

Не грози облаку в Azure, попивая смузи у себя в опенспейсе



Часть 2

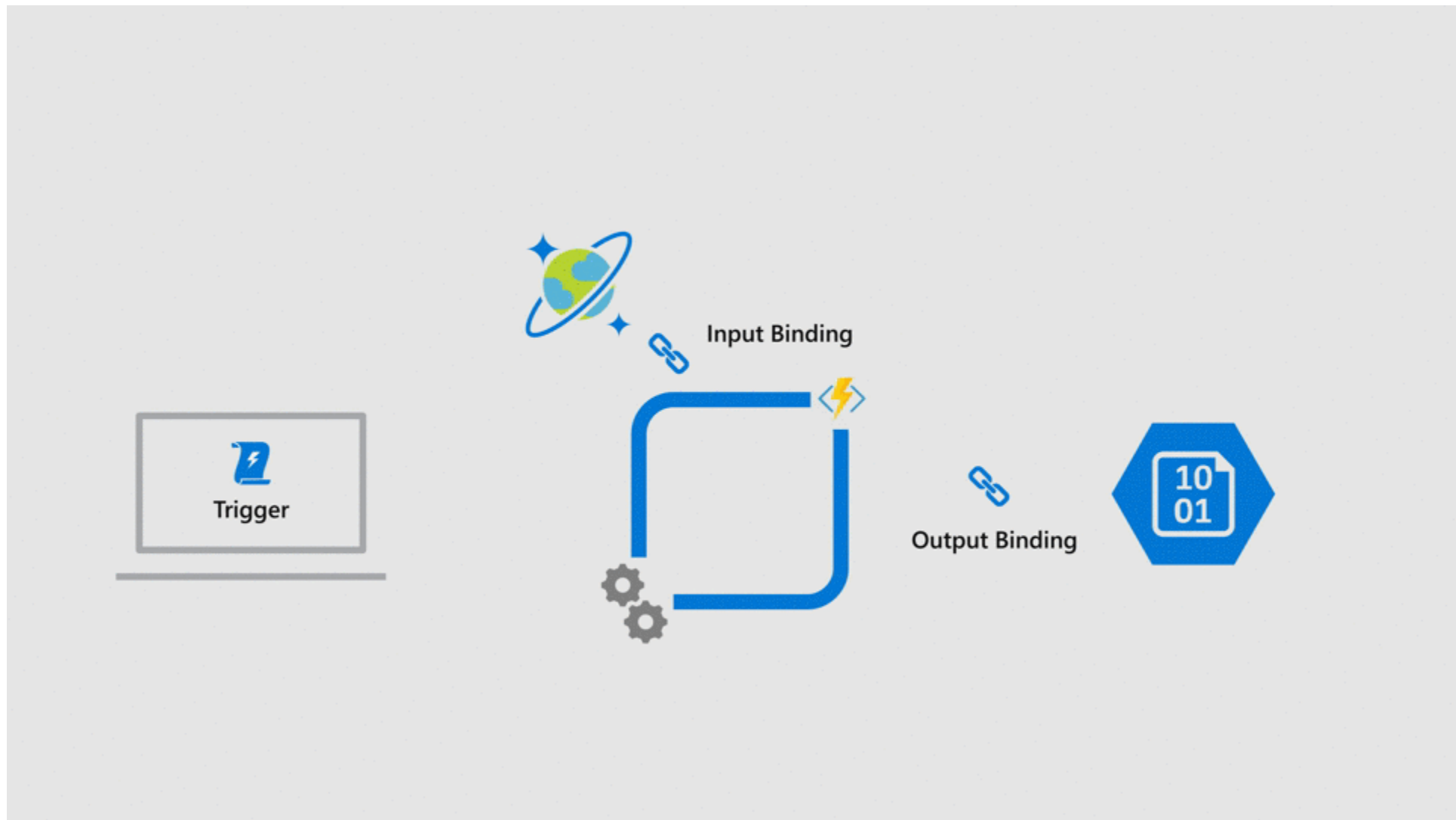


Azure Functions – введение

- Функция как она есть – входные параметры, тело функции, возвращаемые значения
- Кто вызывает? – рантайм Azure Functions
- Когда вызываются? – по триггерам (событиям):
 - HTTP запросы
 - Добавление/изменение данных в БД, файлов
 - Сообщение в очереди сообщений
 - Запуск по расписанию (Cron)
- Объединяются в Function Apps, конфигурируются через переменные окружения



Azure Functions – введение





Azure Functions – работа с HTTP

- Любой роут и метод
- Вход и выход функции – привычные HTTP request и response объекты
- Поддержка пользовательских доменов, HTTPS и HTTP/2 из коробки
- Аутентификация – Azure AD, Microsoft account, сторонние провайдеры (Google, Facebook, Twitter и т.п.)
- Ограничение доступа – доступ по ключу, firewall
- HTTP прокси из коробки



Azure Functions – поддержка языков



- 1.x
 - C# и F# (.NET Framework 4.7)
 - JavaScript (Node 6)
 - Экспериментально – Python, TypeScript, PHP, Batch, Bash, PowerShell
- 2.x
 - C# и F# (.NET Core 2)
 - JavaScript и TypeScript (Node 10)
 - Java (Java 8)
 - Python (Python 3.6)



Azure Functions – модель оплаты и масштабирование

- Стоимость – кол-во выполнений, время выполнения и потреблённая память
- App Service – предсказуемая стоимость, ручное и полуавтоматическое масштабирование, always on, неограниченное время выполнения
- Consumption – стоимость зависит от нагрузки, полностью автоматическое масштабирование, 1.5 GB памяти и 1 CPU на инстанс, максимальное время выполнения – 10 минут
- Premium – положительные стороны первых двух, возможность выбора более мощного «железа»



Azure Functions – особенности разработки

- Каждая функция – отдельный проект и репозиторий (или подмодуль)
- Тестирование – модульные тесты без запуска рантайма, интеграционные тесты с локальным запуском или в dev-аккаунте Azure
- CI/CD – сборка (транспиляция), прогон модульных тестов и деплой в Azure DevOps, возможность деплоить напрямую из репозитория
- Автоматический перезапуск функции при деплое, возможность использования слотов



Azure Functions – мониторинг, логи и аналитика

- Интеграция с Application Insights из коробки
- Метрики производительности – время выполнения, потребление ресурсов, запросы к сторонним зависимостям
- Мониторинг ошибок – падение функции целиком, обработанные исключения, неудавшиеся запросы к сторонним зависимостям
- Логирование – уровни, поддержка произвольных данных, агрегация логов по операциям и сессиям
- Availability тесты и гибкая настройка уведомлений
- Пользовательская аналитика



