Семинар 3. Устойчивость и управление инцидентами

Принципы построения высоконагруженных систем

Георгий Семенов

Институт прикладных компьютерных наук
Университет ИТМО

осень 2025

Содержание

Оставшиеся активности и домашние задания

- Лекция + Семинар 4: Очереди, события и т.п.
- Домашнее задание 3: запрограммировать стратегии устойчивости
- Домашнее задание 4: небольшая курсовая работа с устной защитой

Вспоминаем лекцию

- SRE: SLA/SLO/SLI; расчет совокупного SLA
- Active-passive redundancy (failover): cold, warm, hot
- Active-active redundancy: load balancing, capacity planning
- Geographic redundancy
- Liveness probe, readiness probe
- Graceful degradation
- Circuit breaker
- Retry patterns (идемпотентность): retry, exponential backoff, jitter
- Bulkhead (изоляция ресурсов)
- Rate limiting (+ backpressure) - leaky bucket, token bucket, fixed/sliding window

Содержание

Каскадные сбои

Каскадные отказы – те, что распространяются с течением времени в результате положительной обратной связи[1].

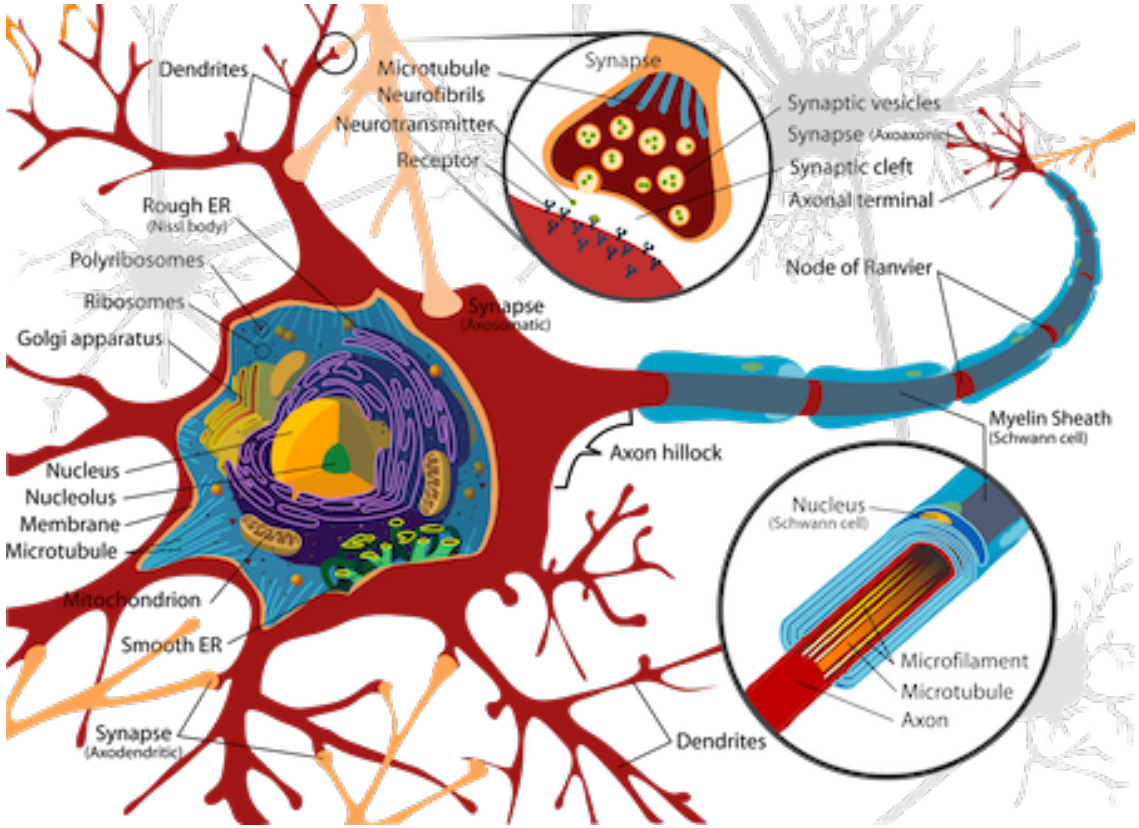
- Каскадный – т.е. прорывающийся вглубь через границы между компонентами системы
- Перегруженность сервера: приняли, подупали, поретраили, и сами себя заDDoS-или
- Истощение CPU: переполнение очередей, зависание потоков, троттлинг (>100%)
- Истощение RAM: падающие поды, «спираль смерти GC», снижение частоты попадания кэша
- и т.д.

