

Одиннадцатая независимая научно-практическая конференция «Разработка ПО 2015»

22 - 24 октября, Москва



Опыт интеграции приложений в SAP.
Толстые против тонких, родные против веб базированных.

Алексей Арсеньев
SAP SE

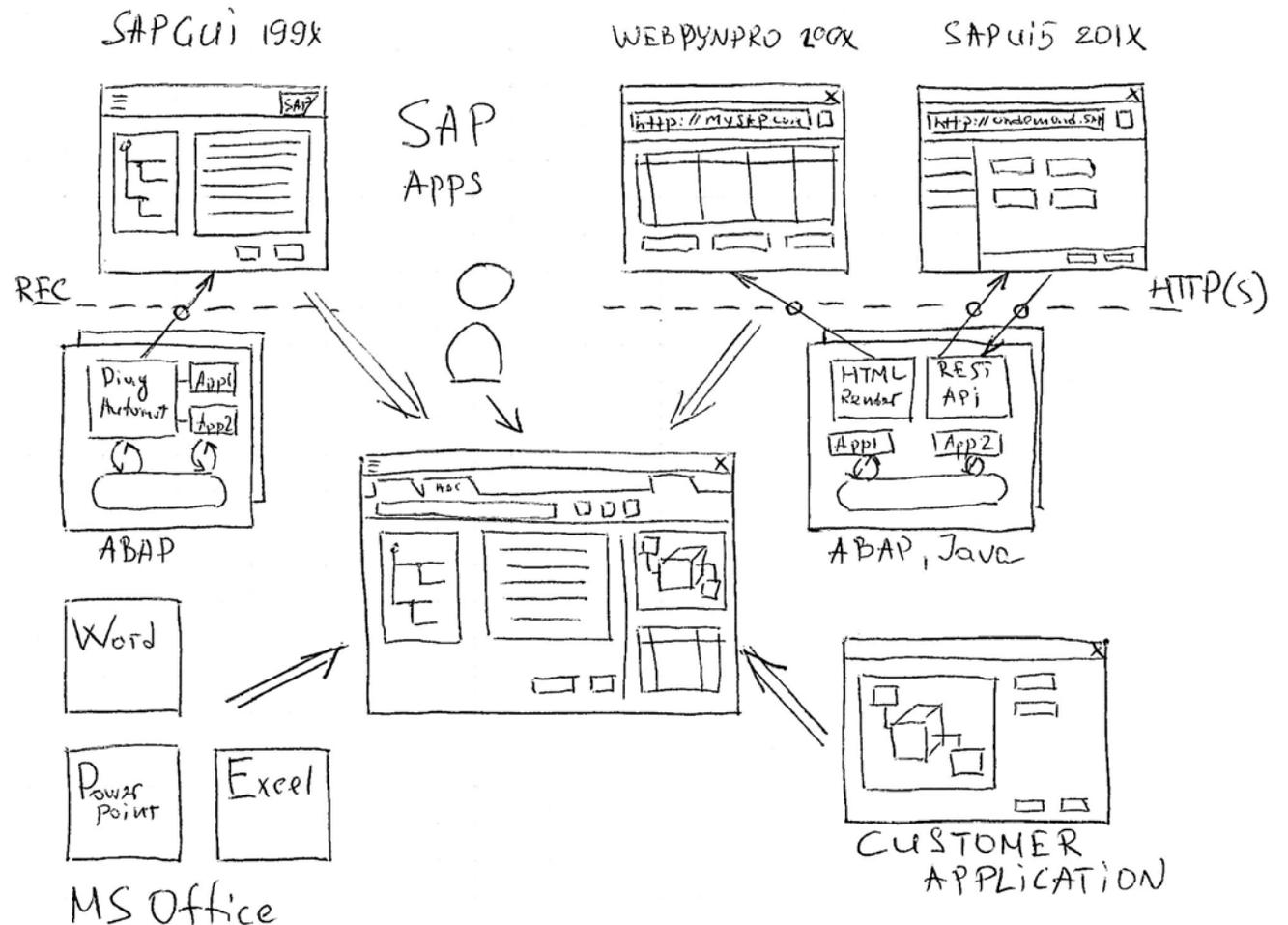
Введение

Зачем надо интеграционное решение

- Единая точка входа для всех приложений
- Единый пользовательский опыт
- Единая тематизации и брендинг
- Взаимодействие между новыми и старыми приложениями