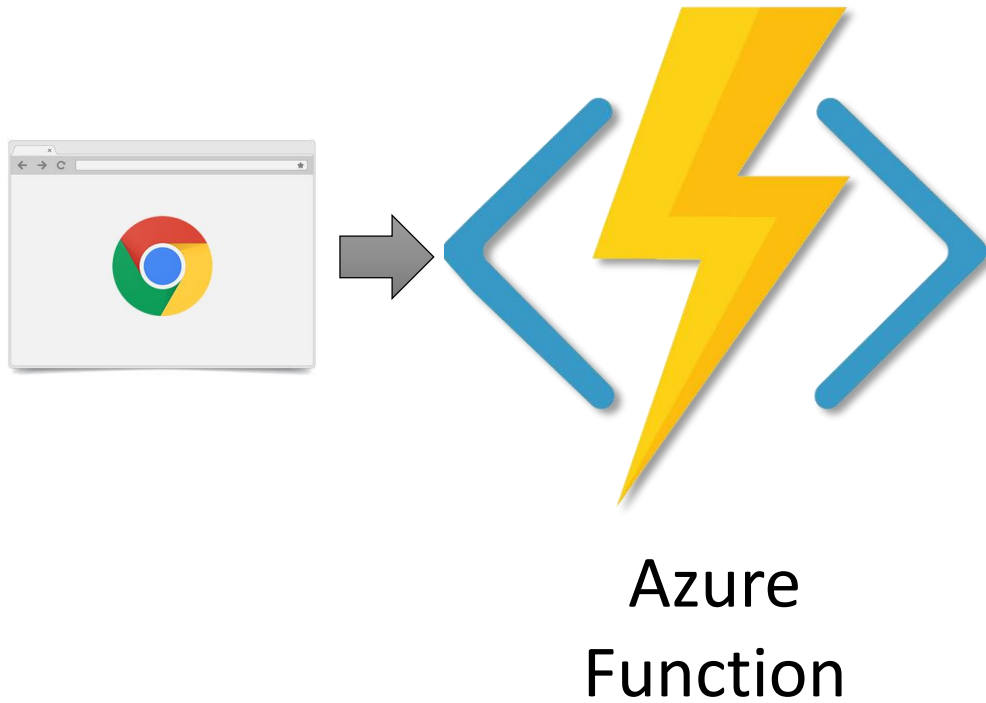
Веб-сервис с поддержкой real-time уведомлений



Client

Веб-сервис с поддержкой real-time уведомлений

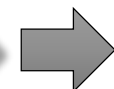
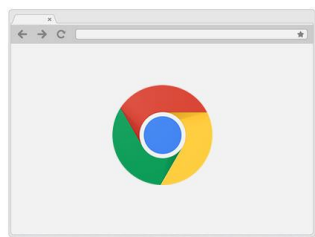
Получаем запрос от
клиента





Веб-сервис с поддержкой real-time уведомлений

Сохраняем контент в
БД



Azure
Cosmos DB



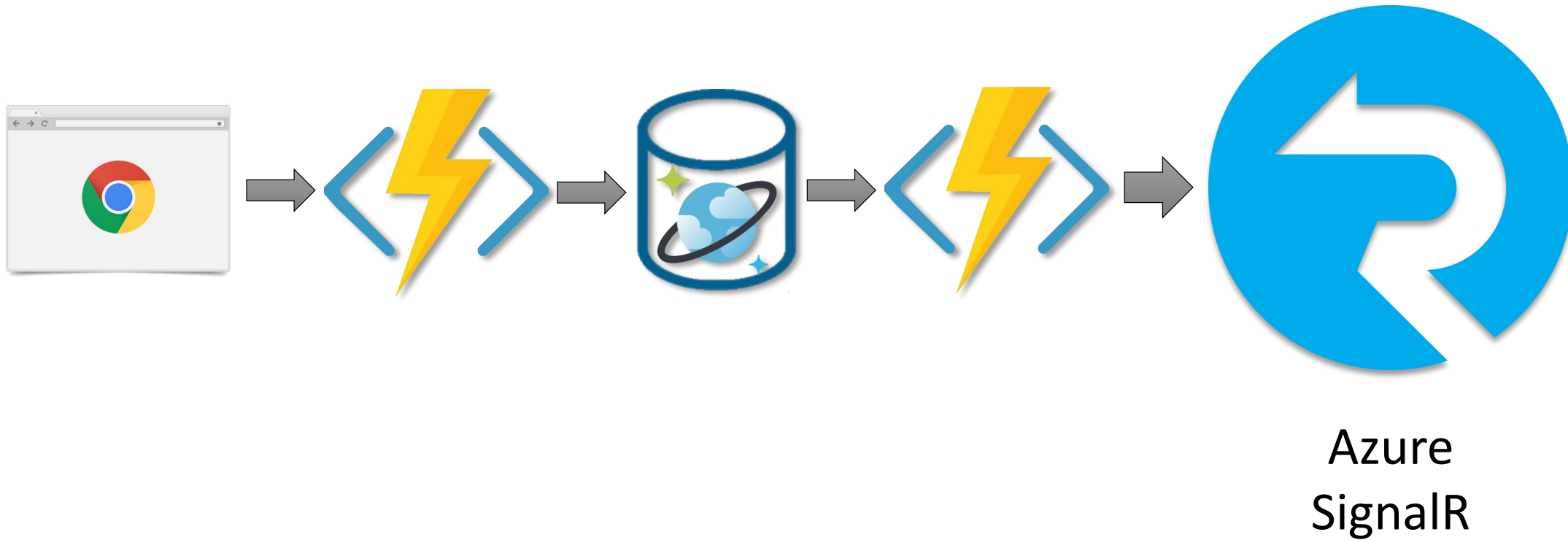
Веб-сервис с поддержкой real-time уведомлений

