

IT – интеграция Современный подход в банках



РосЕвроБанк

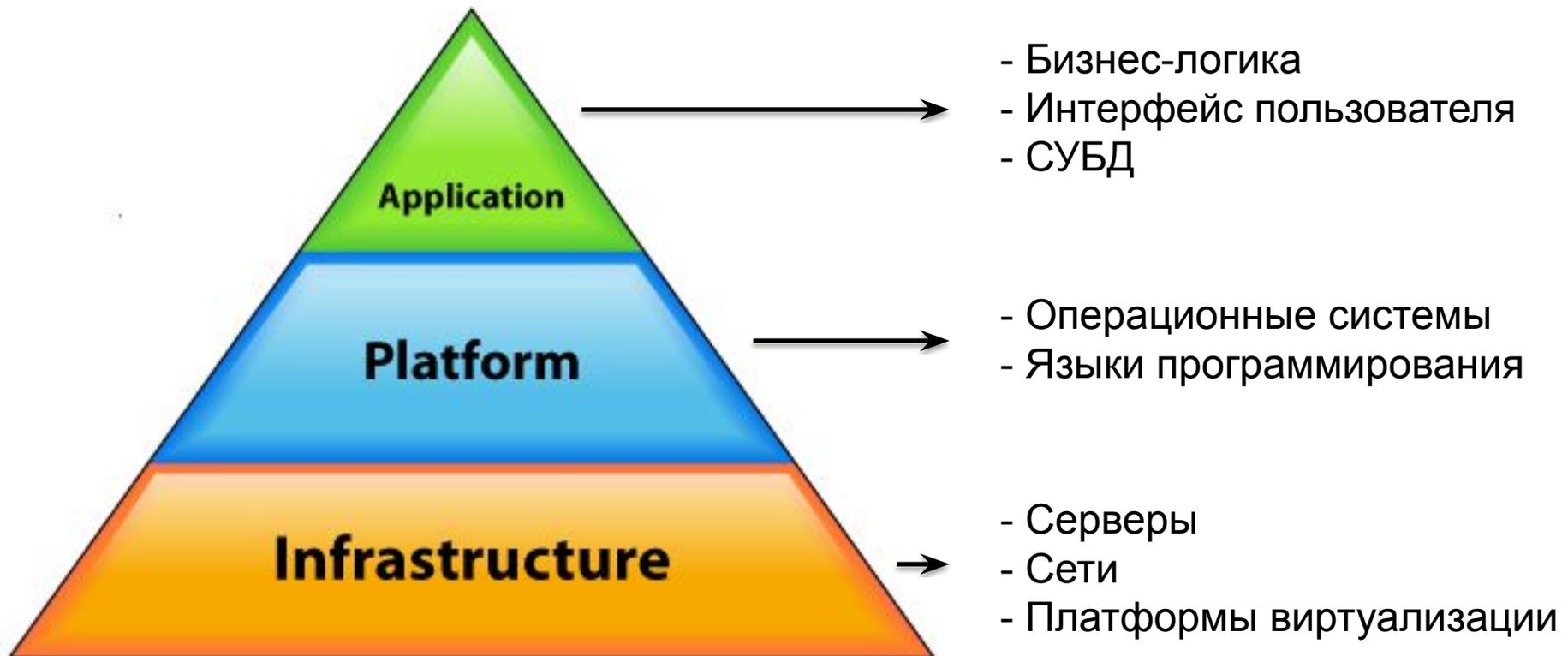
Автор: Бухматов Павел

Интеграция – объединение частей в целое