Задачи для решения

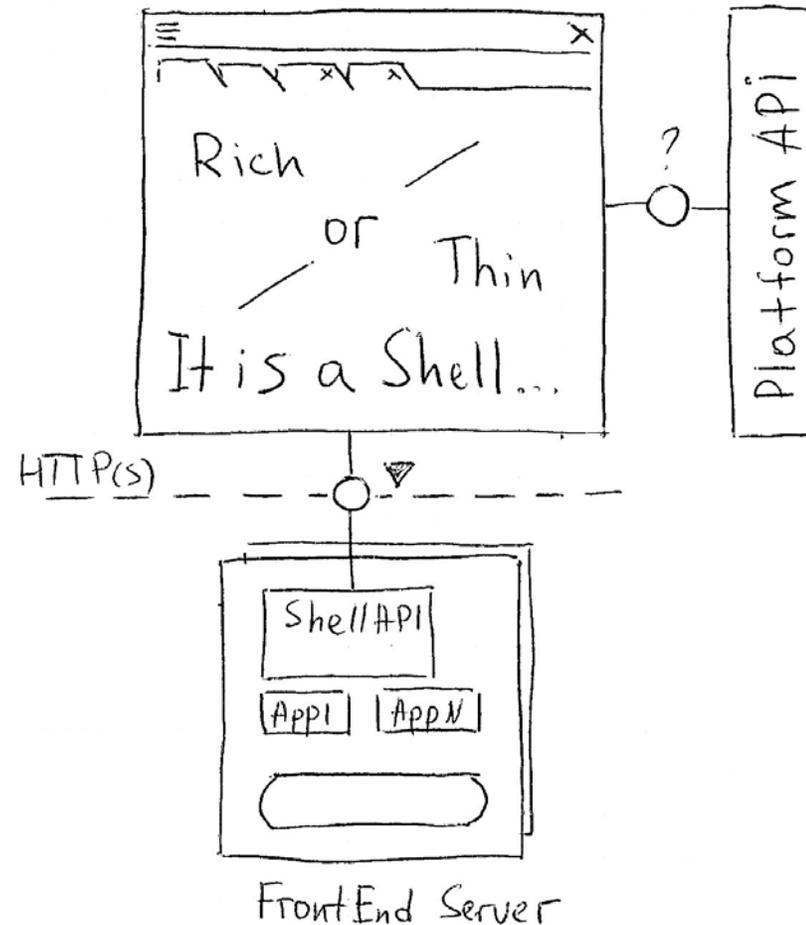
- Технологический зоопарк
- Ожидания пользователей



Веб-интерфейс против родного клиента

Веб-интерфейс

- Отсутствие затрат на установку
- Синхронизированные обновления
- Платформено-независимый пользовательский опыт