Реагируем на
появление нового
контента



Веб-сервис с поддержкой real-time уведомлений

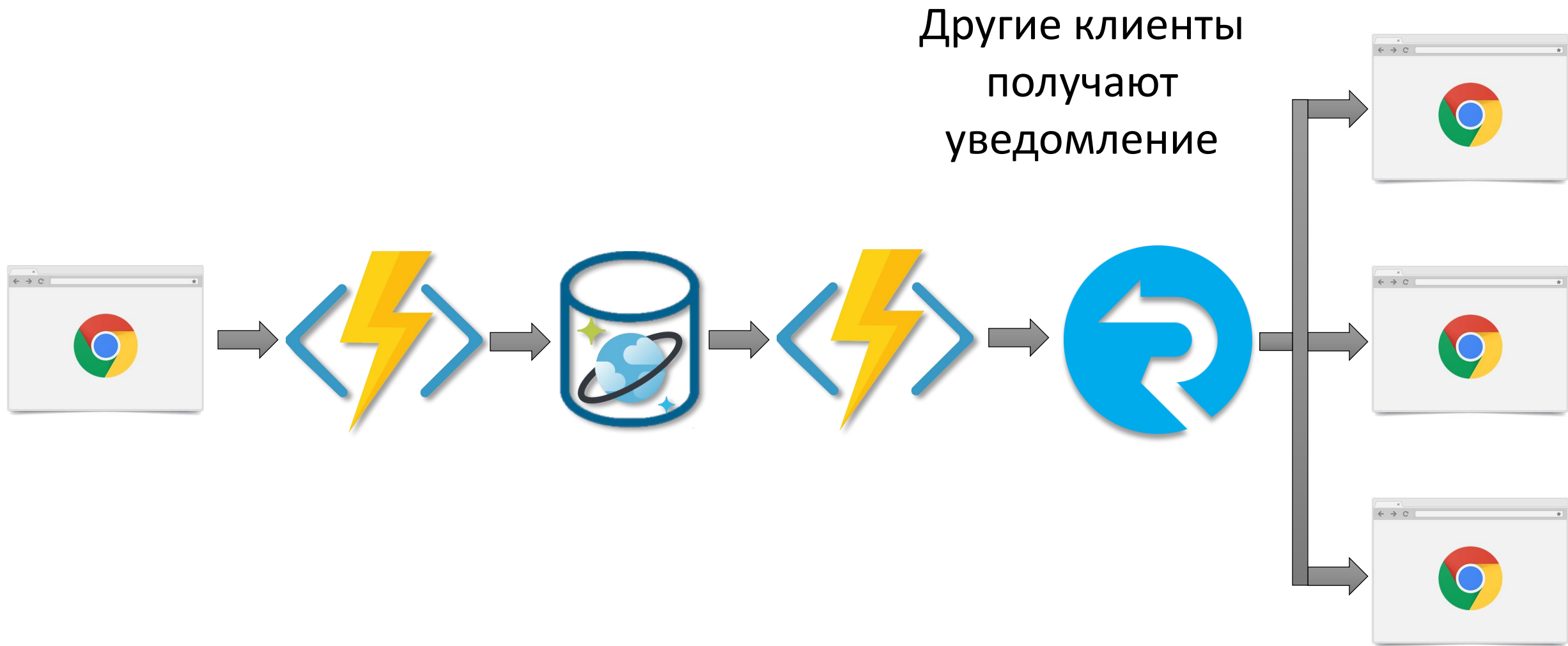
Определяем, кому нужно
отправить уведомление и делаем
рассылку



