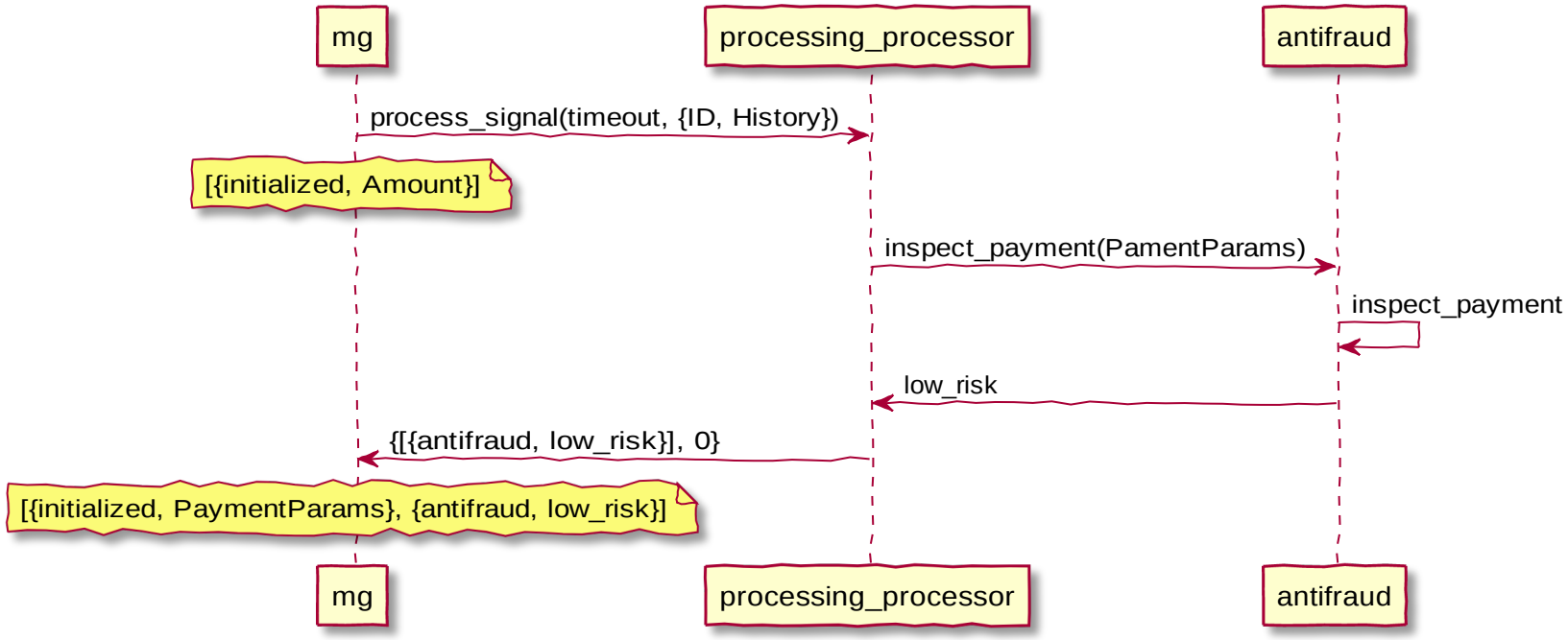
Пример: проведение платежа

Создание платежа



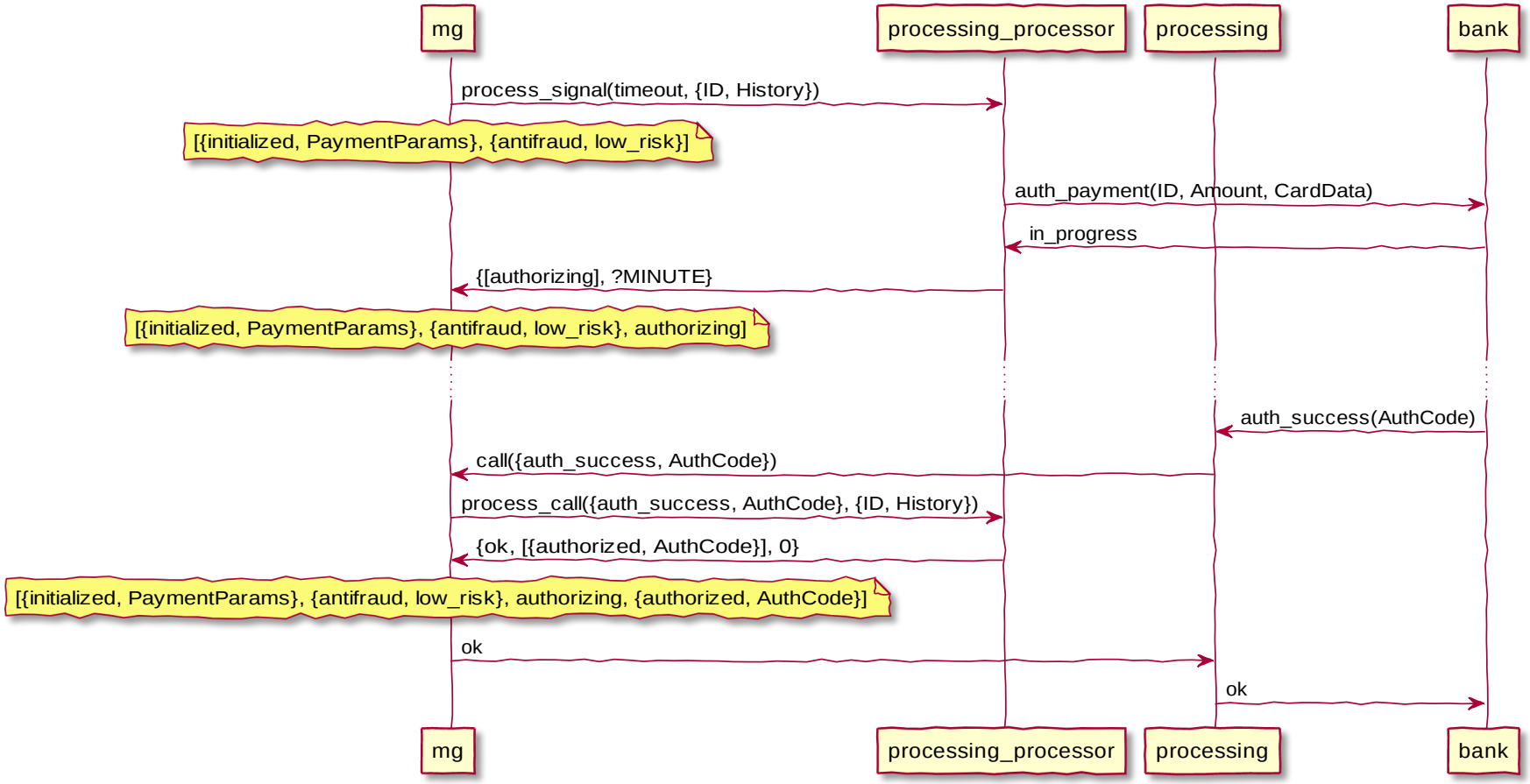
Пример: проведение платежа

Инспекция внешним антифродом



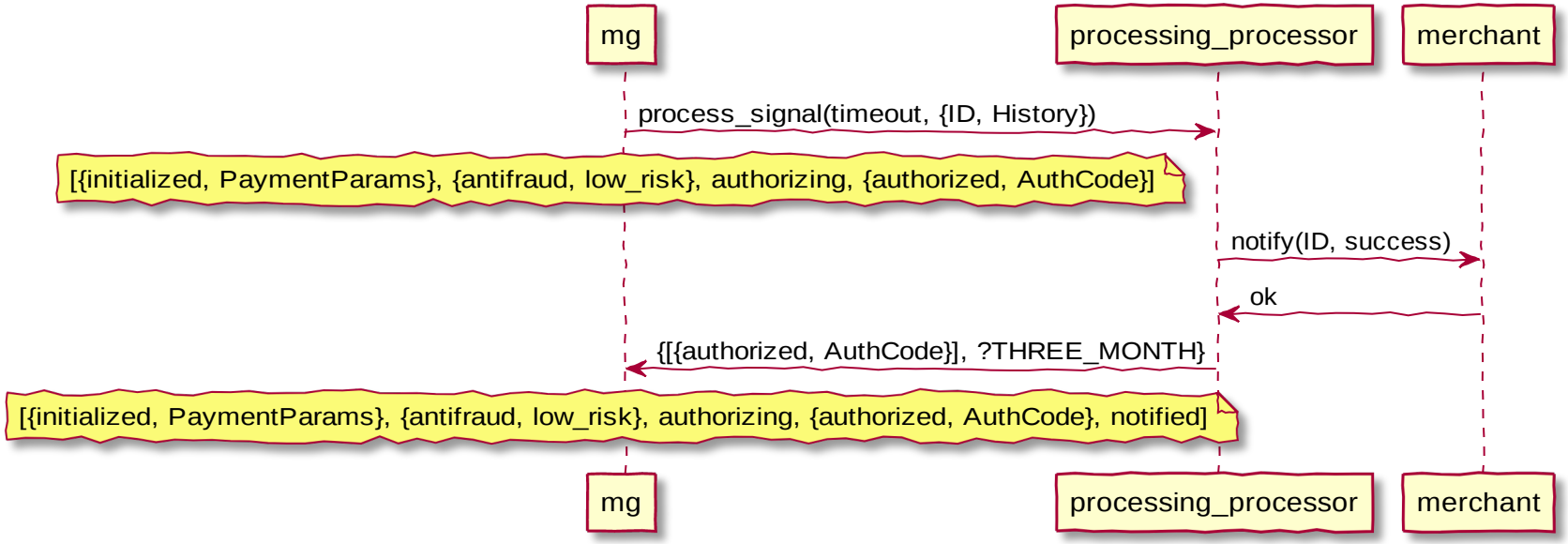
Пример: проведение платежа

Авторизация в банке



Пример: проведение платежа

Уведомление мёрчанта



Кратко о том, как оно работает

Представление машины внутри эрланга

- Машина — gen_server
- Машины лениво загружаются и выгружаются
- Любой запрос к машине gen_server:call

Кратко о том, как оно работает

Представление в базе:

2 таблички:

- machines: {id, state, timer, events_range}
- events: {{machine_id, id}, ts, payload}

Кратко о том, как оно работает

Таймеры

- ts таймера и status — это вторичные ключи в риаке в таблице machines
- процесс поллер раз в секунду делает выборку по вторичному ключу N готовых таймеров

Что в результате

Что в результате

- это работает :)