[1] SRE Book. Глава 22 - Справляемся с каскадными сбоями

Нейрологическая метафора

Серотониновый шторм в синаптических щелях между нейронами:

- Должен быть «тонус»
- Переполнение приводит к плавной деградации (или нет – тогда шторм)



Как предотвращать каскадные сбои?

- Предварительно нагрузочно тестировать
- Graceful degradation: уметь отправлять деградированные результаты
- Rate limiting + backpressure
- Планирование производительности
- Управление очередями (thread pool -> concurrency, aka go, userver и т.п.)

Как мгновенно реагировать на каскадные сбои?

- Увеличить количество ресурсов
- Прекратить выполнять проверки на сбои/«гибель»
- Перезапускать серверы
- Отбрасывать трафик (в т.ч. «плохой»)
- Автоматические/ручные деградации

Проблема холодного старта

- Холодный кэш
- Холодный JIT (например, для Java)

Retries: модель ответов в HTTP

- **2xx** – успешный ответ с полезной нагрузкой в теле ответа.
- **4xx** – ошибка со стороны клиента (некорректный запрос, ошибка аутентификации или авторизации и т.п.).
- **5xx** – внутренняя ошибка сервера.
- **Timeout** – клиентский таймаут ожидания ответа от сервера на текущий запрос.

Retries: типы бюджетов

- **Retry budget** – максимальное количество подзапросов, которым мы можем нагрузить сервер за один запрос, в частности:
  - **Fast errors budget** – «быстрые» ошибки на подзапросах, т.е. с быстрым временем отказа (обычно `4xx`).
  - **Failures budget** – «тяжелые» ошибки, т.е. с некоторым ожидаемым временем работы до падения подзапроса (обычно `5xx`).
  - **Timeout budget** – ошибки типа «timeout», т.е. когда подзапрос завершается из-за истечения клиентского таймаута.
- **Latency budget** – общее время ожидания ответа от сервера на все попытки выполнения запроса.
  - в частности – **Subrequest latency budget** – время ожидания одного подзапроса.