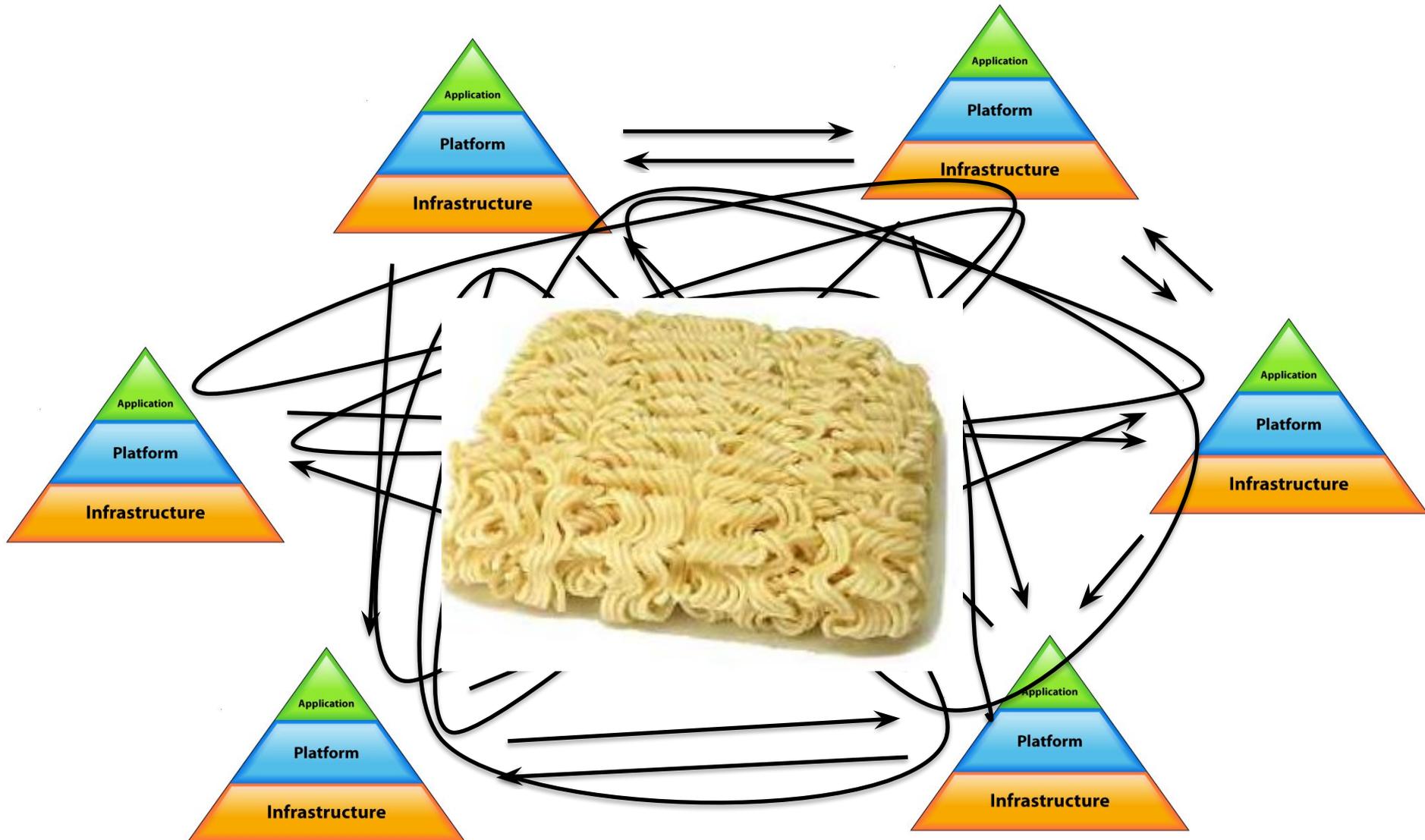


Каких частей?

Информационных систем!



Наивный подход к Интеграции



Нужна ли системная интеграция?

