Расшифровка встречи.

**Менеджер Миша:** Ребята, привет! У нас с вами будет сегодня насыщенная встреча. В команде есть новые сотрудники, а ещё я расскажу про новую фичу, которую нас просит сделать заказчик. Инна расскажет про текущую архитектуру приложения. Договорились?

**Архитектор Инна:** Да, давай начнём.

**Менеджер Миша:** Давайте расскажу сначала про новую фичу. У нас в приложении уже можно много всего сделать: заказать товары для животных с доставкой, смотреть фильмы про животных и телеканалы для животных. Заказчик хочет расширяться, поэтому во время брейншторма мы сгенерировали новую фичу — возможность заказать страховку для животных. По статистике, эта услуга становится всё более востребованной на рынке, особенно для собак. И спрос растёт к лету. Так что нам нужно быстро сделать новый раздел в приложении, где пользователь сможет выбрать подходящую страховку и купить её.

**Архитектор Инна:** А купить как, через корзину? Или нужна будет особая страховая форма?

**Менеджер Миша:** Нет, на этапе MVP пока никаких новых форм делать в приложении не нужно. Точно так же, как сейчас пользователь покупает корм для животных, он покупает и страховку: кладёт её в корзину и оплачивает. Операторы колл-центра потом сами свяжутся с клиентом и запросят всю дополнительную информацию. Если фича «полетит», тогда уже будем прикручивать формочку.

**Архитектор Инна:** Отлично! Тогда задача не выглядит сложной.

**Менеджер Миша:** Ну и не забываем, что нам нужно поддерживать ещё и англоязычную версию приложения. Учтите это, пожалуйста.

**Архитектор Инна:** Хорошо. Давай тогда расскажу немного Лёше и Дане про текущую архитектуру. И потом посмотрим, как будем реализовывать задачу.

**Менеджер Миша:** Да, конечно, передаю тебе слово.

**Архитектор Инна:** Начну с базы. До тех пор, пока у нас был только небольшой интернет-магазин, у нас был монолит. В определённый момент мы решили распилить его на микросервисы, и в итоге монолит распался на сервисы Store, User, Shopping Cart, Shipping и Payment. Store отвечает за каталог товаров и всё, что с ним связано. Сам каталог представлен БД MySQL и ElasticSearch для полнотекстового поиска. Это позволяет оптимизировать нагрузку на сервис. Сервис User, как следует из названия, отвечает за пользователей: хранит и управляет информацией о них. Там тоже используется база MySQL и Redis, чтоб можно было нормально кэшировать данные пользователей. Пока есть вопросы?

**Разработчик Лёша:** Нет, пока всё понятно. Расскажи про Shopping Cart, Shipping и Payment.

**Архитектор Инна:** Когда пользователь кладёт товар в корзину, идёт обращение к сервису Shopping Cart, где эта корзина создаётся. Реализовано это обращение через Kafka. Все новые сервисы, которые предполагают перемещение товаров в корзину, должны быть интегрированы с Shopping Cart. Все созданные корзины хранятся в базе данных, из которой также выполняет чтение CRM-система. Ну не напрямую, конечно, а через сервис. CRM используют операторы для отслеживания и управления заказами.

**Разработчик Лёша:** А СRM — это наша система или внешняя?

**Архитектор Инна:** Внешняя. Её нам подрядчик разрабатывает. Поехали дальше. Когда пользователь решает оформить заказ, то идёт обращение в сервис Shipping и Payment для оформления доставки и оплаты заказа соответственно. Доставка и оплата у нас осуществляются через внешние системы SHIP-SHIP и PAY-PAY. Оба сервиса взаимодействуют с нами через брокер сообщений.

**Разработчик Лёша:** Это все сервисы?

**Архитектор Инна:** Нет. У нас недавно ещё запустилась новая функция — просмотр разного рода контента про животных. Фильмы в основном. И для этого сделали отдельный микросервис Films. У него есть два хранилища: MySQL для описания контента, и файловый хостинг Amazon S3, где хранятся все медиафайлы.

**Разработчик Лёша:** Получается, что под страховку нужно делать новый микросервис? Или всё-таки добавим её в каталог товаров как обычный товар?

**Архитектор Инна:** Миша говорит, что, вероятно, услуга будет развиваться, и появятся дополнительные требования к ней. Так что я предлагаю сразу сделать микросервис под это. Если фича полетит, отлично. Если нет, то просто погасим сервис. А иначе, если всё-таки фича будет развиваться, потом ещё нужно будет вытаскивать её из другого микросервиса. Зачем нам эта головная боль?

**Разработчик Лёша:** Хорошо, тогда делаем отдельный микросервис. Если мы там только описание страховок будем хранить, то предлагаю использовать MongoDB для хранения контента.

**Архитектор Инна:** Да, ничего не имею против.

**Разработчик Лёша:** А как реализован доступ к микросервисам?

**Архитектор Инна:** Для пользователей у нас есть только один интерфейс — это мобильное приложение. Поэтому мы используем стандартный API Gateway. Потом выполняется проверка адреса инстанса в Service Registry, для этого выполняется query-запрос. Дальше уже через Load Balancer осуществляется доступ к сервисам.

**Даня:** А CRM тоже ходит через API Gateway?

**Архитектор Инна:** Да, верно. CRM тоже.

**Разработчик Лёша:** А какой стиль API используете?

**Архитектор Инна:** Для доступа к сервисам только REST.

**Разработчик Лёша:** Вроде картина понятна. Делаем новый микросервис.

**Менеджер Миша:** Только учтите, пожалуйста, что данные должны возвращаться быстро. А то у нас ранее уже были проблемы, когда сервис отвечал по несколько секунд.

**Архитектор Инна:** Да, хорошее замечание. Лёша, обрати внимание, что у нас стоят таймауты. Сервис должен возвращать ответ в пределах 10 секунд. Миша, а что по нагрузке?

**Менеджер Миша:** Мы прогнозируем, что в среднем около 5000 пользователей в сутки будут интересоваться продуктом, и около 100 пользователей выполнят покупку.

**Даня:** То есть порядка 4–5 запросов в минуту на новый сервис. И небольшое увеличение нагрузки на сервисы Shopping Cart, Shipping и Payment.

**Архитектор Инна:** Примерно так, да. Только на Shipping нагрузка не увеличится, так как доставки не будет. Но в целом 100 дополнительных запросов мы спокойно выдержим, это несущественно.

**Даня:** Понял. Я правильно понимаю, что в рамках новой задачи интеграция с новыми сервисами не требуется?

**Архитектор Инна:** Да, всё так. Нужно поднять новый микросервис. А Даниил пока обновит архитектурную схему и набросает профиль API, договорились?

**Менеджер Миша:** Да, Даня, я передам тебе все макеты. Это поможет в составлении API.

**Даня:** Договорились.

**Менеджер Миша:** Всем спасибо за встречу!