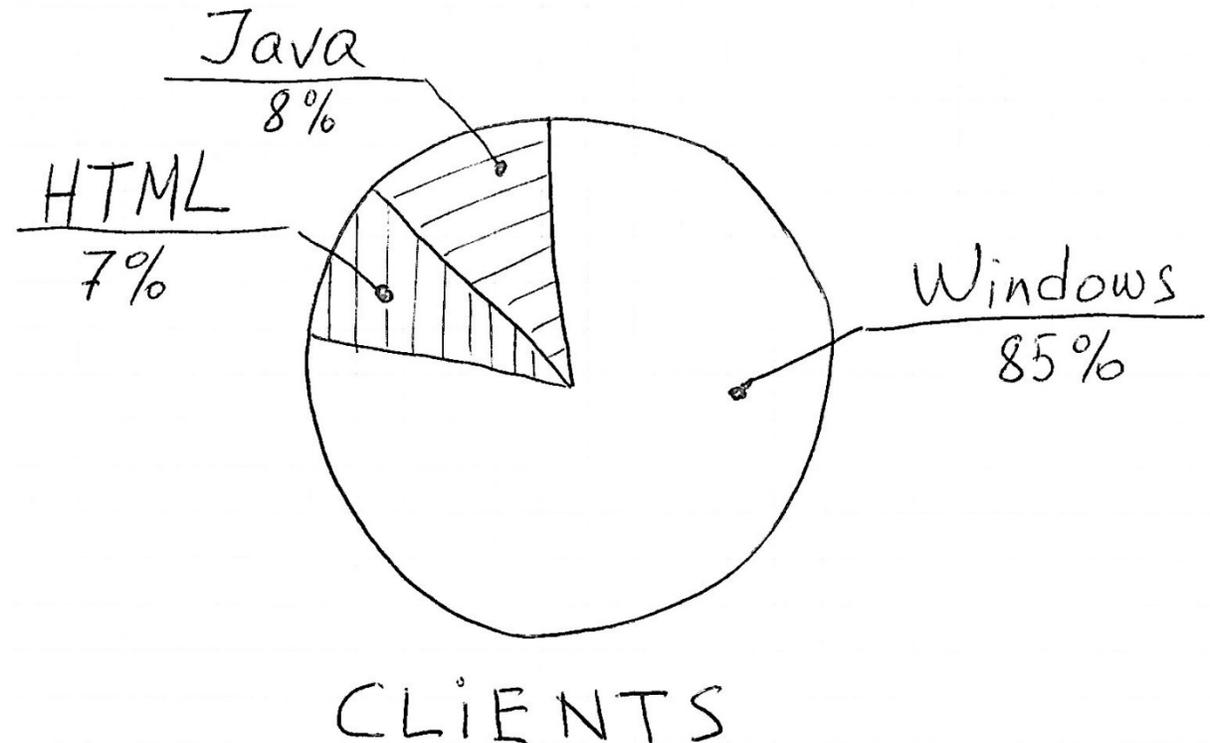
Родной клиент

- Может эффективно интегрировать родные компоненты
- Доступ к сервисам платформы
- Более эффективное кэширование
- Быстрее время запуска
- Родные органы управления
- Нет ограничений / зависимостей браузера

Выбор платформы для клиента

Почему Windows?

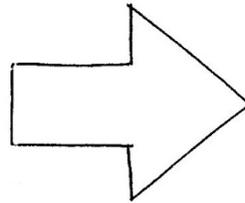
- Что использует корпоративный бизнес?
 - Windows
 - MS Office
 - Internet Explorer
 - MS Exchange
- Что предпочитают бизнес-пользователи?
 - Windows Client ~85%
 - HTML Client ~7%
 - Java Client ~8%
- Нужны ли альтернативы?



Технология клиента

Что нужно?

- Оболочка для интеграции бизнес-приложений
- Быстрая разработка
- Устоявшиеся библиотеки пользовательского интерфейса
- Легкая расширяемость и поддержка
- Простая интеграция родных компонентов



C#/.NET

- Для Windows
- Короткое время разработки
- WPF + XAML
- Логика клиента не требует высокой производительности
- Блоки C ++ можно использовать для критически важных задач

Alternatives

- Java и AWT/Swing или JavaFX
- **C++ и QT**
- C++ и Chrome UI
- C++ и HTML/JS UI

Встраиваемый браузер

Что нужно?

- Простая интеграция веб-приложений
- Никаких дополнительных инсталляций и зависимостей
- Долгосрочная и надежная поддержка
- Встраивание API



MS Internet Explorer

- Хорошо документирован
- Стабильный, доступен на всех Windows
- Бизнес-контент разработан для IE
- Богатый API
- Легкое встраивание COM объектов
- Нет лицензионных сборов

Альтернативы

- **Chrome (CEF)**
- Firefox (GeckoFX)

