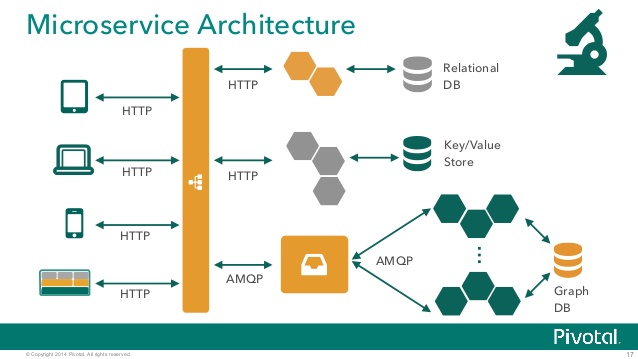
**Задания по проекту**

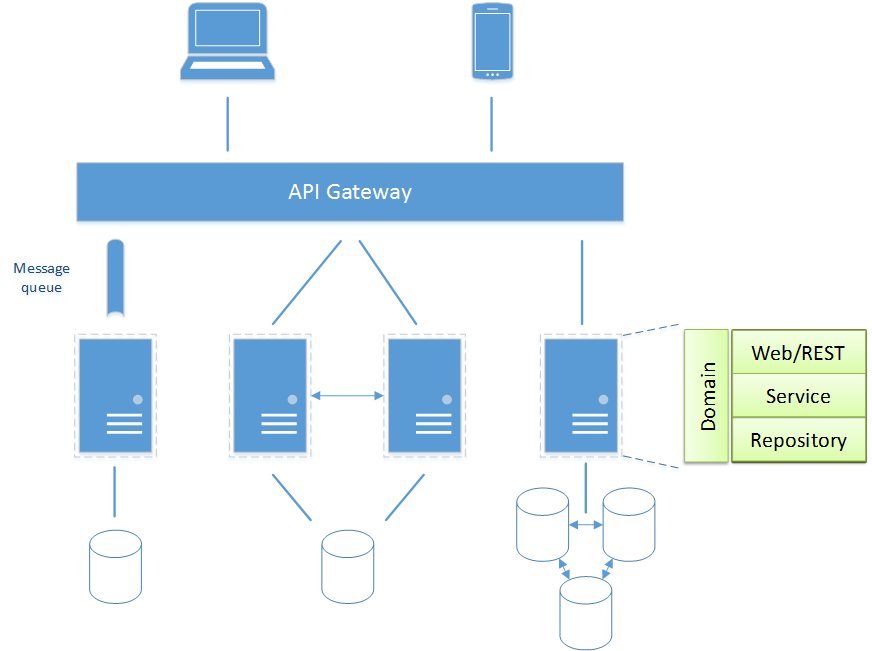
**Общая характеристика и требования к групповому проекту**

* Команда из 3-5 (6) человек
* Методологии разработки: Agile (SCRUM/Kanban)
* Среда для централизованной разработки:
  + Source code repository (GitHub, BitBucket, ..) - обязательно
  + SCRUM board - очень желательно
  + Team collaboration/Task/Issue tracking system - желательно
  + Continuous integration (СI) - желательно
* К зачету разработка должна быть закончена



**Требования к системе**

* система должна быть построена на основе микросервисной архитектуры
* каждый из микросервисов представлять собой трех-уровневое приложение, состоящее из слоя Веб (Web layer) или Native client, слоя бизнес-логики (Service + Domain layer) на сервере приложений/Веб-сервере, слоя доступа к данным (Repository/Persistence layer)
* для каждого микросервиса должна использоваться отдельная БД: MongoDB, Cassandra, Postgres …
* для одной из NoSQL БД должна быть настроена репликация, обеспечивающая ее отказоустойчивость. К примеру, в случае MongoDB если к-во оставшихся нод становится меньше кворума, то БД должна переходить в режим read-only
* для одного из микросервисов, сервера приложений должен дублироваться и быть настроены таким образом чтобы в случае отказа одного из серверов, запросы шли на другой сервер. При этом, данные пользователя (к примеру содержимое сессии) должны сохраняться. Для этого можно использовать in-memory data grid (distributed cache): Redis, Hazelcast, ...
* для одного из микросервисов запросы должны обрабатываться асинхронно с использованием message queue. Приветствуется реализация подхода event sourcing и CQRS
* клиентская часть приложения может непосредственно выступать агрегатором микросервисов или работать через некий API Gateway. Приветствуется построения приложения с использованием RESTful API
* микросервисы могут быть написана на любых из языков программирования



**Результаты разработки**

* Требования к системе в виде:
  + краткого описания предназначения (Vision),
  + сценариев / вариантов использования (use cases) / product backlog
* Диаграммы: архитектуры системы
* Исходный код системы в системе контроля версий
* Работающая система

Вся документация также может быть размещена в системе контроля версия