Немного про ТМО

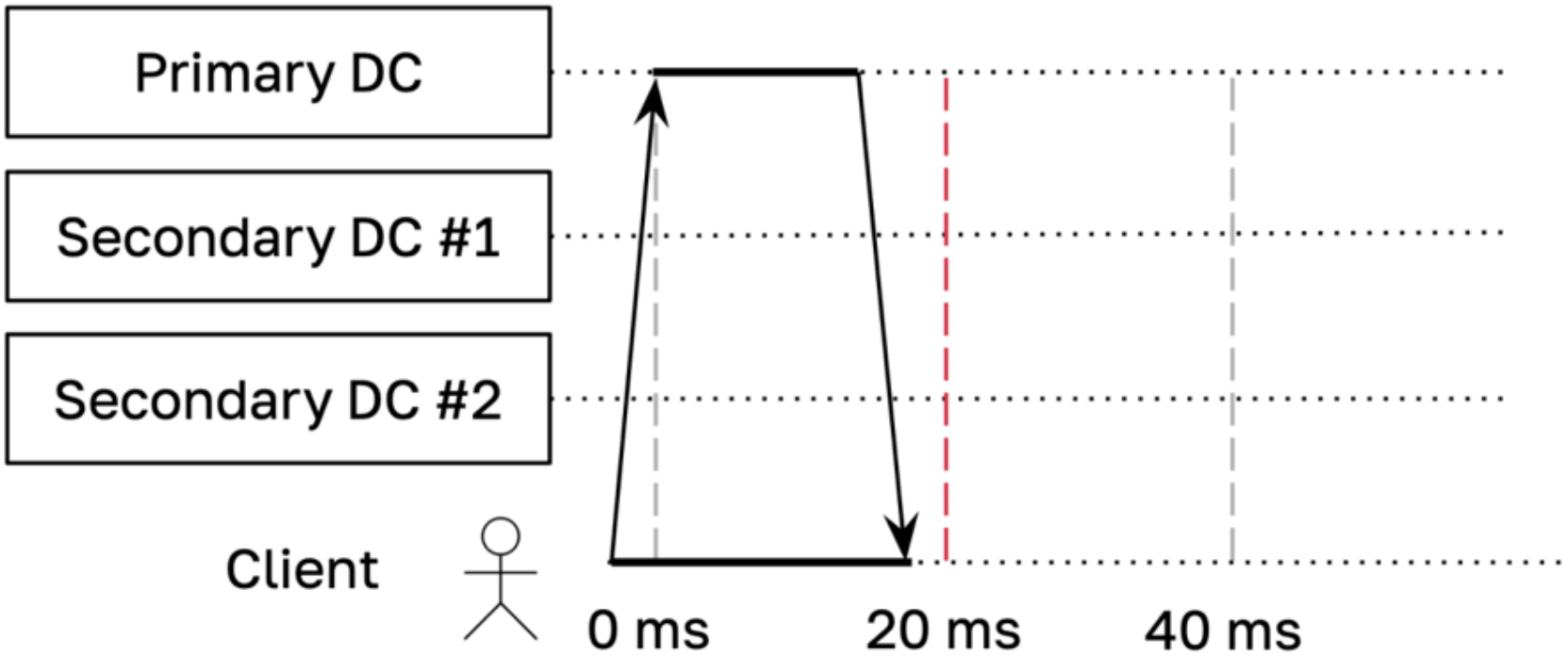
- Теория массового обслуживания - раздел математики
- Центральный пример - распределение Пуассона (в ДЦ каждый день меняют по три диска)

