

РАЗРАБОТКА СОВРЕМЕННОЙ АРХИТЕКТУРЫ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ РЕШЕНИЯ КОРПОРАТИВНЫХ ЗАДАЧ

© 2018 А. Г. Юрочкин, Д. Ю. Жулябин

*Российская академия народного хозяйства и государственной службы
при Президенте Российской Федерации (г. Воронеж, Россия)
ОАО «Пигмент» (г. Воронеж, Россия)*

Статья посвящена анализу современной архитектуры веб-приложения для решения корпоративных задач. Представлена многослойная архитектура системы.

Ключевые слова: веб-приложение, архитектура, корпорация.

Нашей целью является создание прототипа архитектуры системы, позволяющей оценить структуру приложения. Потребителями архитектуры могут стать как IT-специалисты, разработчики так и менее подготовленная аудитория.¹

Для создания общего представления необходимо провести ряд действий.

1. Определиться с типом приложения. В данном случае это веб-приложение, так, как только этот тип отвечает выдвинутым требованиям.

2. Выбрать вариант развертывания. Предполагается распределенное развертывание приложения.

3. Определить архитектурный стиль. Основным архитектурным стилем, используемым при проектировании архитектуры будет многослойная архитектура.

Многослойная архитектура дает возможность сгруппировать связанную функциональность приложения в разные слои, выстраиваемые вертикально, поверх друг друга. Где функциональность каждого из слоев объединяется общей ответственностью или ролью. Слои слабо связаны, и между ними осуществляется явный обмен данными. Корректное деление приложения на слои дает возможность поддержать строгое разделение функциональности, что, создает удобство, гибкость и простоту обслуживания приложения. При строгом делении на отдельные слои компоненты одного слоя имеют возможность взаимодействовать только с компонентами, расположенными на

том же слое или теми, что расположены в слое прямо под данным слоем. При более свободном разделении на слои появляется возможность компонентам слоев взаимодействовать с компонентами того же и всех нижестоящих слоев.

Общие особенности проектирования с использованием многослойной архитектуры:

– данный вариант архитектуры представляет систему как единое целое, обеспечивая при этом достаточное количество деталей для понимания ролей и ответственностей слоев и отношений между ними;

– при проектировании нет необходимости делать предположения о методах, типах данных, реализации или свойствах так как они скрыты в рамках слоя;

– деление функциональности между различными слоями очень четкое, верхние слои, посылают команды нижним и могут реагировать на события, происходящие в этих слоях, давая возможность передачи данных между различными слоями вниз и вверх;

– точное разделение границ ответственности для каждого из слоев и включение в слой только той функциональности, которая связана с его задачами, может обеспечить высокую связность в пределах слоя;

– отсутствие зависимости между верхними и нижними слоями дает потенциальную возможность их повторного использования;

– слабое связывание, для реализации слабой связанности между слоями связь между ними основывается на событиях и абстракции.

Многослойная архитектура рекомендуется при условии, что в приложении должны поддерживаться различные устройства и типы клиентов или может потребоваться

Юрочкин Анатолий Геннадьевич – Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации, д. т. н., профессор, yuroch_kin912@yandex.ru.
Жулябин Дмитрий Юрьевич – ОАО «Пигмент», специалист, zhulyab_0912f@yandex.ru.

создание сложных бизнес-правил и процессов.

При проектировании многослойной архитектуры для разделения приложения на слои было применено распределение слоев по функционалу, на слой бизнес-функциональности, слой представления, слой доступа к данным и слой сервисов.

Деление модулей на слои представлено ниже.

- Слой представления:
 - веб-интерфейс пользователя, администратора или эксперта системы;
 - мобильный интерфейс.
- Слой сервисов:
 - API доступа к системе и получению данных.
- Слой бизнес-логики:
 - фасад приложения, отвечающий за взаимодействия со слоями представления и слоем сервисов;
 - модуль интерактивного взаимодействия пользователей;

- модуль постановки и решения задач;
- модуль администрирования системы.
- Слой доступа к данным:
 - модуль работы с базой данных;
 - модуль работы с базой знаний;
 - модуль поиска данных для решения задач;
 - модуль администрирования методов и моделей.

Распределение слоев и компонентов приложения по разным физическим уровням. На первом этапе разработки для проектирования «пилотной версии приложения» предполагается нераспределенное развертывание. Учитывая то, что для прототипа приложения не требуется наличие границ доверия между слоями, то компоненты бизнес-слоя и слоя представления целесообразно расположить на одном уровне (рис. 1). В дальнейшем потребуются отделение бизнес-логики и слоя доступа к данным на отдельный сервер.

