

пользователей будут сначала поступать на шлюз «Ak Kamal e-Security Suite» и только после этого передаваться на Web-приложения организации. Это значит, что ни один пользователь не сможет получить прямой доступ к Web-приложению.

Такой подход позволяет не акцентировать внимание на области применения Web-приложения и обеспечивать безопасность в любых информационных системах.

УДК: 004.429

SOFTWARE AS A SERVICE Жакупова А.Т, Толегинова М.Н.

Университет имени Сулеймана Демиреля

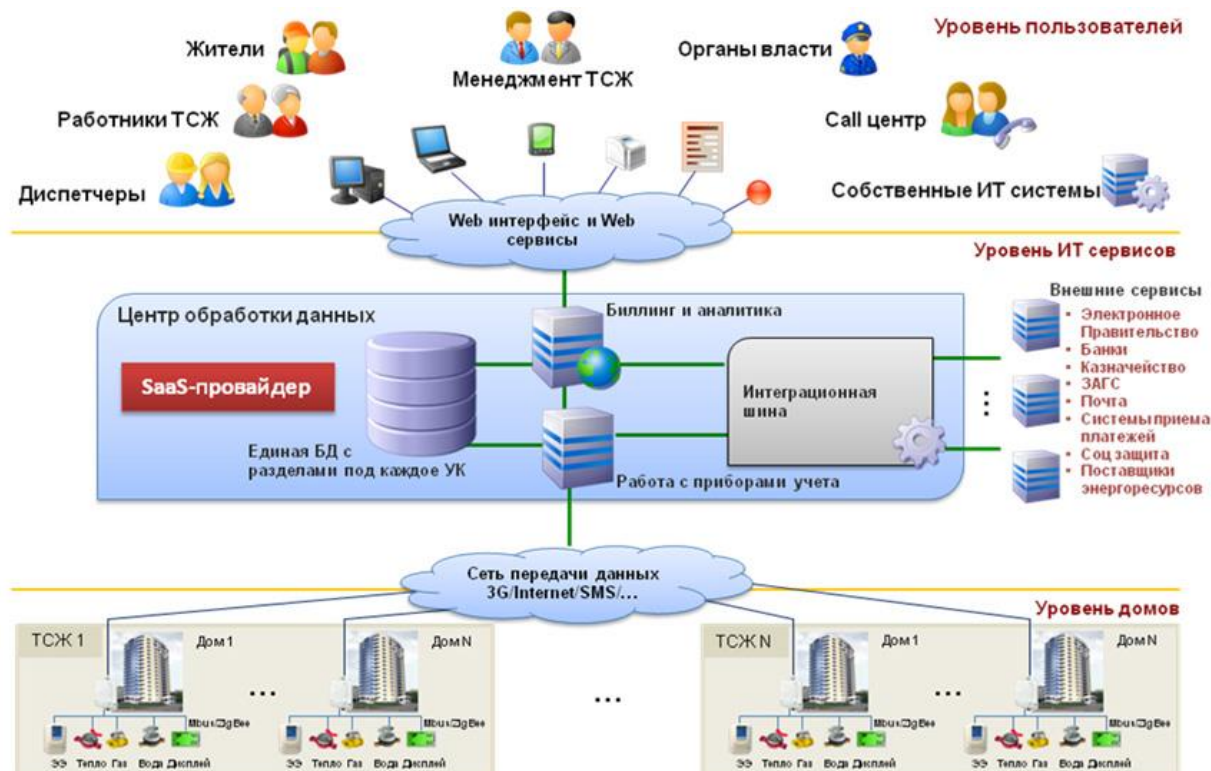
Сервис

История SaaS

Первые компьютеры стоили огромных денег и их вычислительные мощности сдавались заказчикам. С другой стороны, такую аренду нельзя считать разновидностью SaaS, поскольку заказчики получали доступ к компьютерам напрямую, а не с помощью глобальных сетей связи.

Поскольку модель SaaS ориентирована на предоставление услуг с помощью Интернета, её развитие непосредственно связано с развитием глобальной сети. Первые компании, предлагавшие программное обеспечение как услугу, появились в западных странах в 1997—1999 годах, а акроним SaaS вошел в широкое употребление в 2001 году.

Модель SaaS (Software-as-a-Service) предполагает использование различных приложений в аренду. Суть в том, что заказчик не покупает программное обеспечение, а платит за тот сервис, который оно обеспечивает. Заказчик не размещает софт на своих серверах, а пользуется мощностями разработчика. И платить ему приходится только за сервис, обслуживание сервера и консультации. В итоге заказчик получает доступ к необходимому приложению на сервере разработчика, что позволяет ему сэкономить средства и быстро внедрить программу



Преимущества SaaS

Во-первых, ему не нужно покупать само программное обеспечение, а во-вторых, аппаратную платформу для его размещения. Не стоит забывать и о том, что после внедрения системы на предприятие клиенту придется обеспечивать его дальнейшую работоспособность. Если же весь этот фронт работ отдать на откуп разработчику, то заказчику останется лишь пользоваться функционалом программы и вносить регулярные платежи за доступ к системе и сопутствующие услуги.

Второе отличие: в SaaS заказчик получает не само программное обеспечение, а лишь тот функционал, которое оно обеспечивает (в качестве веб-сервиса). А ведь именно реализация бизнес-функций и нужна заказчику.

Третье: в SaaS процедура внедрения сведена к минимуму и очень проста – заказчику нужно просто получить логин/пароль от программы и войти в нее. SaaS-системы не нуждаются в продолжительной настройке, ювелирно тонкой адаптации к требованиям заказчика и дорогостоящих консалтинговых услугах. В результате же сокращается время выполнения проекта и все затраты, связанные с ним.