Хеджирование

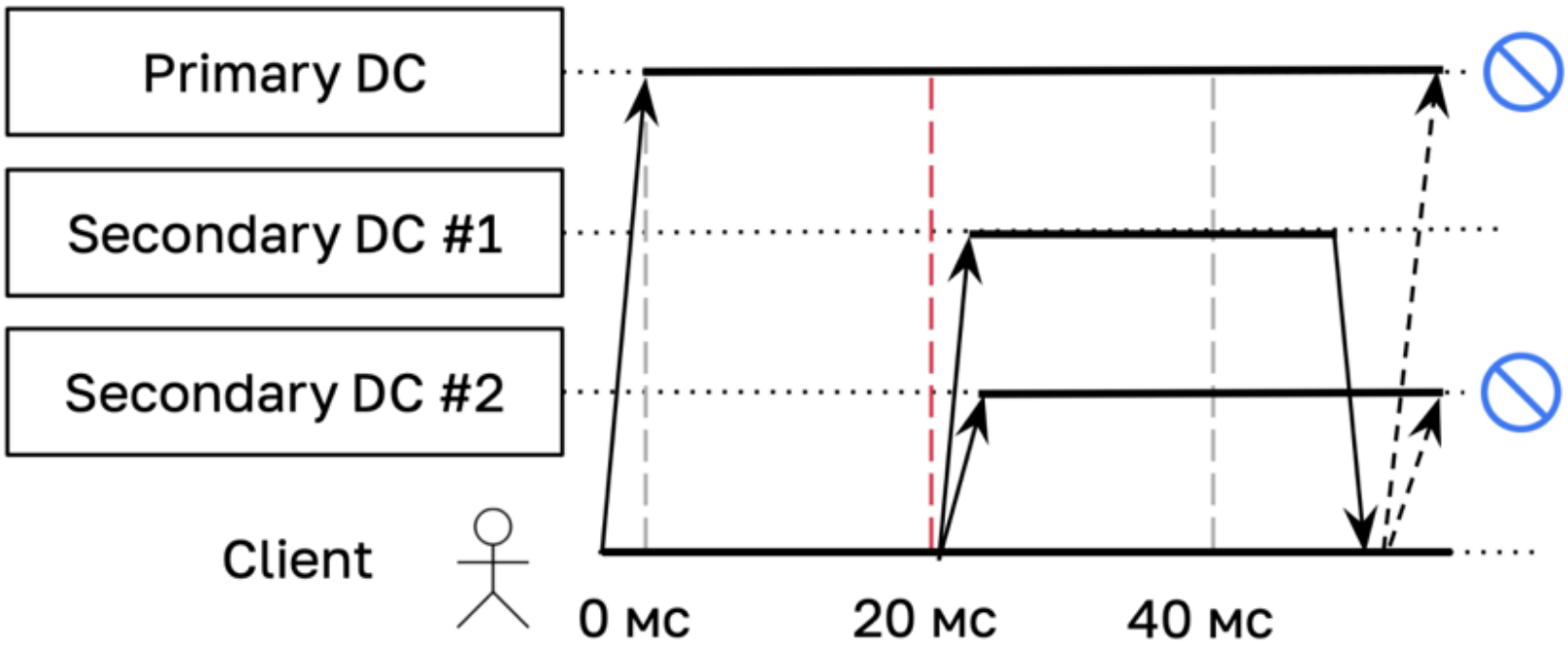
Хеджирование – спекулятивная повторная отправка дополнительных запросов с целью минимизировать время ответа на запрос.

- Первый подзапрос должен отправляться на первый по порядку *интерфейс отправки подзапроса* из их списка.
- Если в течение параметризуемого времени `hedging_delay` не приходит ответ на первый подзапрос, следующие подзапросы должны параллельно отправляться на оставшиеся *интерфейсы отправки подзапроса*.
- Стратегия завершается успешно, когда приходит первый успешный ответ от любого из подзапросов.
- Стратегия завершается ошибкой, когда все попытки завершились или исчерпан бюджет на latency запроса.

Хеджирование: case 1



Хеджирование: case 2



Содержание

Инцидент - спекулятивное определение

- В 3 ночи поднимаются лиды и разработчики
- Дружно каскадно перезапускают упавшие сервисы, пока утром не польется активный трафик
- Сидят в зуме и интересуются, как кого что импактит внутри

Кто такая дежурная смена?

- Существуют линии поддержки - Support Lines:
- SL1 - поддержка пользователей (принимает сообщения об отказах)
- SL2 - дежурная смена (24 на 7) + алертинг + мониторинг (подтверждает идентификацию и эскалацию инцидентов)
- SL3 - ответственные команды и разработчики