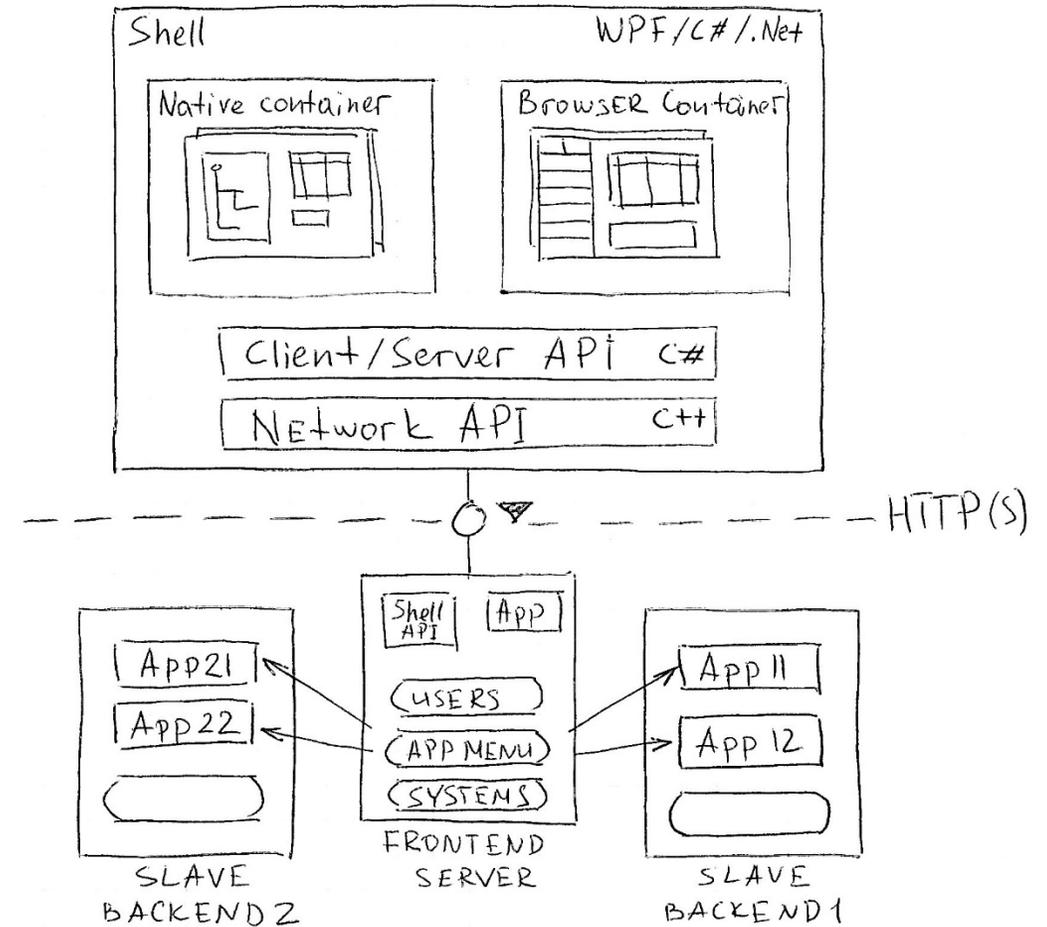
Шаблон архитектуры

Клиент

- WPF / C # / .NET для оболочки и UI
- C ++ библиотеки для сетевых протоколов
- Интегрированный веб-браузер
- Контейнер для родных приложений
- Клиент/серверный API
- Доступ к каталогу систем