Azure
SignalR



Веб-сервис с поддержкой real-time уведомлений



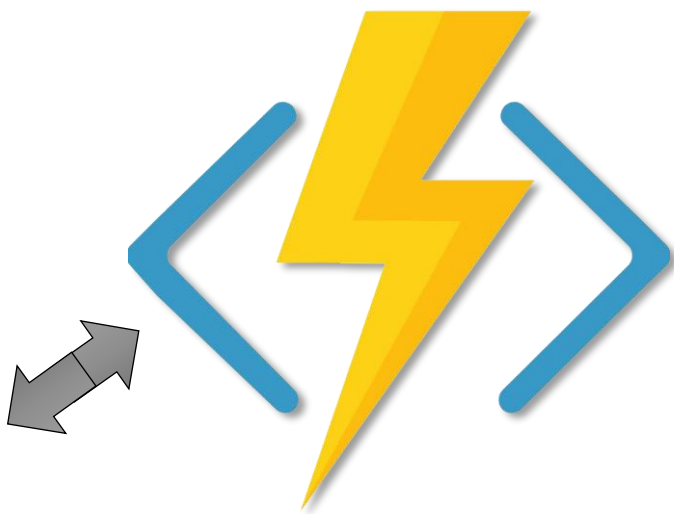


Интеграция с платежной системой





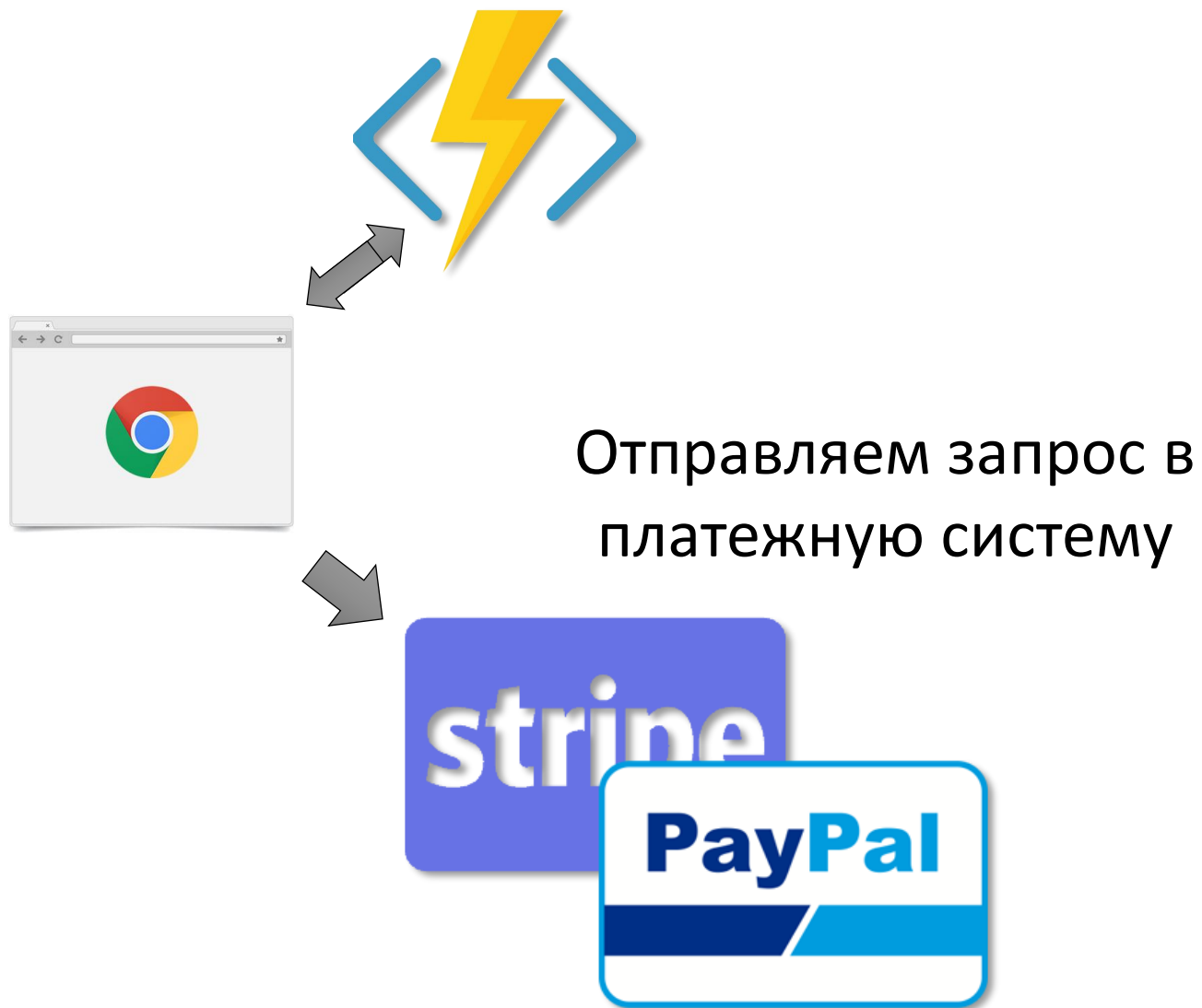