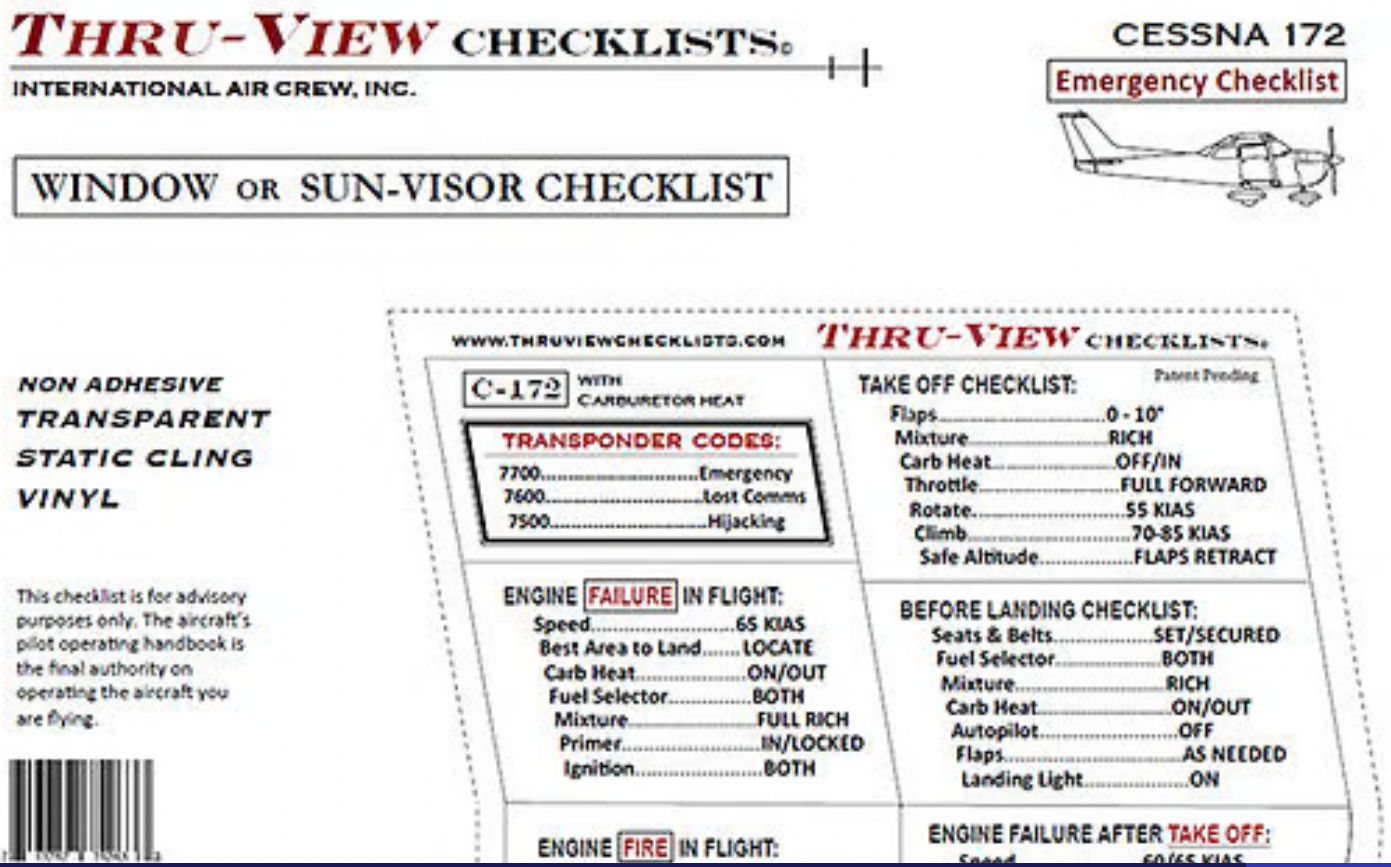
Эскалация - глоссарий

- Вызванивать
- Будить
- Подключаться
- Влияние
- Подскажите, какой сейчас статус

Авиа- метафора

Человек должен «успевать» за системой:

- Интерпретируемость автоматического реагирования системы на отказы
- Адекватность архитектуры и ее мониторинга
- Искусство координации инцидентным звонком



Post mortem

- Инцидент - это срабатывание риска, связанное с упущенной прибылью
- Цель реагирования – как можно скорее **устранить влияние**
- Цель разбора – **установить причину** инцидента, чтобы устранить ее
- Разбор не должен обвинять никого, он должен быть нейтральный

Как разбирать инцидент?

Проблемные вопросы:

- Что случилось? (и как долго?)
- Как (кто?) заметил? (благодаря случайности или процессу?)
- Какие были warning sings?
- Как можно оценить влияние?
- Какие action items можно выделить для устранения причин (предотвращения в будущем)?
- Что в итоге сделали?

