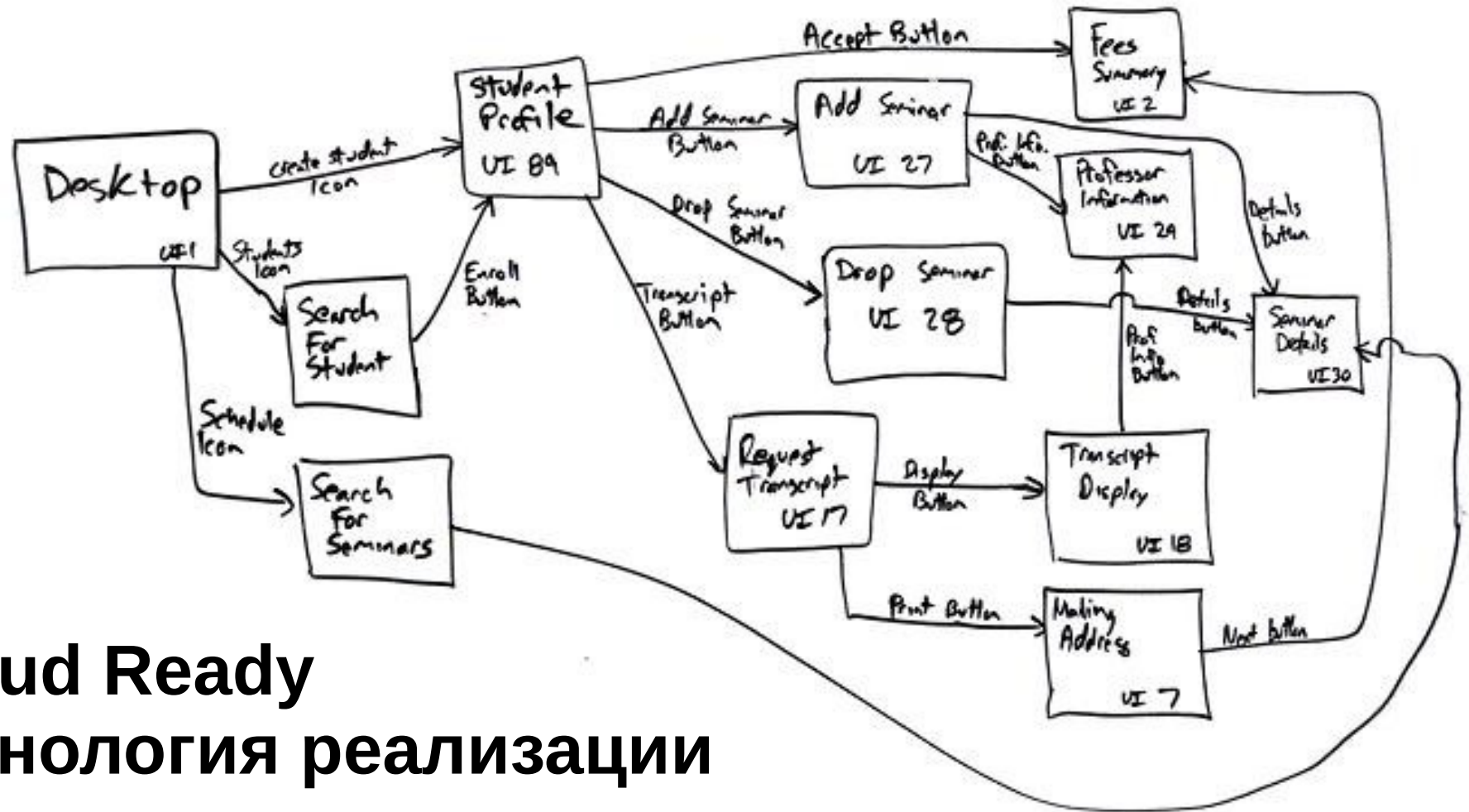


# Stacker



**Cloud Ready**  
технология реализации  
фронтальных сценариев для  
ARM, мобильных и Web приложений

# Stacker для бизнеса



# РЕАЛИЗАЦИЯ САМЫХ СМЕЛЫХ ИДЕЙ

при разумных затратах на разработку и сопровождение

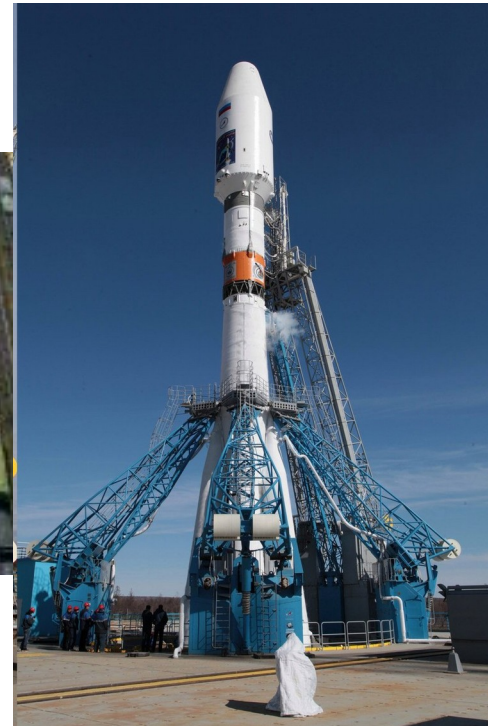
# Stacker для разработчика



Инструмент, позволяющий  