Сервер

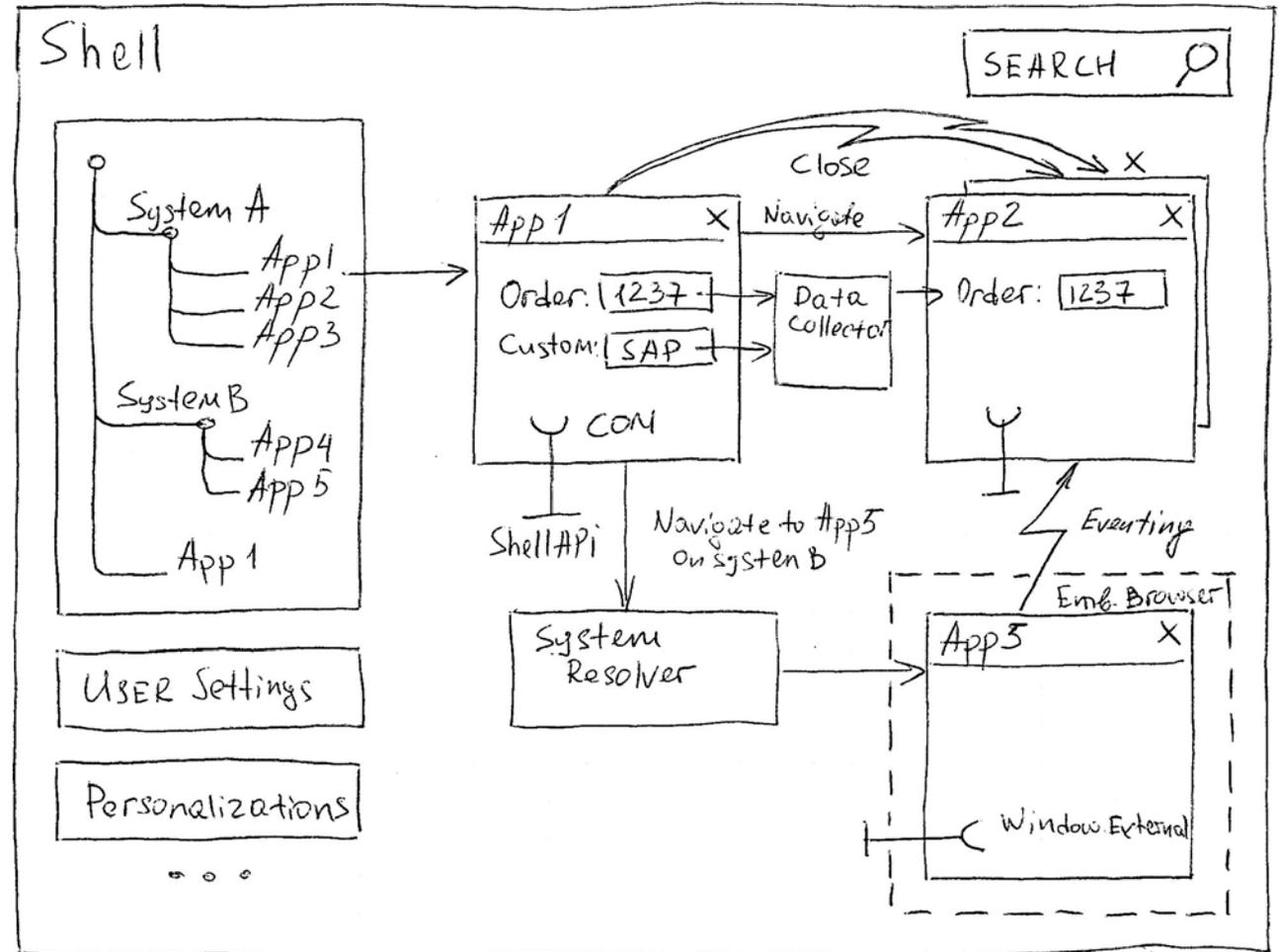
- Контент, выбранный на основе роли пользователя
- Управляющая система для Single Sign On
- Пользовательские предпочтения
- Федерация контента
- Настройки зависимых систем