Интеграция с платежной системой



Подготавливаем
данные платежа

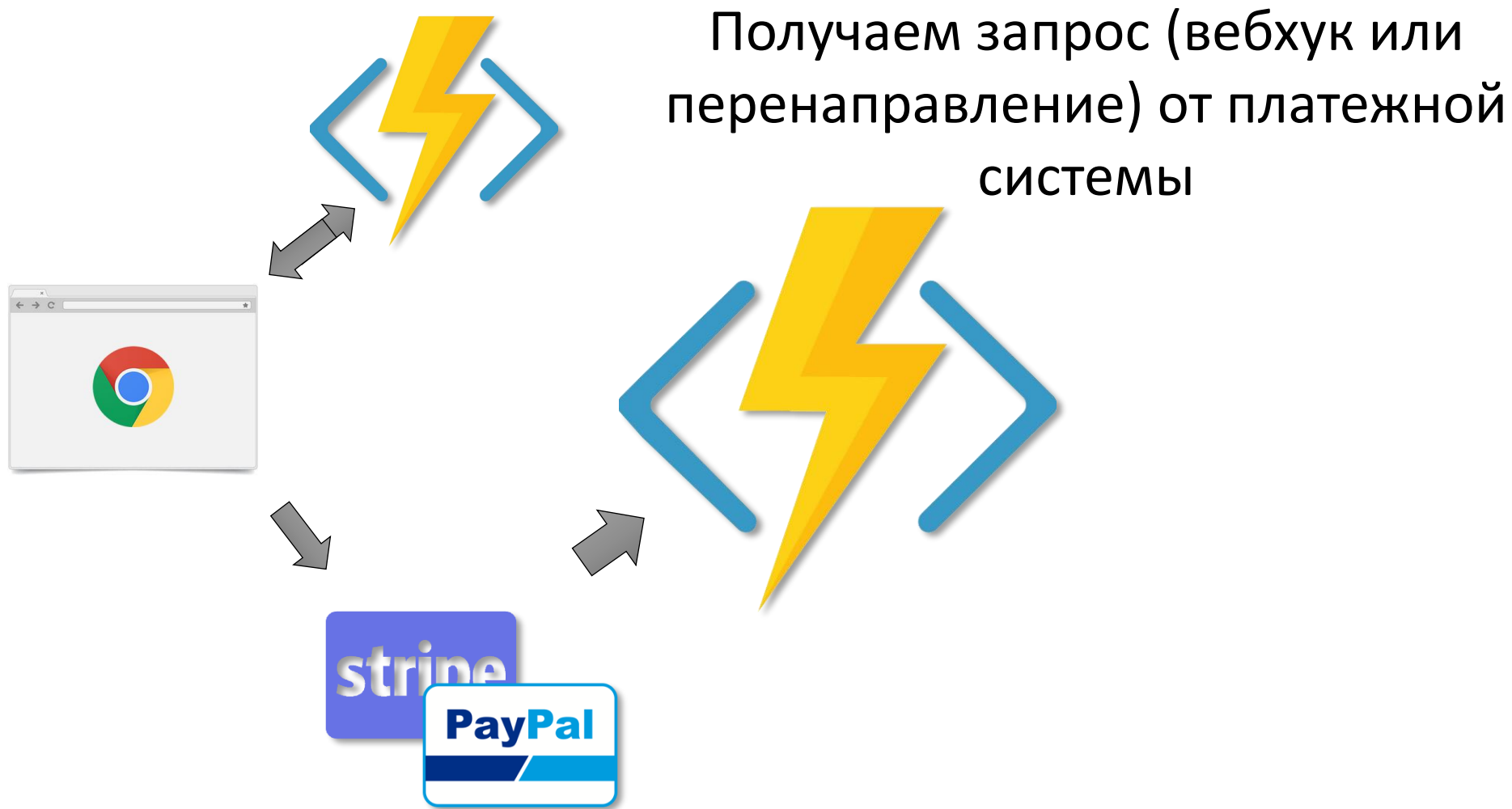


Интеграция с платежной системой



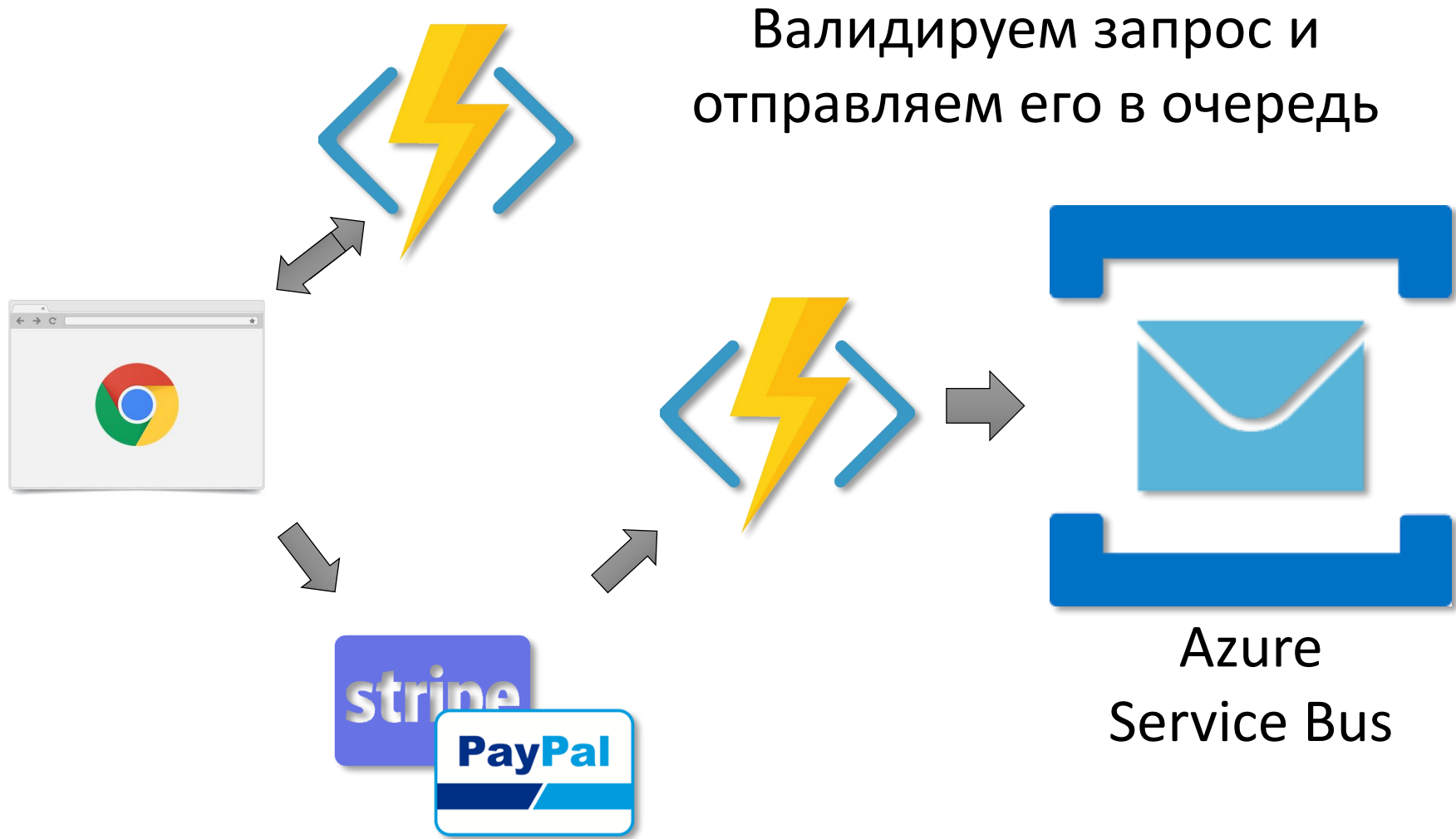


Интеграция с платежной системой