Содержание

IRL: обновление DNS в пятницу вечером

- Пятница вечер, 22:00 – с подов сервиса пропадает доступ (обнаруживается через мониторинг сервиса) к двум из трех балансировщиков СУБД
- Трафик на оставшийся кластер возрос в 3 раза
- Выясняется, что в это время обновили конфигурацию DNS (добавили ipv6 AAAA-записи)
- Вопрос 1: почему это помешало работе системы?
- Вопрос 2: какую можно установить причину для инцидента?

BGP routing issue 2021 (6 часов)

Некорректные манипуляции с BGP привели к 6-часовой недоступности Facebook, Instagram и WhatsApp

05.10.2021 08:47

Facebook столкнулся с крупнейшим сбоем в своей истории, в результате которого все сервисы компании, включая facebook.com, instagram.com и WhatsApp, оказались недоступны в течение 6 часов - с 18:39 (MSK) в понедельник до 0:28 (MSK) во вторник. Источником сбоя стало изменение в настройках BGP на магистральных маршрутизаторах, управляющих трафиком между датацентрами, которое привело к каскадному нарушению связности датацентров Facebook с остальной глобальной сетью. Со стороны произошедшее выглядело так, как будто кто-то разом отключил кабели ото всех датацентров Facebook.

Интересно, что сбой привёл к нарушению работоспособности внутренних информационных систем и систем связи, из-за чего сотрудники, большая часть которых работала удалённо, не смогли подключиться к инфраструктуре и связаться с коллегами, что существенно усложнило работы по восстановлению, так как ключевые сетевые инженеры также работали удалённо. Более того, возникли проблемы с получением физического доступа, так как идентификационные карты сотрудников и система контроля доступа в помещения были завязаны на централизованные сервисы и также перестали работать.

AWS S3 Outage 2017 (4 часа)

- February 28, 2017, from mid-morning to early afternoon PST (around 9:40 AM to 12:36 PM PST)
- Location: Primarily the US-East-1 (N. Virginia) region, the primary AWS region.
- Cause: An authorized S3 team member executed a command with an incorrect input while trying to debug a slow billing system, inadvertently removing more servers than intended, affecting crucial S3 subsystems.
- An authorized S3 team member executed a command with an incorrect input while trying to debug a slow billing system, inadvertently removing more servers than intended, affecting crucial S3 subsystems.

CloudFlare Outage 2019 (27 мин)

The events of July 2

On July 2, we deployed a new rule in our WAF Managed Rules that caused CPUs to become exhausted on every CPU core that handles HTTP/HTTPS traffic on the Cloudflare network worldwide. We are constantly improving WAF Managed Rules to respond to new vulnerabilities and threats. In May, for example, we used the speed with which we can update the WAF to push a rule to protect against a serious SharePoint vulnerability. Being able to deploy rules quickly and globally is a critical feature of our WAF.

Unfortunately, last Tuesday's update contained a regular expression that backtracked enormously and exhausted CPU used for HTTP/HTTPS serving. This brought down Cloudflare's core proxying, CDN and WAF functionality. The following graph shows CPUs dedicated to serving HTTP/HTTPS traffic spiking to nearly 100% usage across the servers in our network.