Функции оболочки

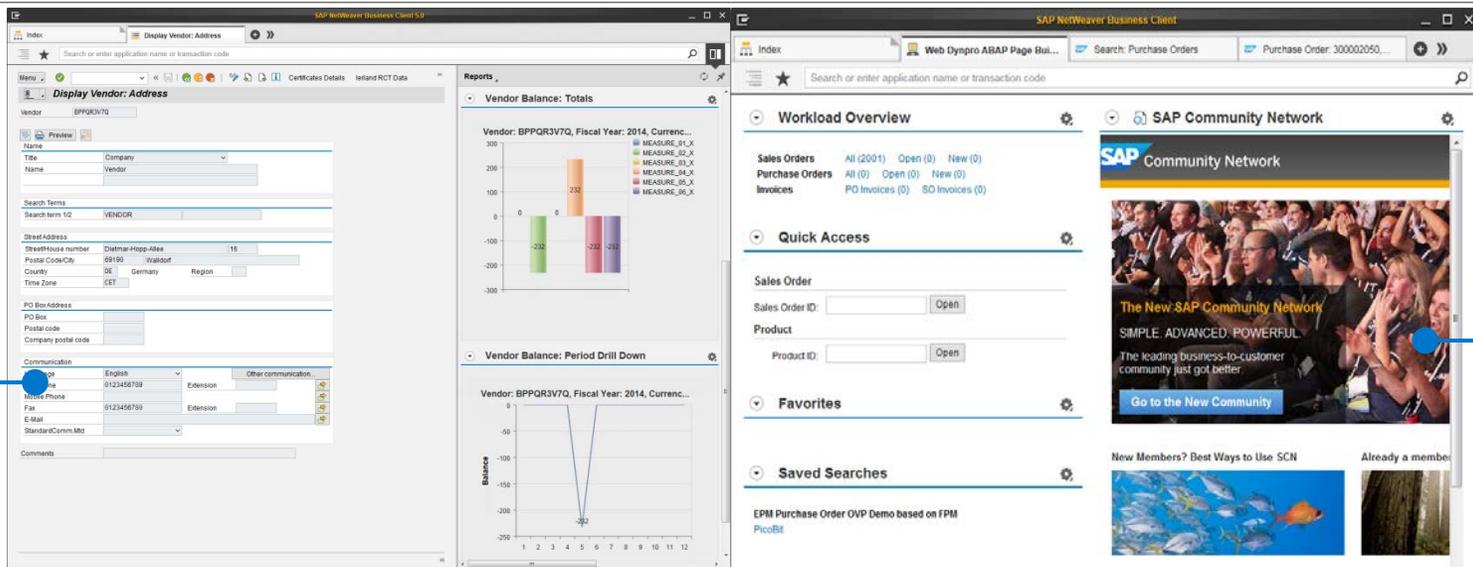
Какие сервисы нужны

- Навигация
 - Обзор и поиск приложений
 - Разрешение системных псевдонимов
 - Разрешение бизнес целей
- Обмен данными
 - Сбор данных приложения через UI
 - Информирование о UI событиях
- Поиск
- Централизованный контроль сессий
- API контейнеров приложений
 - Родных
 - Веб базированных



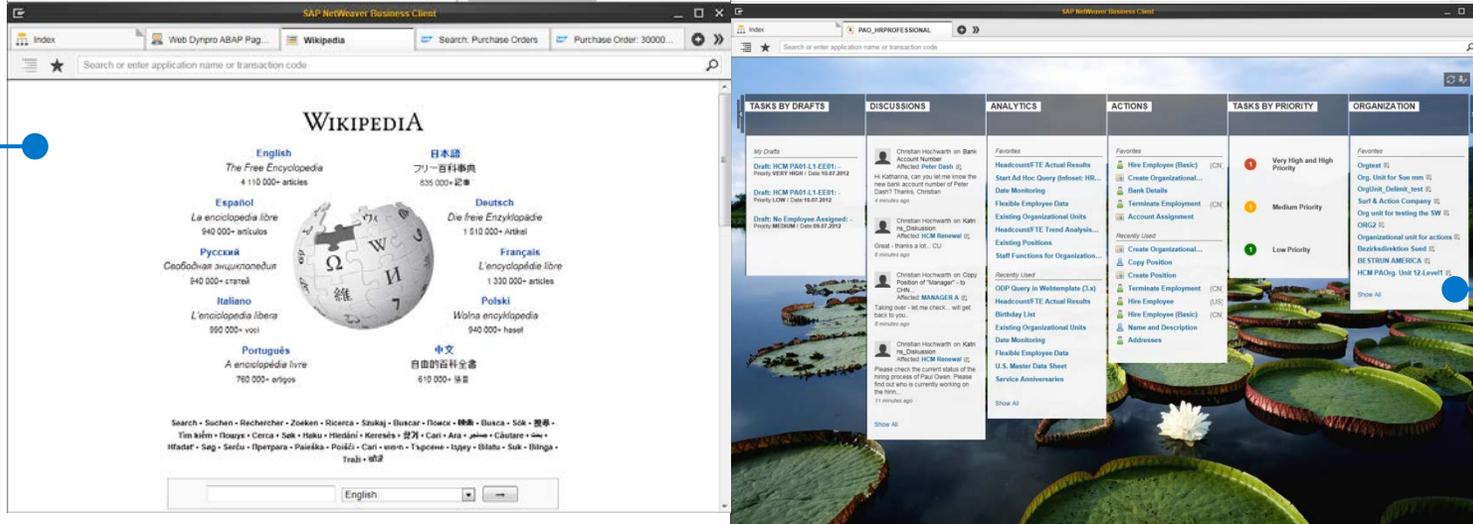
Что в итоге

Родные +
панели-
расширения



HTML приложения
(серверная HTML
генерация)

Внешние



Приложения на
JavaScript UI



Спасибо

Контактная информация:

Алексей Арсеньев

Архитектор программного обеспечения

alexey.arseniev@sap.com

<http://de.linkedin.com/in/arseniev>