

Асратян Р.Э., Владимирова С.С. Курако Е.А., Орлов В.Л.

Особенности обеспечения технологической независимости в разработках систем с сервис-браузерной архитектурой

Аннотация: Проведен анализ существующих архитектур клиент-серверного взаимодействия на основе веб-технологий для разработки информационных систем. Рассмотрены минимально необходимые принципы для клиентского программного обеспечения, позволяющие реализовать надежные клиентские приложения с учетом требований технологического суверенитета и импортозамещения. Предложено решение задачи быстрого создания информационных систем с заданными параметрами защиты и удобства разработки.

Ключевые слова: распределенные системы, импортозамещение, технологический суверенитет, веб-архитектура, сервисы

Импортозамещение информационных систем в настоящее время становится одной из наиболее актуальных и неотложных задач с точки зрения обеспечения технологического суверенитета страны [1]. Следует учитывать, что при разработке систем такого рода необходимо иметь целый спектр базовых программ, которые составляют определенный фундамент как для проектирования, так и для организации эксплуатации разработанных программных комплексов. Этот фундамент в процессе разработки обычно содержит следующие компоненты:

- операционная система (ОС);
- система управления базой данных (СУБД);
- язык (языки) программирования (ЯП);
- библиотеки и (или) фреймворки (БФ);
- интегрированная среда разработки (ИСР);
- программа организации работы сервера (ПОРС);

– программа организации работы клиента (ПОРК).

В процессе эксплуатации системы этот список, как правило, сокращается до пяти позиций:

ОС;

СУБД;

БФ;

ПОРС;

ПОРК.

Рассмотрим, какие же конкретные компоненты могут обеспечить технологическую независимость.

Обычно при построении информационных систем используются ОС Windows и Linux. Но здесь и в последующем нужно иметь в виду, что нам необходимо ориентироваться на программные продукты, входящие в реестр российского программного обеспечения [2]. К сожалению, Windows не включена в этот реестр. Там также отсутствуют операционные системы, позволяющие запускать программные модули, оттранслированные для Windows. Это определяется, прежде всего, закрытостью ОС. Таким образом, необходимо ориентироваться на Linux-подобные системы. Таких систем, переработанных для использования в России и включенных в российский реестр достаточно много. Например, Astra Linux, Альт, Ред ОС, Atlant, Роса и другие. Пожалуй, наибольшее распространение получила система Astra Linux.

В качестве СУБД в последнее время в России обычно используются PostgreSQL и Postgres Pro. Нужно отметить, что Postgres Pro базируется на открытой платформе PostgreSQL. Заметим, что PostgreSQL (определенные версии) входит в состав Astra Linux и, таким образом, при использовании этой операционной системы данные версии считаются российскими.

Выбор языка программирования для создания крупных информационных систем фактически сводится к рассмотрению альтернативы, включающий два языка – C# и Java. Каждый из этих языков, по существу, предназначен для этой цели. Остальные языки в этом случае также используются, но значительно реже, так как имеют свою специфику. Кроме того, C# и Java широко распространены (что немаловажно), и в обоих управление памятью вынесено за конструкции языка, что позволяет избежать ряда

существенных и непросто определяемых ошибок программирования. Язык C# создан позднее, что позволило ему включиться в систему .NET, обеспечивающую использование одного фреймворка для программ, написанных на различных языках. Кроме того, удалось улучшить процесс сборки и выполнения сложных программ, позволяющий в результате несколько увеличить скорость выполнения. Именно поэтому язык C# часто используется в процессе разработки информационных систем.

В качестве интегрированной среды разработки – ИСР (или IDE – Integrated Development Environment) для Windows-подобных систем широкое распространение получил пакет MS Visual Studio. Его аналог в реестре российского программного обеспечения отсутствует. Возможно использование открытой интегрированной среды разработки в среде Linux, например, MonoDevelop, которая хоть и не включена в реестр, но предоставляет исходные коды, которые можно проверить и оттранслировать.

Для организации работы сервера чаще всего используется программный комплекс, который включает «web-сервер». Для Windows – это IIS (Internet Information Services). В Linux таковым является Apache 2, по сути, представляющий собой модульную платформу с открытым исходным кодом. Web-сервер представляет собой программу, обеспечивающую прием запросов от клиентов, передачу их обработчику, возврат ответов запрашивающему.

Отдельный вопрос – выбор программы организации работы клиента. Казалось бы, без программ этого типа можно было бы вовсе обойтись. Действительно, достаточно для каждого типа клиента написать свой код и установить соответствующие модули на определенные компьютеры-клиенты. Но это резко затрудняет сопровождение, так как непросто проводить в процессе эксплуатации различные изменения на множестве компьютеров, особенно в ситуации, когда в любой момент конфигурация как в аппаратной, так и в программной части может измениться. Естественным представляется выход, когда программы для клиента загружаются в процессе работы по его инициативе. В этом случае сервер только дает разрешения и отвечает на запросы, то есть не думает о каждом клиенте и не обслуживает его без обращения.