**снизить** сложность и **уровень боли**  
при разработке “кровавого Enterprise”



# Stacker обеспечивает



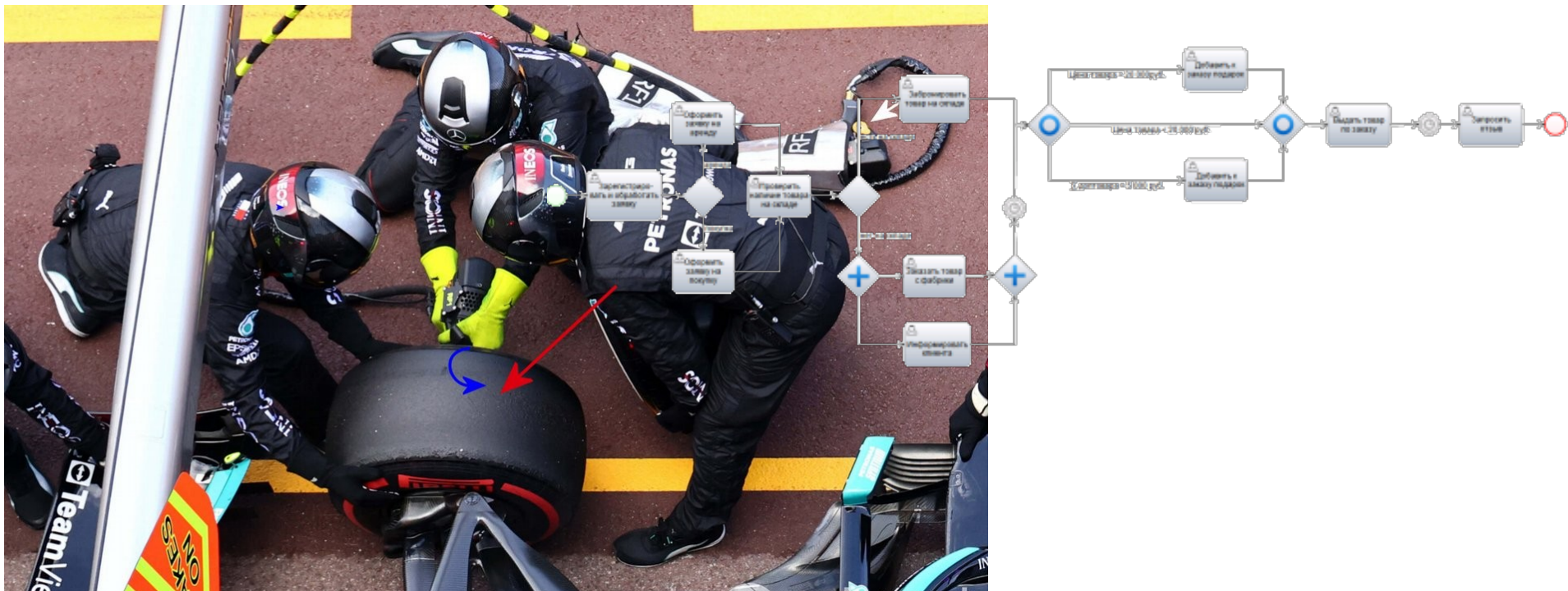
- Построение способных к развитию функциональности горизонтально-масштабируемых систем
- Широкие возможности для переиспользования компонентов
- Омниканальность