Что в результате

- это работает :)
- не дописали :((распределённую версия готова на 50%)

Что в результате

- это работает :)
- не дописали :((распределённую версия готова на 50%)
- удобно использовать только в том случае, когда автоматов много, и они маленькие

Что в результате

- это работает :)
- не дописали :((распределённую версия готова на 50%)
- удобно использовать только в том случае, когда автоматов много, и они маленькие
- писать логику довольно сложно, но можно забыть про множество проблем (обработка ошибок, работа с базой, реализация таймеров и тд)

Что в результате

- это работает :)
- не дописали :((распределённую версия готова на 50%)
- удобно использовать только в том случае, когда автоматов много, и они маленькие
- писать логику довольно сложно, но можно забыть про множество проблем (обработка ошибок, работа с базой, реализация таймеров и тд)
- к риаку никаких вопросов (~100kk записей)

Что в результате

- это работает :)
- не дописали :((распределённую версия готова на 50%)
- удобно использовать только в том случае, когда автоматов много, и они маленькие
- писать логику довольно сложно, но можно забыть про множество проблем (обработка ошибок, работа с базой, реализация таймеров и тд)
- к риаку никаких вопросов (~100kk записей)
- основные проблемы были от таймеров и обработки ошибок от процессора

Что в результате

- это работает :)
- не дописали :((распределённую версия готова на 50%)
- удобно использовать только в том случае, когда автоматов много, и они маленькие
- писать логику довольно сложно, но можно забыть про множество проблем (обработка ошибок, работа с базой, реализация таймеров и тд)
- к риаку никаких вопросов (~100kk записей)
- основные проблемы были от таймеров и обработки ошибок от процессора
- проект открыли, но пока его собрать без внутренней инфраструктуры не получится

Вопросы?