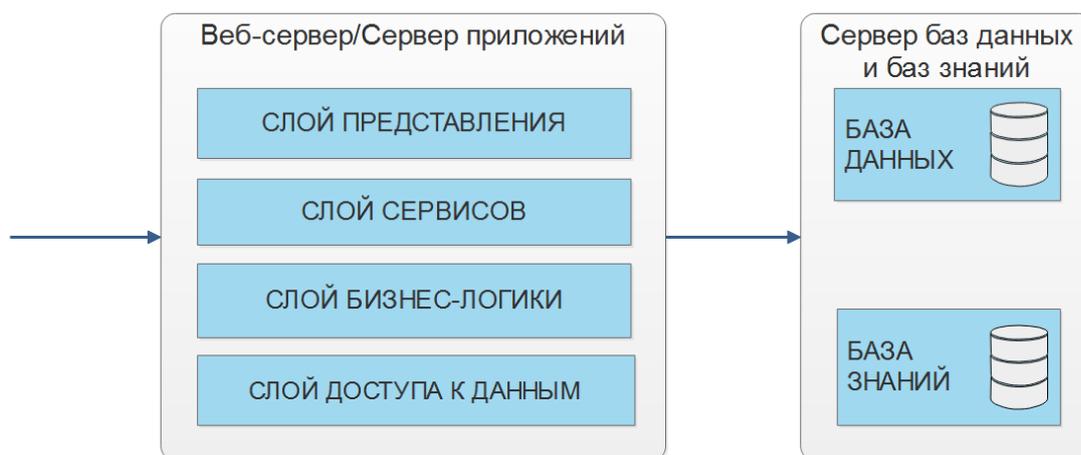


Рисунок 1. Иллюстрация модели развертывания системы.

Взаимодействие между слоями. В системе предполагается строгое взаимодействие, когда каждый должен взаимодействовать только со слоем, расположенным непосредственно под ним. Такое правило позволяет обеспечить строгое разделение, когда каждому из слоев известно только о том слое, который расположен под ним. Важной особенностью такого варианта взаимодействия является то, что при изменениях в интерфейсе слоя, это затронет только слой, расположенный над ним. Это позволяет проще расширять функционал в дальнейшем.

Сквозная функциональность. Это функциональность, охватывающая все слои приложения, определение сквозных функций улучшает обслуживание и возможность повторного использования. Возможности этих модулей системы должны быть доступны всем слоям, даже если слои физически располагаются на разных уровнях. В нашей системе к сквозной функциональности относятся:

- аутентификация и авторизация;
- мониторинг работы системы;
- протоколирование;
- кеширование;
- валидация;

– связь.

Многослойная архитектура системы представлена на рисунке 2.

– *Интерфейсы между слоями.* Главная их цель слабое связывание между слоями, что бы слою не требовалось раскрывать внутренние детали, от которых может зависеть другой слой. Интерфейс должен быть спроектирован таким образом, чтобы минимизировать зависимости, предоставляя открытый интерфейс, скрывающий детали слоя. Наиболее подходящий для системы вариант, это интерфейс, основанный на обмене сообщениями. Взамен прямого взаимо-

действия с компонентами других слоев используя доступ к их свойствам или вызов методов, возможно применить связь посредством обмена сообщениями в качестве реализации интерфейсов и для взаимодействия между различными слоями. Данный интерфейс дает возможность определять контракты уведомления о сбоях, схемы данных, операции, политики системы безопасности и т. д. Этот вид интерфейса рекомендуется применять при разработке веб приложений, и приложений с различными типами клиентов.