# Stacker обеспечивает



- Независимый цикл разработки, тестирования, внедрения и сопровождения компонентов
- Исключение ошибок на ранних этапах разработки
- Низкий порог входа

# Stacker обеспечивает



- Горячую замену компонентов в процессе эксплуатации
- Само-документируемость - система в реальном времени может быть визуализирована как диаграмма BPMN или EPC

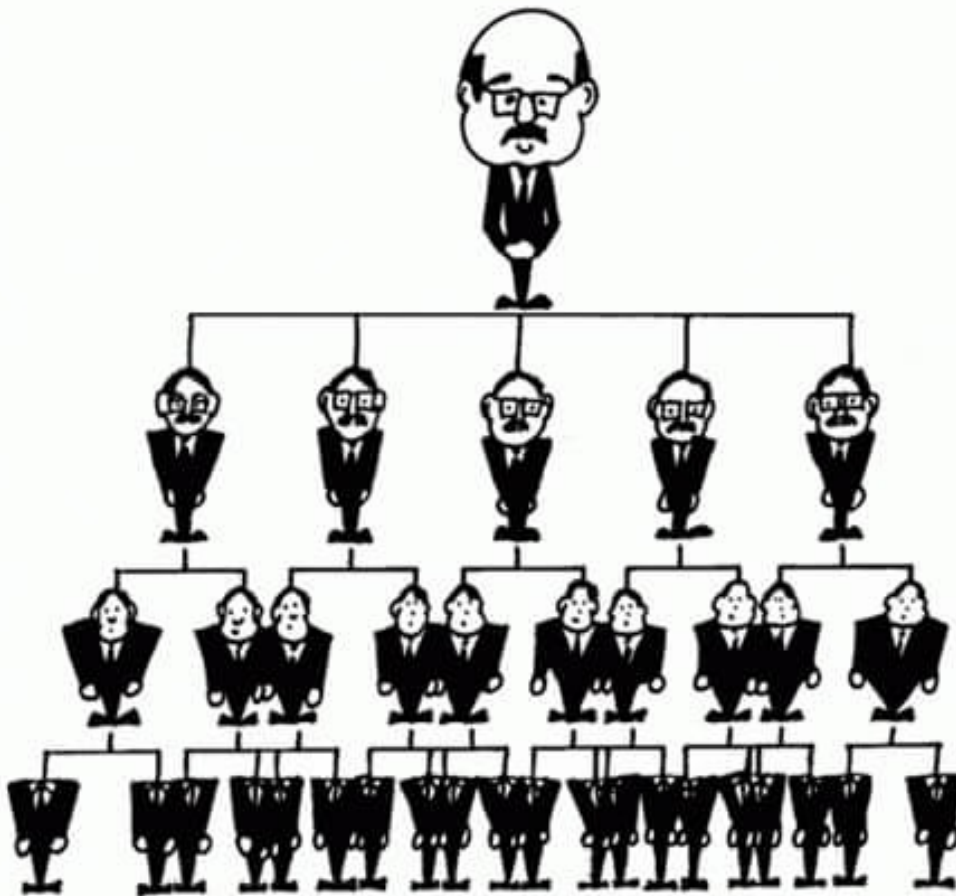
# Stacker – скучные подробности





# Stacker - устройство

- Фронтальный сценарий описывается как **иерархический Workflow**
- Элементы иерархии - компоненты - простые, плоские **Workflow**, реализуемые как **stateless microservices**
- Все взаимодействия в иерархии **явно** описываются **формальными контрактами**



