Краткий конспект по теме «Введение в распределенные системы обработки информации». Он на то и краткий, что затронет только важные моменты лекции, НЕ включая полное разъяснение каждого из пунктов

1. **Определение РСОИ:**
    1. Каждая информационная система имеет свою логику работы, причем с точки зрения других участников эта логика полностью не просматривается, система выглядит как некоторый "черный ящик".
    2. Обмен информации между участниками распределенной системы происходит по сетевым протоколам через публичные каналы доступа.
    3. Каждый участник РСОИ имеет свою базу данных, в которых хранит состояние своей модели предметной области. Прямой доступ на уровне SQL к этим БД другим участникам невозможен.
    4. Невозможно получить достоверное состояние другой системы.
    5. Нельзя гарантировать все участники системы работают.
2. **Требования:**
    1. Открытость
    2. Надёжность
    3. Масштабируемость

3. **Принципы построения сервис ориентированной архитектуры:**
    1. Стандартизированный контракт взаимодействия
    2. Слабая связность между сервисами
    3. Сервисная абстракция
    4. Повторное использование сервисов
    5. Сервисы не содержат состояние клиентов
    6. Автономность
4. **Плюсы:**
    1. Писать и поддерживать небольшие сервисы всегда проще, чем большие
    2. Горизонтальное масштабирование приводит к экономии денег, так как система может работать на множестве сравнительно недорогих машин
    3. Высокая стабильность
    4. Разнообразие технологий
    5. Значительно упрощается обновление системы, т.к. для добавления функциональности требуется обновить лишь часть системы, что с учетом правильно настроенного деплоя и умением других систем корректно обрабатывать недоступность, становится практически незаметно для пользователя.
5. **Ограничения:**
    1. Система должна работать быстро, т.к. теперь ко времени выполнения самих операций требуется прибавлять время взаимодействия по сети.
    2. Нужно иметь хорошую систему деплоя и развертывания новых виртуальных машин, т.к. количество сервисов будет расти и настройка всего окружения вручную будет проблематична.
    3. Требуется хорошее описание внешнего API.
    4. С появлением большого количество сетевого трафика накладывается большая ответственность на сеть и отказоустойчивость оборудования.
    5. В связи с этим, нужно при проектировании программы обязательно обрабатывать недоступность и ошибки от внешней системы.
    6. Часто становится очень трудно разбить систему на сервисы из-за сильной связанности данных.

6. **Микросервисная архитектура (МА)** — всего лишь набор более строгих правил и соглашений, как писать все те же сервисы SOA.
    1. Сервисы маленькие
    2. Сфокусированные
    3. Слабосвязные
    4. Высокосогласованные
7. **Характеристики микросервисов:**
    1. Разделение на компоненты (сервисы).
    2. Группировка по бизнес-задачам.
    3. Сервисы имеют бизнес-смысл.
    4. Умные сервисы и простые коммуникации.
    5. Децентрализованное управление.
    6. Децентрализованное управление данными.
    7. Автоматизация развертывания и мониторинга.
8. **Сложности реализации:**
    1. Сетевая инфраструктура надежная
    2. Нулевая задержка
    3. Бесконечная пропускная способность
    4. Сеть абсолютно безопасна
    5. Топология сети никогда не меняется
    6. Сетевой администратор только один
9. **Структура МА:**
    1. Frontend:
        1. Отдача статики, картинок, css и js – то есть всего, что не требует вычислений
        2. Кэширование
        3. Балансировка нагрузки между backend’ами
    2. Backend
    3. Data Storage
10. **Структура микросервиса:**
    1. Подсистема обмена сообщениями:
        1. ***передача сообщений;***
        2. ***валидация;***
        3. ***фильтрация.***
    2. Подсистема выполнения бизнес-операций.
    3. Подсистема работы с базой данных.