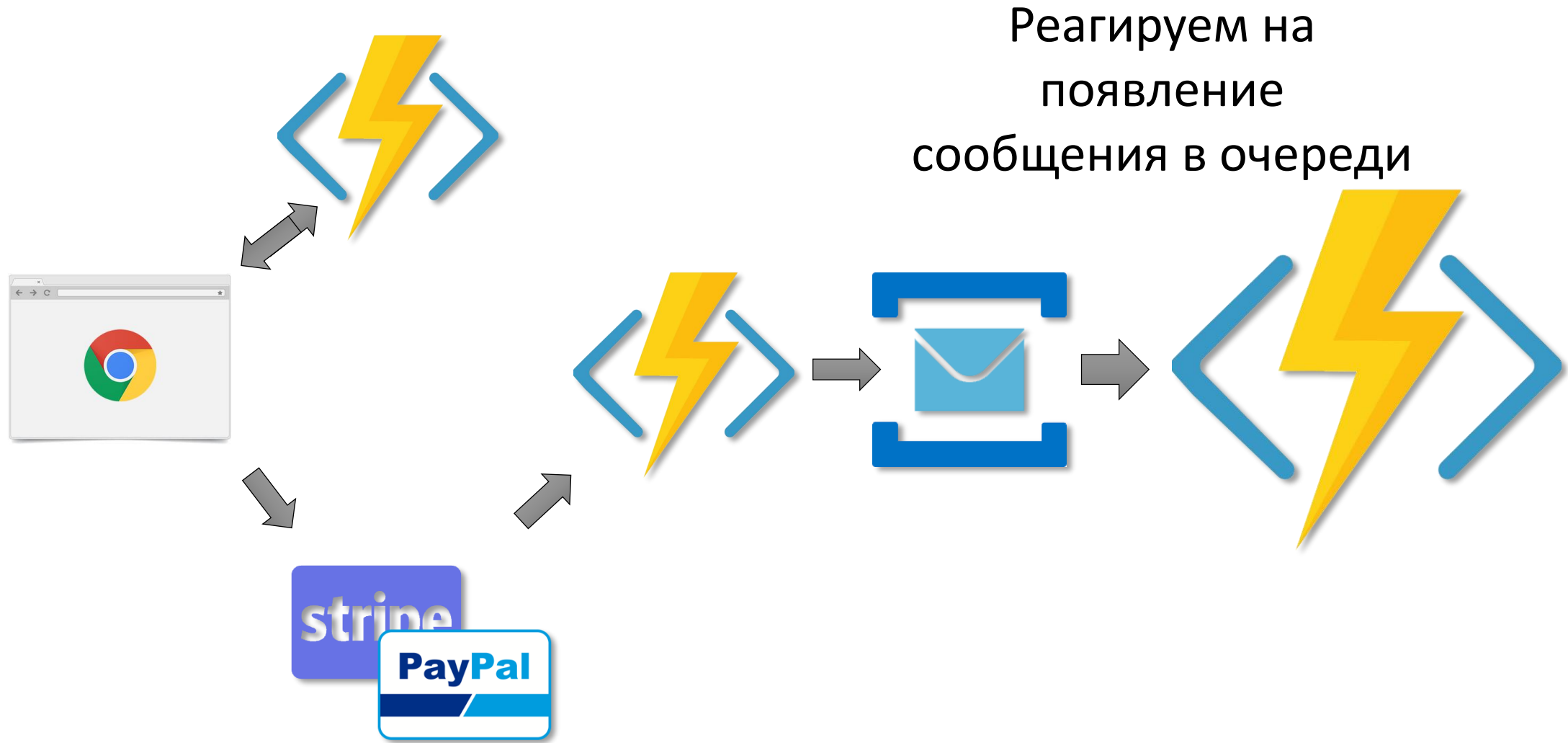


Интеграция с платежной системой



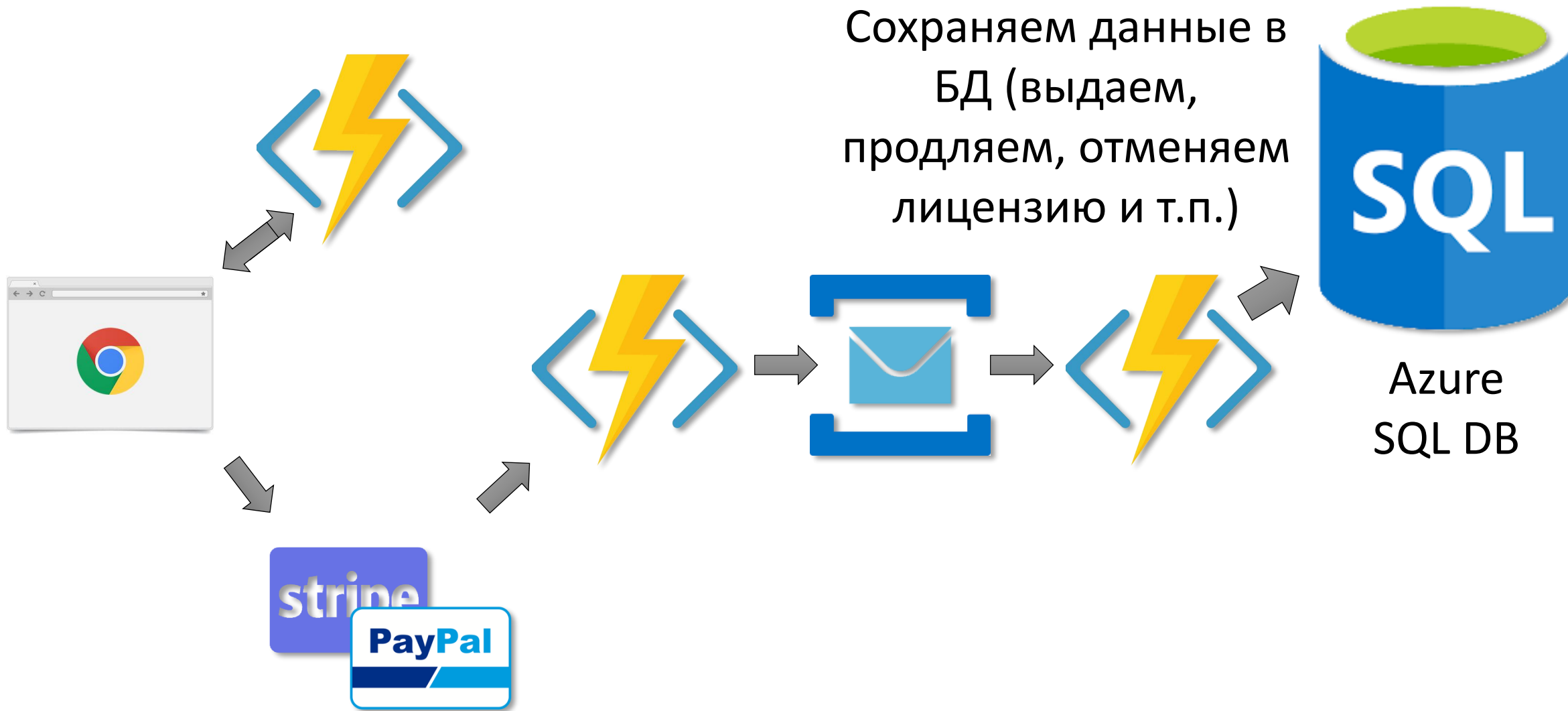


Интеграция с платежной системой



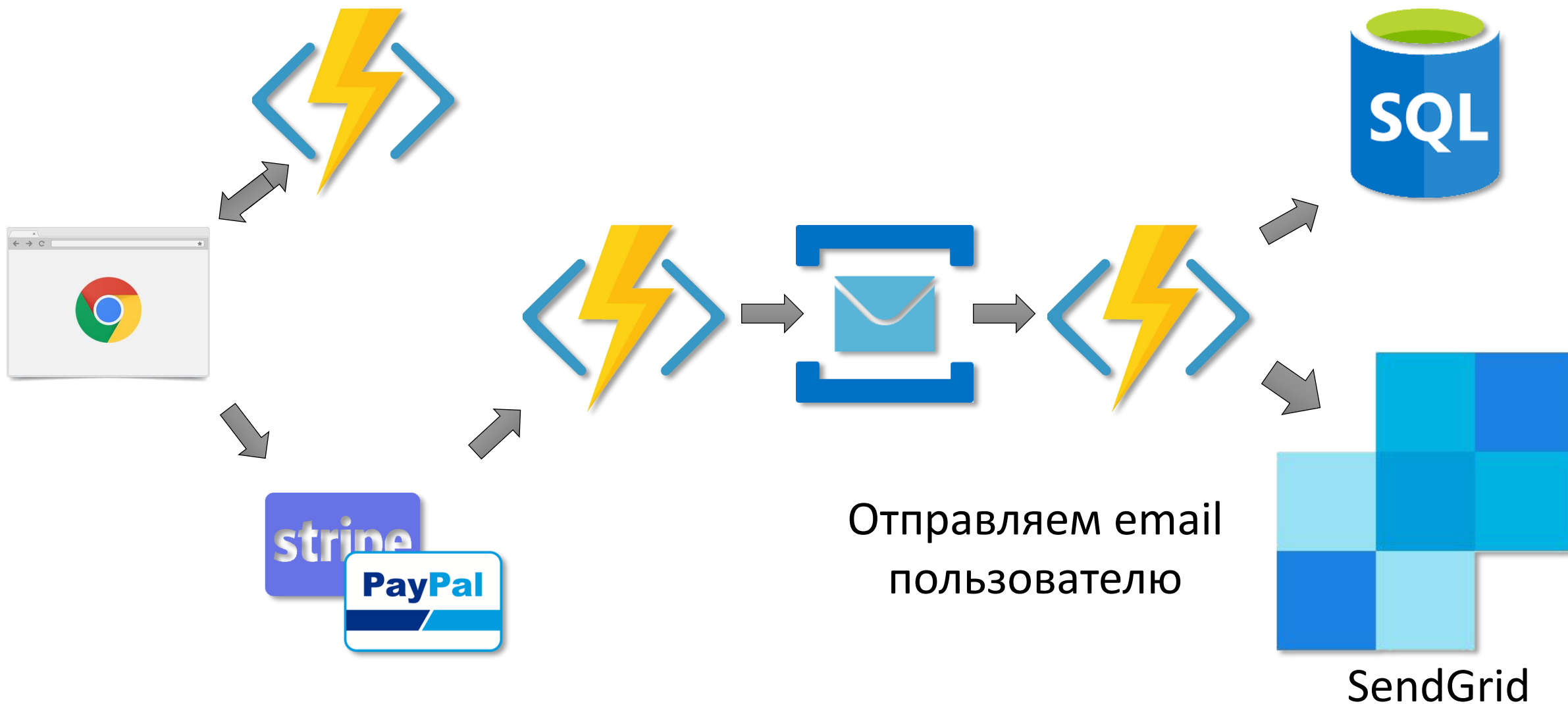


Интеграция с платежной системой



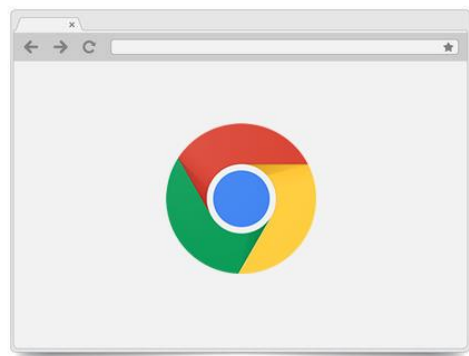


Интеграция с платежной системой



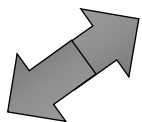
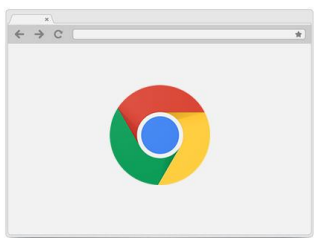