Содержание

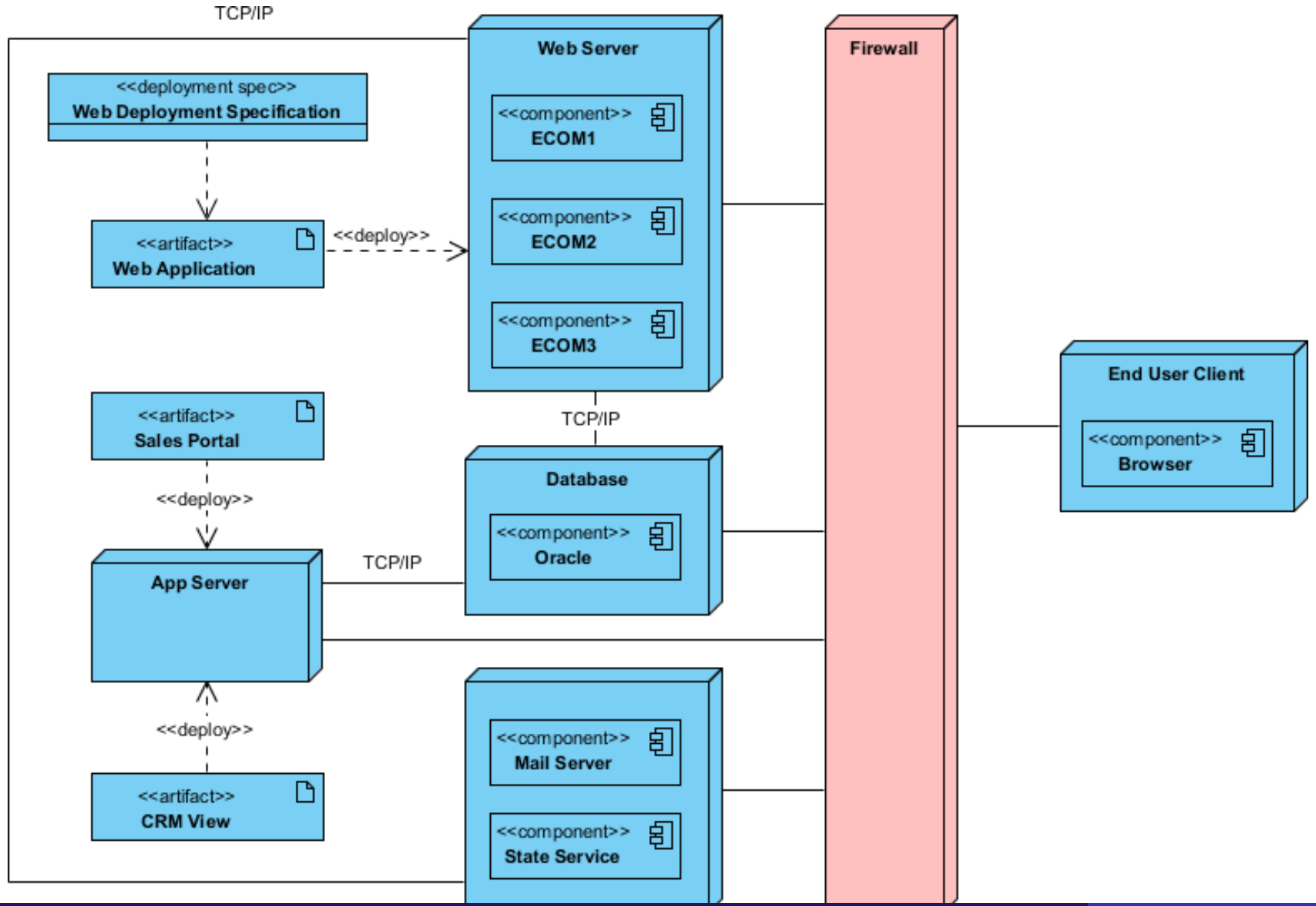
Архитектура

Набор компонентов и связей между ними (но не только); набор решений, необходимых для разработки системы:

- Системная архитектура - какие сервисы, как расположены, как общаются
- Программная архитектура – парадигма программирования, модульная организация, паттерны и т.п.
- Архитектура данных – какие данные, схема хранения, СУБД и т.п.