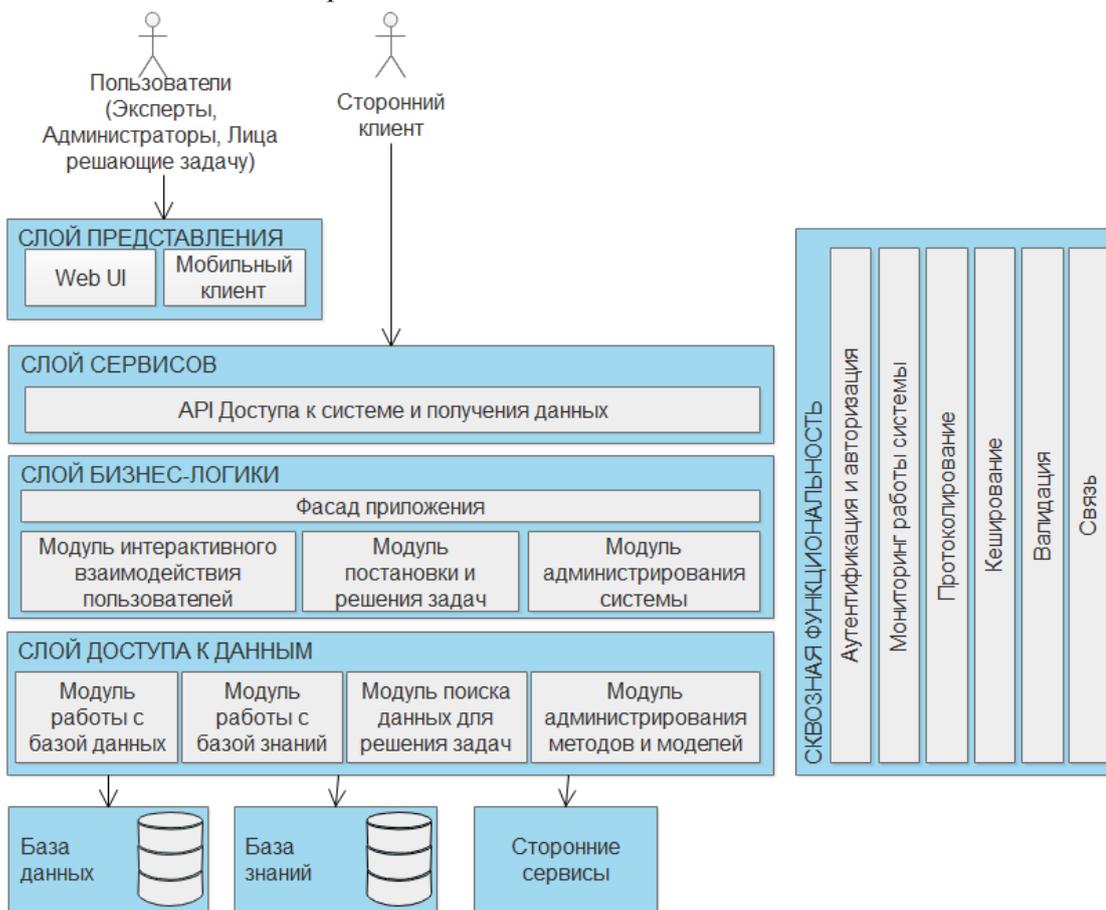


Рисунок 2. Представление многослойной архитектуры системы.

Выбор стратегии развертывания. Так как в дальнейшем предполагается высокая нагрузка на систему, проектируя приложение необходимо учесть возможности распределенного развертывания. Взаимодействие должно осуществляться посредством протокола TCP с бинарным кодированием и посредством обмена сообщениями, которое должно обеспечить достаточную производительность, и возможности избежать привязки к конкретному серверу, что дает возможности для горизонтального масштабирования системы и позволяет балансировку на-

грузки. Предполагаемый вариант развертывания представлен на рисунке 3.

Выбор протоколов связи. Для работы системы предполагаются стандартные протоколы, такие как TCP для связи по внутренним сетям и HTTP/HTTPS для связи через Интернет.

Кеширование должно использоваться для сокращения вызовов между браузером и веб-сервером и между веб-сервером и нижестоящими серверами. Кеширование существенно влияет на производительность системы, особенно когда проект становится «высокона-

груженным» и возрастает нагрузка на все компоненты системы. В системе должны кэшироваться редко меняющиеся данные и статические страницы. Также может быть оправдано кэширование базы данных, когда первое

обращение идет в кэш, и лишь при отсутствии там данных идет обращение в базу данных. Но это может быть необходимо реализовать на более поздних стадиях проекта.