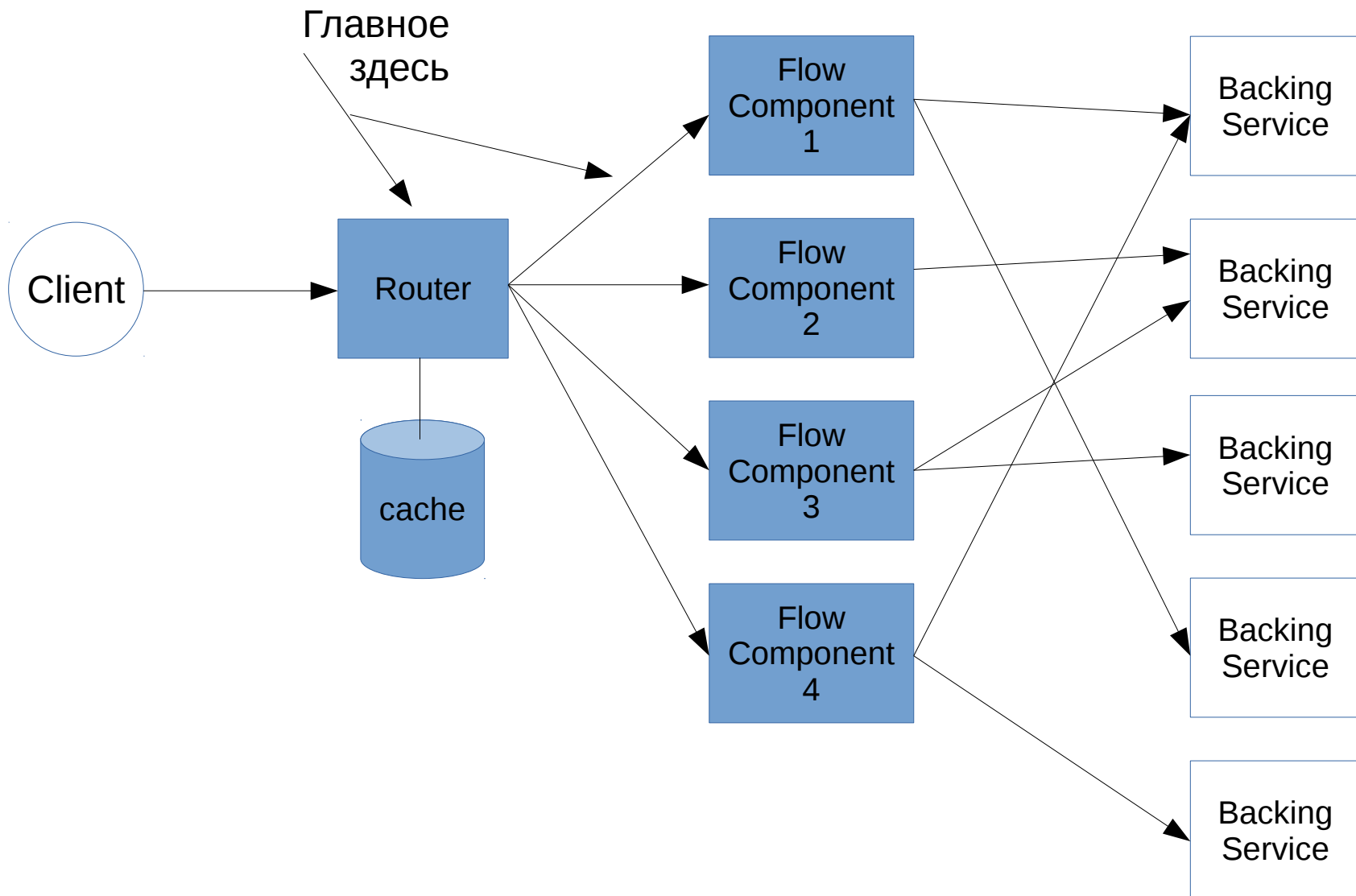


# Stacker - главное