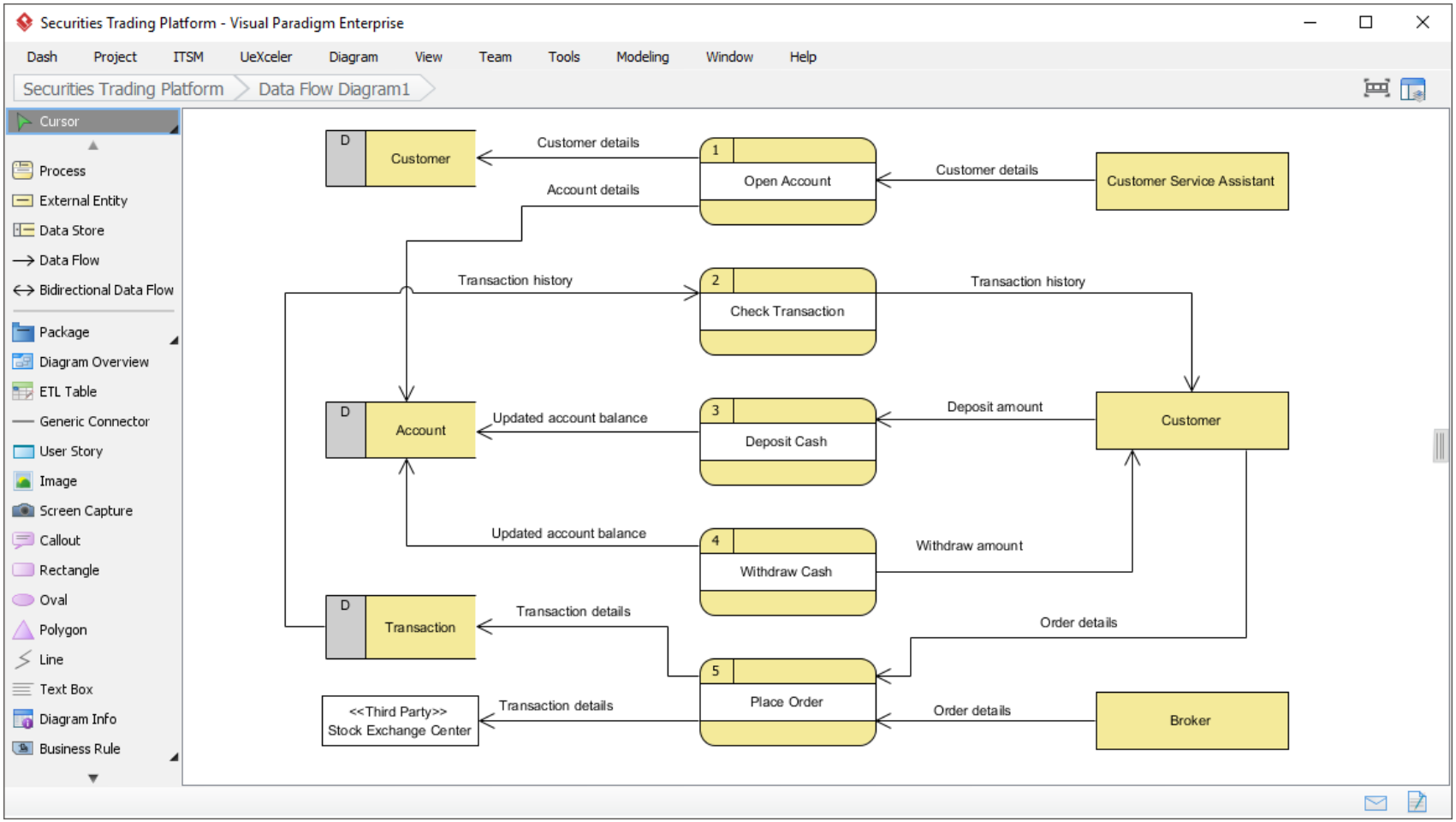
Диаграмма развертывания

Можно нарисовать в Visual Paradigm Online.



Dataflow diagram

Можно нарисовать в Visual Paradigm Online.



Как пройти процесс проектирования?

- Итеративный подход – задаем себе вопросы по кругу, пока не успокоимся
- Не является ли компонент X единой точкой отказа? Как это исправить? – Исправляем.
- Моделируем ситуации отказа сразу же при проектировании – отказ сервиса, сетевой разрыв, заполнение СУБД, DDoS и т.п. – моделируем инцидент, исправляем первопричины

Содержание

Итоги

- Обсудили паттерны устойчивости и каскадные сбои
- Погрузились в аспекты IM
- Выдано третье + четвертое домашнее задание