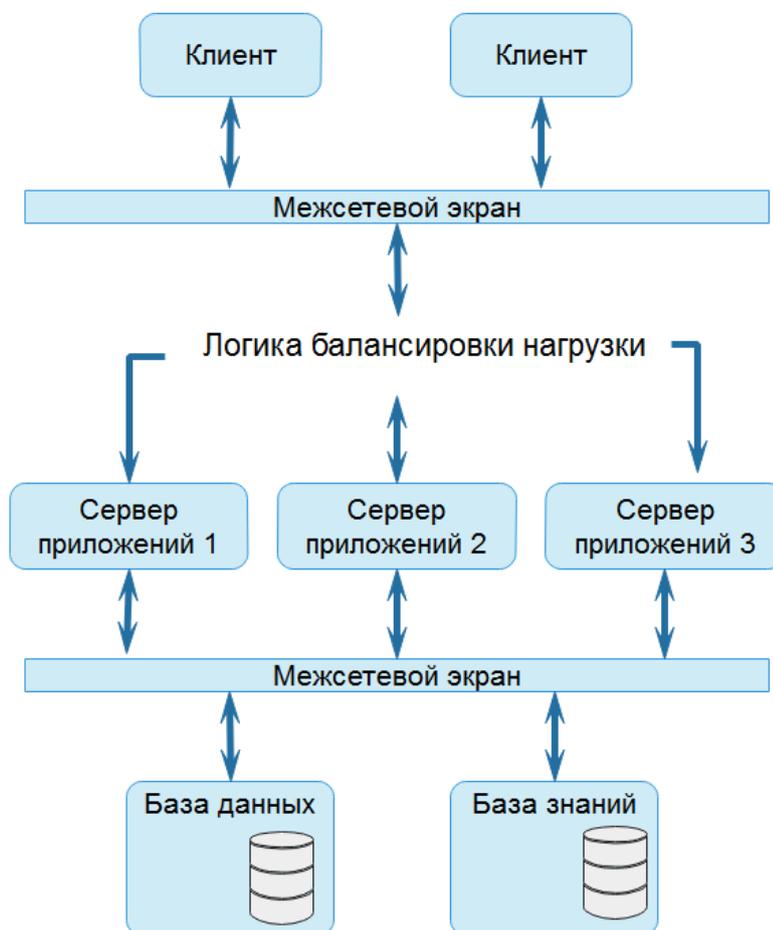


Рисунок 3. Возможный вариант развертывания приложения.

Протоколирование и инструментирование. В системе необходимо осуществлять протоколирование и аудит действий. Журналы регистрации событий обычная практика для профилактики атак на систему и регистрации действий в системе. Должен выполняться аудит критических событий, критических бизнес-операций, необычных действий. Доступ к журналам должен быть только для записи.

Аутентификация пользователей должна проходить на границах доверия, например, при удаленном доступе к бизнес-слою со слоя представления. Конфиденциальные данные, например, пароль или cookie аутентификации, должны передаваться с использованием протокола Secure Sockets Layer (SSL).

Необходимо учесть различные аспекты безопасности на разных этапах работы сис-

темы. Основные аспекты безопасности представлены на рисунке 4.

Авторизация определяет какие действия в системе может выполнять и к каким ресурсам имеет доступ аутентифицированный пользователь. Авторизация должна выполняться при пересечении границ доверия. Для доступа к страницам и каталогам должна использоваться авторизация URL.

Валидация позволяет снизить вероятность внедрения вредоносного кода в систему и повысить уязвимость к другим видам атак. С этой целью необходимо проверять все данные вводимые в систему, должна проводиться проверка длины, диапазона, формата, типа вводимых данных проверка должна проходить и на стороне пользователя, и на стороне сервера.

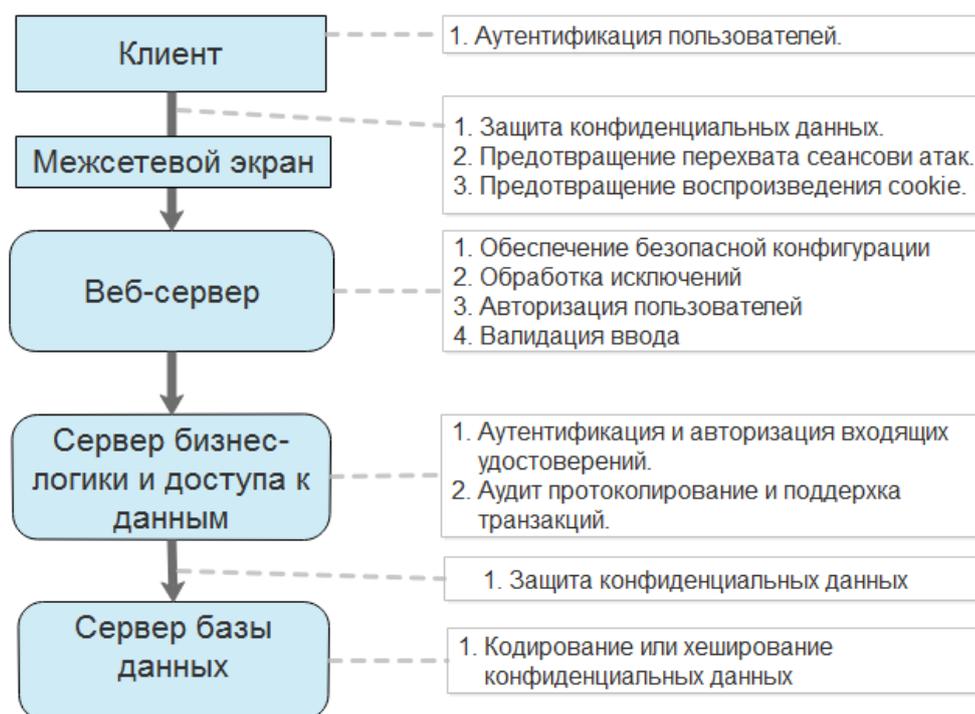


Рисунок 4. Основные аспекты безопасности, которые необходимо учесть в системе.