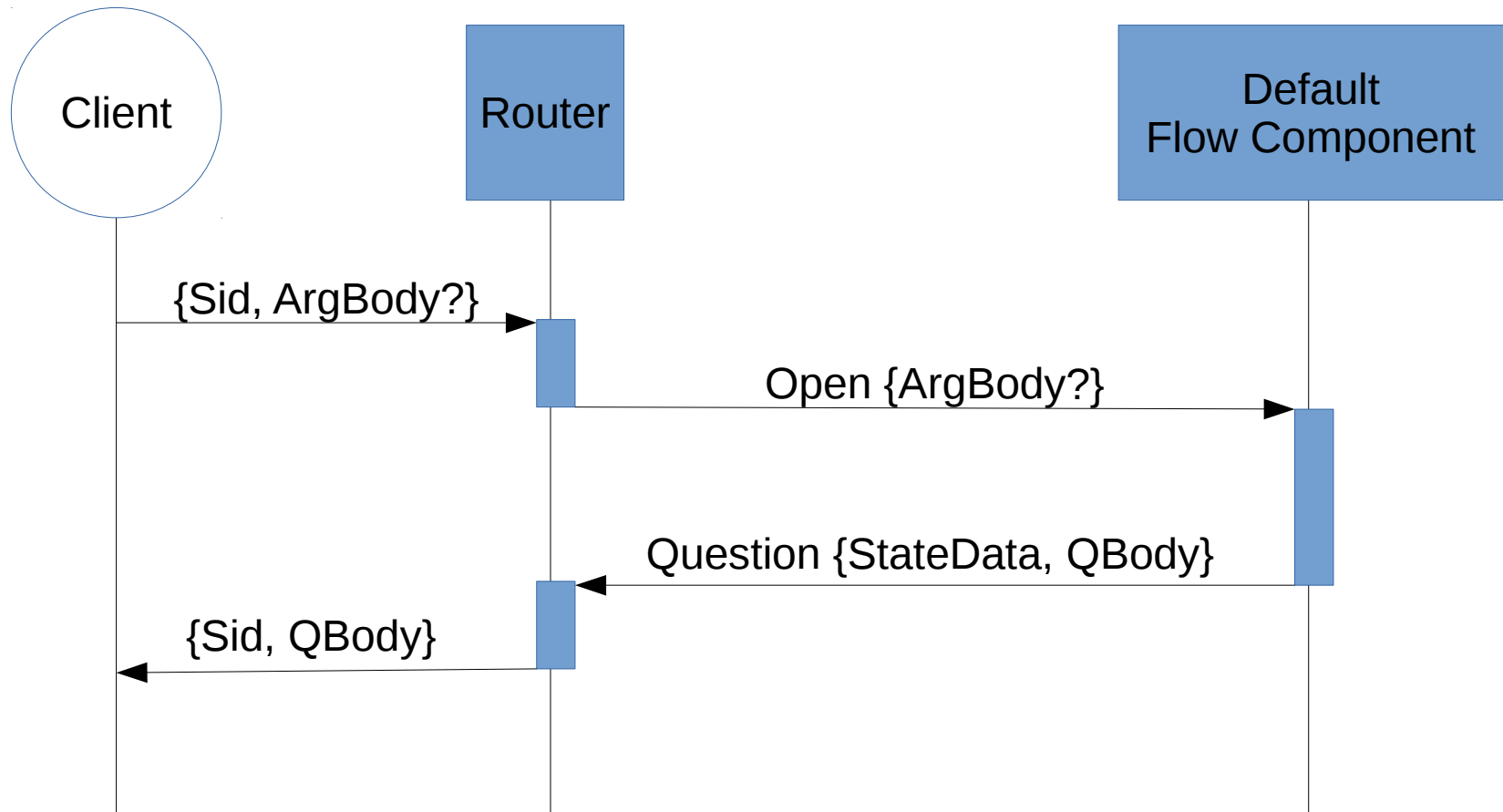
**ФУНКЦИЯ УПРАВЛЕНИЯ**  
вынесена в отдельный **microservice**