1. Использование браузеров для организации работы клиента

Итак, для удобного сопровождения клиента требуется управляющая программа, которая будет загружать необходимые программы по запросу клиента и инициировать их выполнение. Если взглянуть на набор уже существующих программ, то естественным выбором для этой цели становится web-браузер.

Действительно, в современных web-браузерах можно вызывать и отображать не только html-страницы, содержащие текст и рисунки. На этих страницах могут размещаться и программы, написанные на языке JavaScript, которые могут выполняться на клиентском месте.

Вроде все есть. На компьютере-клиенте изначально нет никаких программ информационной системы. Эти программы подгружаются с сервера. Если программы обновились, то при следующем вызове идет загрузка новых версий. Казалось бы, все проблемы решены. И закрываются глаза на то, что у web-браузеров:

- программы клиента могут выполняться только в режиме интерпретации, что снижает быстродействие;
- идет разделение по языкам. То есть для клиента можно использовать только один язык – JavaScript, а для серверной части используются другие более мощные языки. Практика показывает, что писать один программный продукт на разных языках – возможно, но крайне неудобно;
- снижается гибкость программирования при использовании языков более высокого уровня, которые обеспечивают и серверные вычисления и формируют клиентские страницы, включающие JavaScript;
- существует избыточность web-браузеров, которые включают множество функций, не используемых в информационных системах. Иногда обращение к этим функциям приводит к выходу за пределы информационной системы, а значит снижению уровня защиты;
- отсутствуют встроенные средства обеспечения безопасности для клиента, что требует создания их для каждой системы, а это повышает трудоемкость разработки.

Именно поэтому возникает потребность в разработке специализированного браузера, который свободен от этих

недостатков, и в то же время включает все необходимые методы для применения в качестве управляющего средства клиентских приложений. Основы построения сервис-браузерной архитектуры такого браузера разработаны в ИПУ РАН [3-4]. Но реальное использование сервис-браузеров на начальном этапе было возможным только в среде Windows.

2. Построение сервис-браузера в среде Linux

Для обеспечения технологической независимости в России необходимо организовать работу сервис-браузера в Linux-подобной среде. В таком случае он сможет работать в зарегистрированных российских системах, например, Astra Linux, Alt и других.

Проблема состоит в том, что перевод сервис-браузера в среду Linux не является простым действием. И это прежде всего зависит от того, что в Windows вместе с языком C# используется фреймворк Microsoft.NET Framework, а в Linux-подобных системах необходимо применять пакет «Моно» [5], который присутствует в реестре российских программ. И тут нужно учитывать несколько обстоятельств.

Прежде всего, следует иметь ввиду, что интерфейс системных функций в Windows и Linux весьма различен. Кроме того, в процессе работы с той или иной операционной системой необходимо учитывать ее особенности.

Существенно также то, что клиент активно работает с графическим интерфейсом, а его формирование в Microsoft.NET Framework и Моно весьма различно. А так как мы работаем с клиентом, где организация экранного отображения является весьма объемной задачей, то это приводит к тому, что довольно значительные фрагменты кода необходимо изменять.

Важно также учитывать различия в крипто-библиотеках, использующихся в разных системах.

Набор элементов для графического интерфейса за время существования .Net стал весьма разнообразным. На текущий момент в Моно нет такого разнообразия, но постепенно набор элементов будет расширяться. Для переноса сервис-браузера в среду Linux потребовалось создать три дополнительных графических компонента:

- C_Grid – библиотека для работы с таблицами;

- C_PopUpCalendar – виджет выбора даты;
- C_Print - библиотека организации печати.

В результате удалось разработать сервис-браузер, работающий в среде Linux. Нужно отметить, что вместе сервис-браузером в новую среду мигрирует также базовый модуль «Администратор системы», с помощью которого производится настройка параметров функционирования информационной системы. Прикладные модули требуется переносить самостоятельно, используя правила и компоненты, разработанные в процессе создания сервис-браузера.

Литература:

1. В поисках альтернативы: варианты импортозамещения ПО в России. – URL: <https://aif.ru/boostbook/importozameshchenie-po.html?ysclid=ikwlespbu729683692#solut> (дата обращения 07.08.2023).

2. Реестр российского программного обеспечения. – URL: <https://reestr.digital.gov.ru/> (дата обращения 15.12.2023).

3. Курако Е.А., Орлов В.Л. Сервис-браузеры для информационных систем // Программная инженерия. – 2017. – Том 8, № 9. – С. 413-421. – DOI: 10.17587/prin.8.413-421.

4. Курако Е.А., Орлов В.Л. Способ организации взаимодействия клиента с сервером приложений с использованием сервис-браузера: Патент на изобретение RU 2656735 С1; Зарегистрирован 06.06.2018. Заявлено 17.05.2017. Опубликовано: 06.06.2018 Бюллетень № 16.

5. Моно – Лаборатория 50 (lab50.net). – URL: <https://lab50.net/моно> (дата обращения 10.01.2024).

Чебан А.Г., Анисимова Е.А.

Принципы организации и построения защищенных систем видеоконференции

Аннотация: В работе рассмотрены принципы организации и построения защищенных систем видеоконференцсвязи, рассмотрены современные технологии шифрования