ЛИТЕРАТУРА

1. Бузунов А. Н. Особенности информационно-диагностических систем / А. Н. Бузунов // Международный студенческий научный вестник. – 2018. – № 3-3. – С. 387-391.
2. Лавлинская О. Ю. Обзор инструментальных средств и технологий разработки веб-приложений / О. Ю. Лавлинская, М. С. Бондаренко // Вестник Воронежского института высоких технологий. – 2016. – № 4 (19). – С. 73-78.
3. Львович И. Я. Особенности проектирования корпоративных компьютерных сетей / И. Я. Львович, А. П. Преображенский, О. Н. Чопоров // Оптимизация и моделирование в автоматизированных системах. Материалы всероссийской молодежной научной школы. Министерство образования и науки РФ, Воронежский государственный технический университет, Российский фонд фундаментальных исследований. – 2017. – С. 16-20.
4. Львович И. Я. Применение информационных технологий в медицинской сфере / И. Я. Львович, А. П. Преображенский, О. Н. Чопоров // Интеллектуальные информационные системы. Материалы всероссийской конференции с международным участием. – 2017. – С. 164-165.
5. Львович И. Я. Основы информатики / И. Я. Львович, Ю. П. Преображенский, В. В. Ермолова / учебное пособие, Воронеж,

Издательство: Воронежский институт высоких технологий. – 2014. – 339 с.

6. Львович Я. Е. Проблемы построения корпоративных информационных систем на основе web-сервисов / Я. Е. Львович, И. Я. Львович, Н. В. Волкова // Вестник Воронежского государственного технического университета. – 2011. – Т. 7. – № 6. – С. 8-10.

7. Львович Я. Е. Сравнительный анализ методов поиска текстовых фрагментов / Я. Е. Львович, А. В. Косых // Информатика: проблемы, методология, технологии. Сборник материалов XVIII международной научно-методической конференции: в 7 т. Под редакцией Н. А. Тюкачева; Воронежский государственный университет. – Воронеж. – 2018. – С. 37-40.

8. Максимов И. Б. Принципы формирования автоматизированных рабочих мест / И. Б. Максимов // Вестник Воронежского института высоких технологий. – 2014. – № 12. – С. 130-135.

9. Максимов И. Б. Классификация автоматизированных рабочих мест / И. Б. Максимов // Вестник Воронежского института высоких технологий. – 2014. – № 12. – С. 127-129.

10. Преображенский Ю. П. Характеристики информационно-образовательного пространства вуза / Ю. П. Преображенский // Антропоцентрические науки: инновационный взгляд на образование и развитие личности. Материалы VII

Международной научно-практической конференции. – 2018. – С. 218-219.

11. Тамбовцев Г. А. Программный продукт для тестирования в образовательном учреждении / Г. А. Тамбовцев // Студенческая наука для развития информационного общества. Сборник материалов VII Всероссийской научно-технической конференции. – 2018. – С. 105-109.

12. Тамбовцев Г. А. Разработка системы, предназначенной для анализа активно-

сти пользователей / Г. А. Тамбовцев, А. А. Маркова, А. В. Косых // Международный студенческий научный вестник. – 2018. – № 3-3. – С. 426-428.

13. Шкирман С. О. Разработка и реализация алгоритма распределения ресурсов в сложной системе / С. О. Шкирман // Международный студенческий научный вестник. – 2018. – № 3-3. – С. 438-442.

THE DEVELOPMENT OF MODERN ARCHITECTURE WEB APPLICATION FOR THE SOLUTION OF CORPORATE PROBLEMS

© 2018 A. G. Yurochkin, D. Yu. Zhulyabin

*Russian Academy of national economy and public administration under the President of the Russian Federation (Voronezh, Russia)
JSC «Pigment» (Voronezh, Russia)*

The paper is devoted to the analysis of the modern architecture of the web application for solving corporate problems. The multilayer architecture of the system is presented.

Key words: web application, architecture, corporation