# Stacker - архитектура



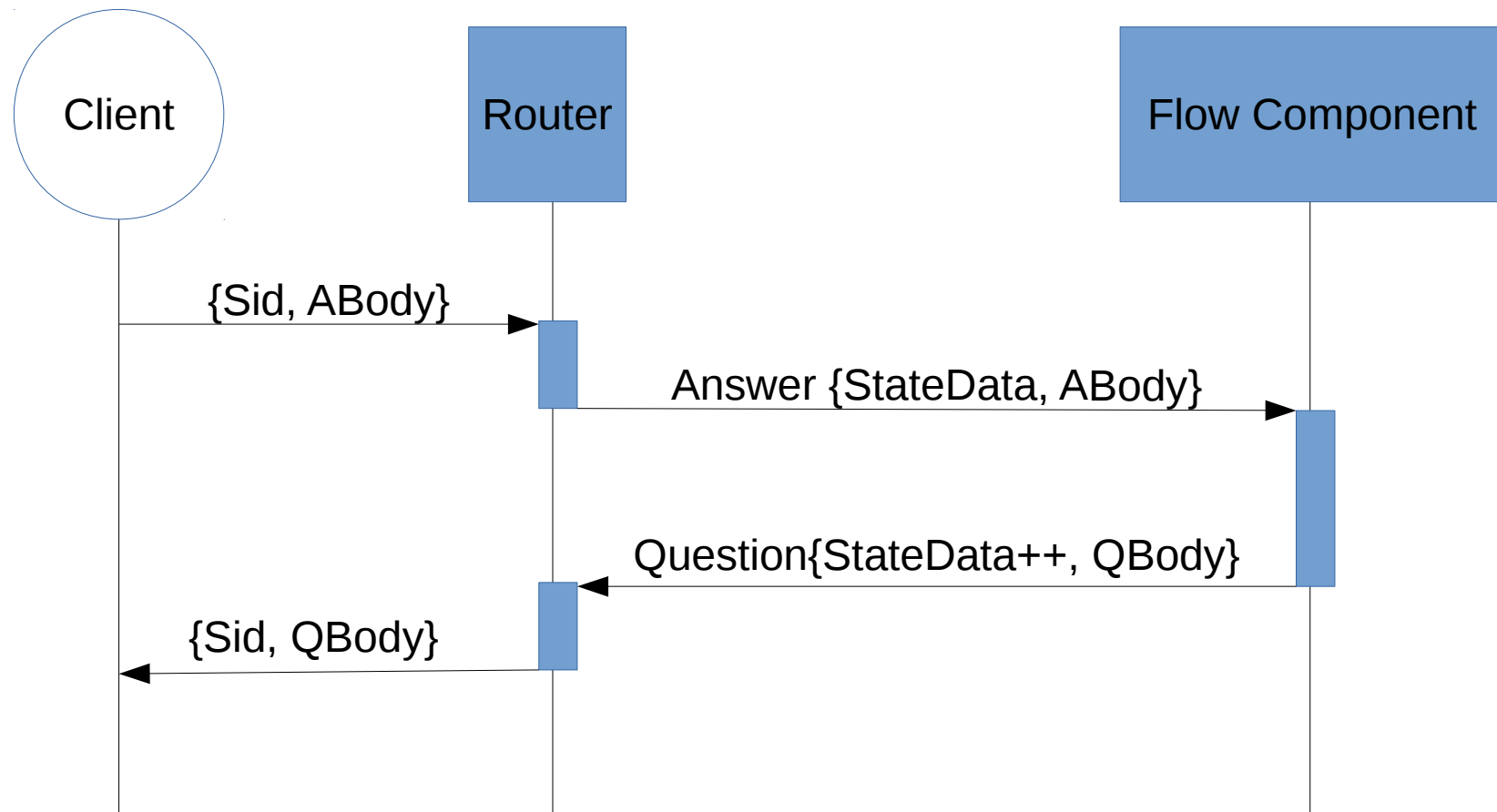
# Stacker – детали

Схема 1, первичный вход, нет данных о состоянии



# Stacker – детали

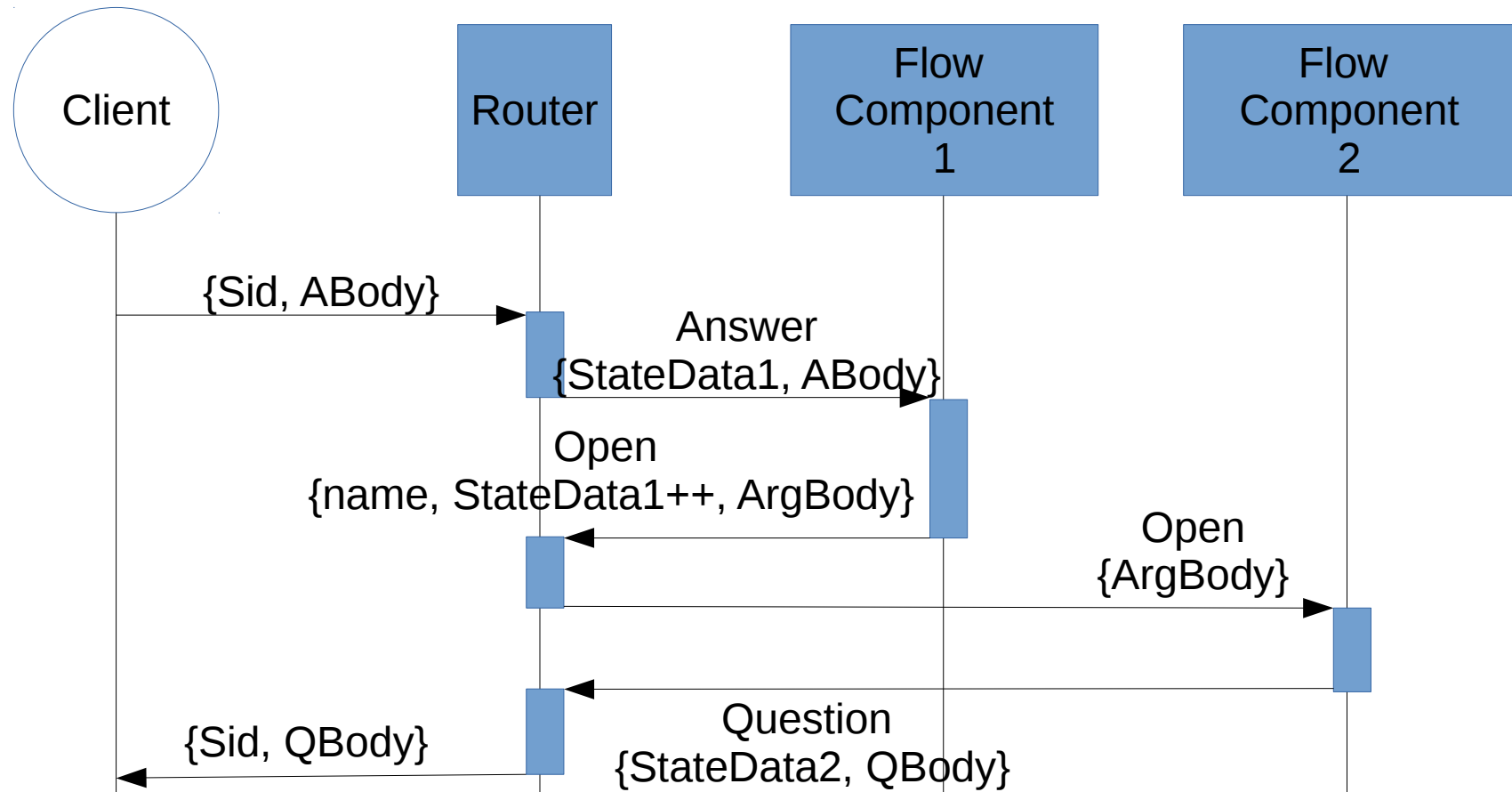
Схема 2, взаимодействие с клиентом





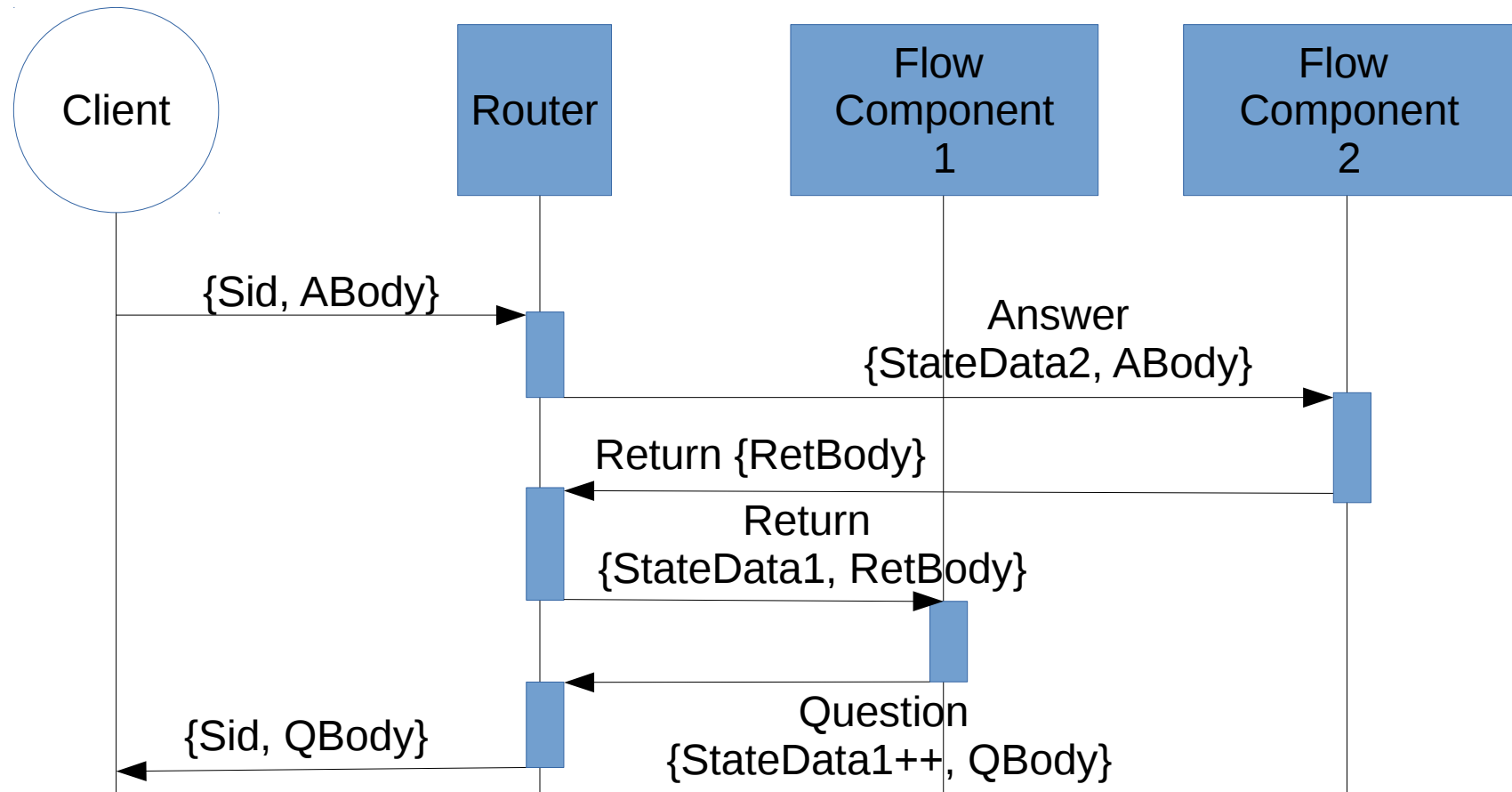
# Stacker – детали

Схема 3, вызов другого компонента



# Stacker – детали

Схема 4, возвращение к вызывавшему компоненту



# Stacker – уже сделано

Первые  
90%

- Разработан контракт взаимодействия между Router и Flow компонентом
- Разработан Router
  - Класс Router, конфигурация
  - Интерфейсы транспорта и хранилища сессий
  - Стэк и элемент стэка
- Разработана референсная реализация сервера Router
- Разработан абстрактный Flow компонент
  - Абстрактный Flow
  - Абстрактный State и несколько его подтипов
  - Класс Contract, - декларация контрактов взаимодействия Flow компонентов между собой и с клиентом
- На базе абстрактного Flow компонента разработан и внедряется в настоящий момент в банке LiS – фреймворк для построения композитных транзакционных сервисов
- Разработана референсная реализация сервера Flow

# Stacker – ещё не сделано Вторые 90%

- Предоставление из контекста дополнительных данных для клиента
- Расширение контракта для обработки ошибок взаимодействия
- Хлебные крошки
- NameSpace для группировки компонентов и ограничения доступа
- Приложение для управления конфигурацией Router
- Визуализация системы в виде схем
- Множество всяких инфраструктурных штук



# Stacker

Спасибо за внимание