В-четвертых, SaaS-модель обеспечивает повсеместный доступ к нужному приложению из любого места, где есть интернет. Большинство SaaS-провайдеров обязуются предоставлять практически постоянный доступ к сервису.

В-пятых, SaaS-модель позволяет малому и среднему бизнесу использовать приложения, которые ранее были недоступны из-за дороговизны. Вместо покупки программы заказчик за небольшую плату арендует бизнес-функции, которые она реализует.

В модели SaaS:

- приложение приспособлено для удаленного использования;
- одним приложением пользуется несколько клиентов (приложение коммунально);
- оплата взимается либо в виде ежемесячной абонентской платы, либо на основе объема операций;
- техническая поддержка приложения включена в оплату;
- модернизация и обновление приложения происходит оперативно и прозрачно для клиентов.

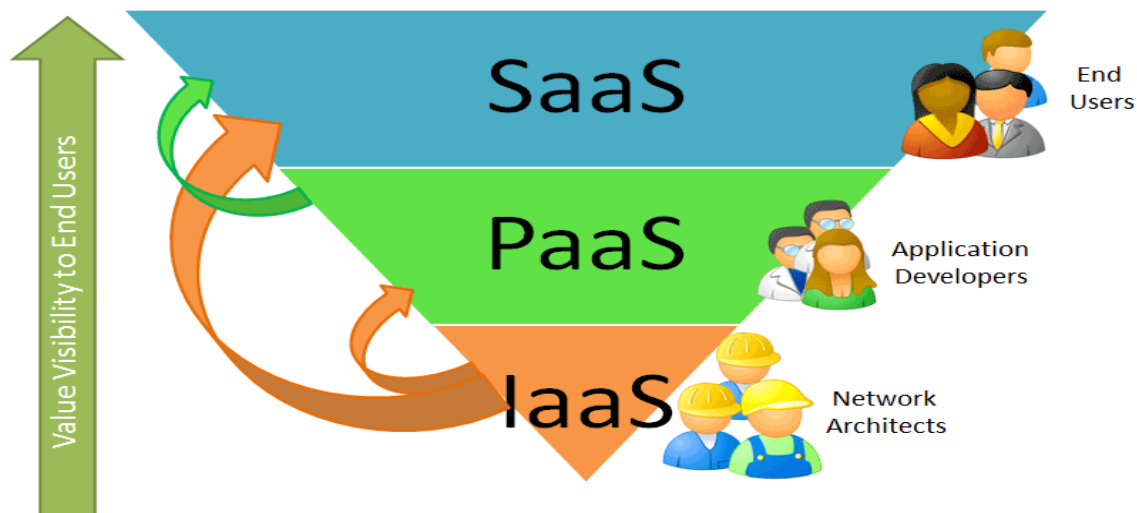
Положительные факторы SaaS для заказчиков

- Отсутствие необходимости установки ПО на рабочих местах пользователей — доступ к ПО осуществляется через обычный браузер;
- Радикальное сокращение затрат на развёртывание системы в организации. Это расходы на аренду помещения, организацию дата-центра, оплату труда сотрудников и т. д.;
- Сокращение затрат на техническую поддержку и обновление развернутых систем (вплоть до их полного отсутствия);
- Быстрота внедрения, обусловленная отсутствием затрат времени на развертывание системы;
- Понятный интерфейс — большинство сотрудников уже привыкли к использованию веб-сервисов;
- Ясность и предсказуемость платежей, защита инвестиций;
- Мультиплатформенность;
- Возможность получить более высокий уровень обслуживания ПО.

Положительные факторы SaaS для разработчиков

- Рост популярности веб-сервисов для конечных пользователей;
- Развитие веб-технологий, большие функциональные возможности веб-приложений и простота их реализации;
- Быстрые процессы внедрения и сравнительно низкие затраты ресурсов на обслуживание конкретного клиента;

- Лёгкое проникновение на глобальные рынки;
- Отсутствие проблем с нелегальным распространением ПО;
- В отличие от классической модели, заказчик SaaS привязывается к разработчику — он не может отказаться от услуг разработчика и продолжать использовать систему. Таким образом, обеспечивается защита инвестиций разработчика в процесс продаж;
- В долгосрочном периоде доходы от SaaS могут оказаться выше прибыли, полученной от продажи лицензий и оказания технической поддержки (даже с учётом расходов на хостинг и управление приложениями).



IaaS – инфраструктура как сервис. Аренда оборудования, оплата за конкретные аппаратные средства)

PaaS – платформа как сервис. Аренда программной платформы (среда разработки + исполнения). Конечно же, это автоматически подразумевает и использование железа, но в неявном виде для пользователя. Оплачивается не железо (вы вообще не знаете какое оборудование применяется), а ресурсы в терминах софта, причем, возможно, в виде, например, числа транзакций, а не ресурсов памяти,

SaaS - софт как сервис. Тут присутствует все – и аппаратура, и программная платформа, и прикладной софт. Но оплата ведется в терминах решения бизнес-задач (функционал, число пользователей).

Примеры SaaS:

1. Google Docs
2. Google Calendar
3. Gmail
4. Google Talk
5. Talk Fusion

Список литературы:

- 1) <http://ru.wikipedia.org/wiki/SaaS>
- 2) <http://12news.ru/doc8974.html>
- 3) <http://www.pcweek.ru/its/blog/its/1400.php>
- 4) http://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82:Google_Apps_Premier_Edition