Отправка обратной связи





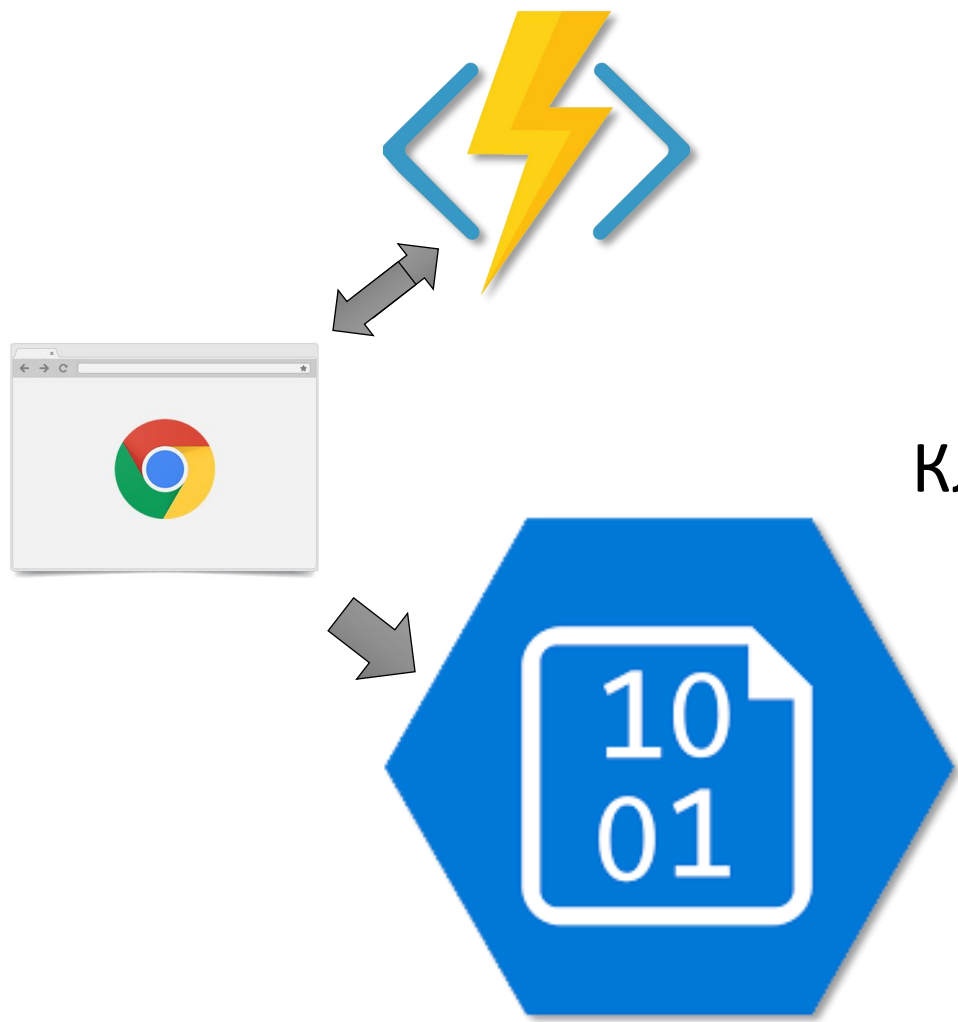
Отправка обратной связи



Подготавливаем
ссылку для загрузки
файла



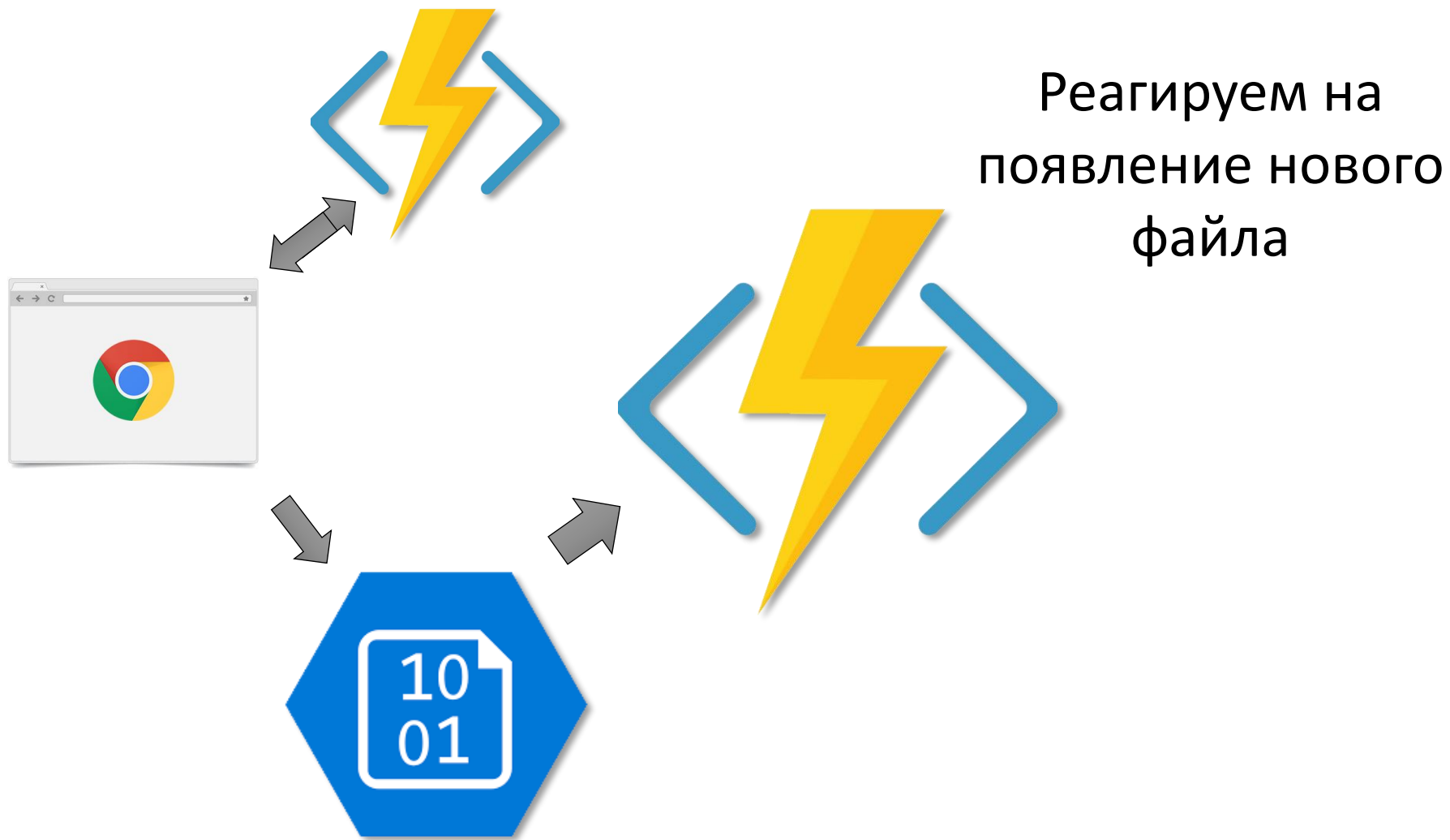
Отправка обратной связи



Клиент загружает
файл

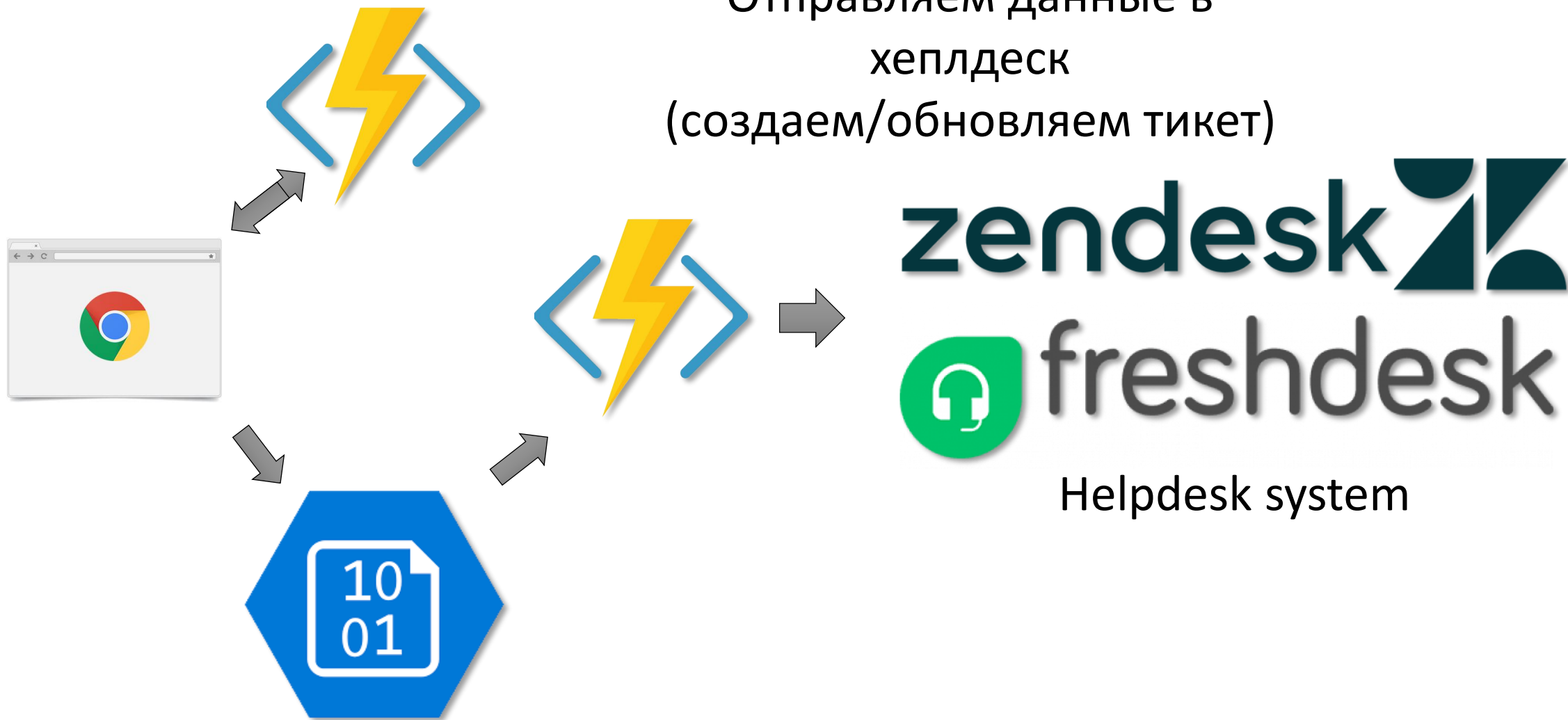


Отправка обратной связи





Отправка обратной связи





Azure Functions – что хорошо, а что плохо

- Плюсы:
 - + Быстрая разработка, простой деплой и поддержка
 - + Подходит как для прототипирования, так и для production ready задач
 - + Масштабирование, мощная работа с HTTP
 - + Возможность развернуть рантайм вне Azure (локально, Azure Stack, Kubernetes)
- Минусы:
 - Холодный старт и ограничение времени работы
 - Пока еще не все типы байндингов в v2.x