Одна система

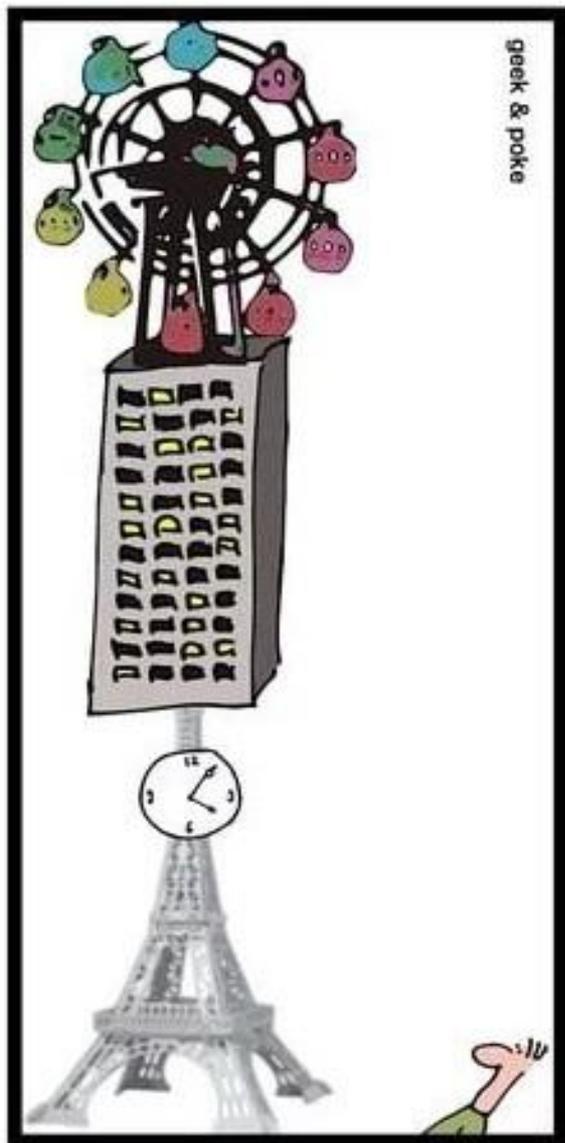


VS

**Много систем и
интеграционный слой**



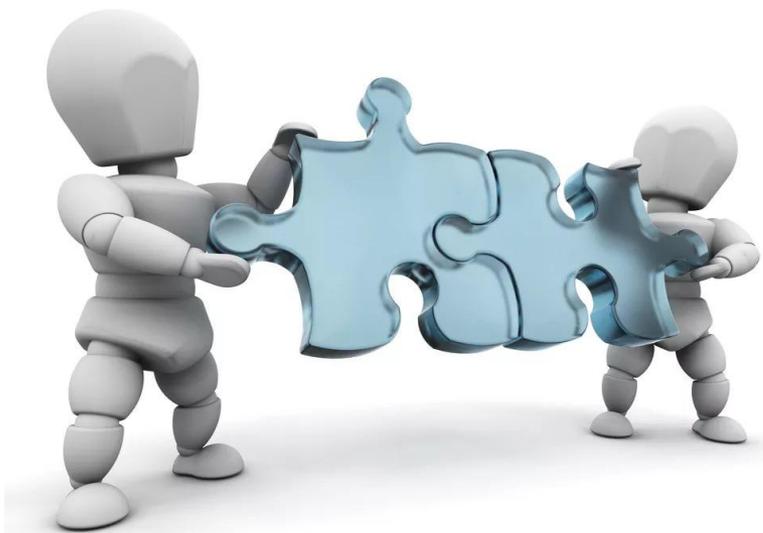
Одна система



- + Легко настроить мониторинг и отладку
- + Просто разворачивать
- +/- Весь код в едином контексте
- /+ Сложно обновлять**
- Ошибки имеют глобальный характер**
- Тяжело масштабируется**
- Ошибки тяжело отследить**

Много систем и слой интеграции

- + Ошибки в системах имеют локальный характер
- + Легче масштабируется
- + Легче разворачивается
- **Может отвалиться весь интеграционный слой (?)**
- **Мониторинг — сложная задача**
- **Ошибки все еще тяжело отследить...**



Интеграция...?



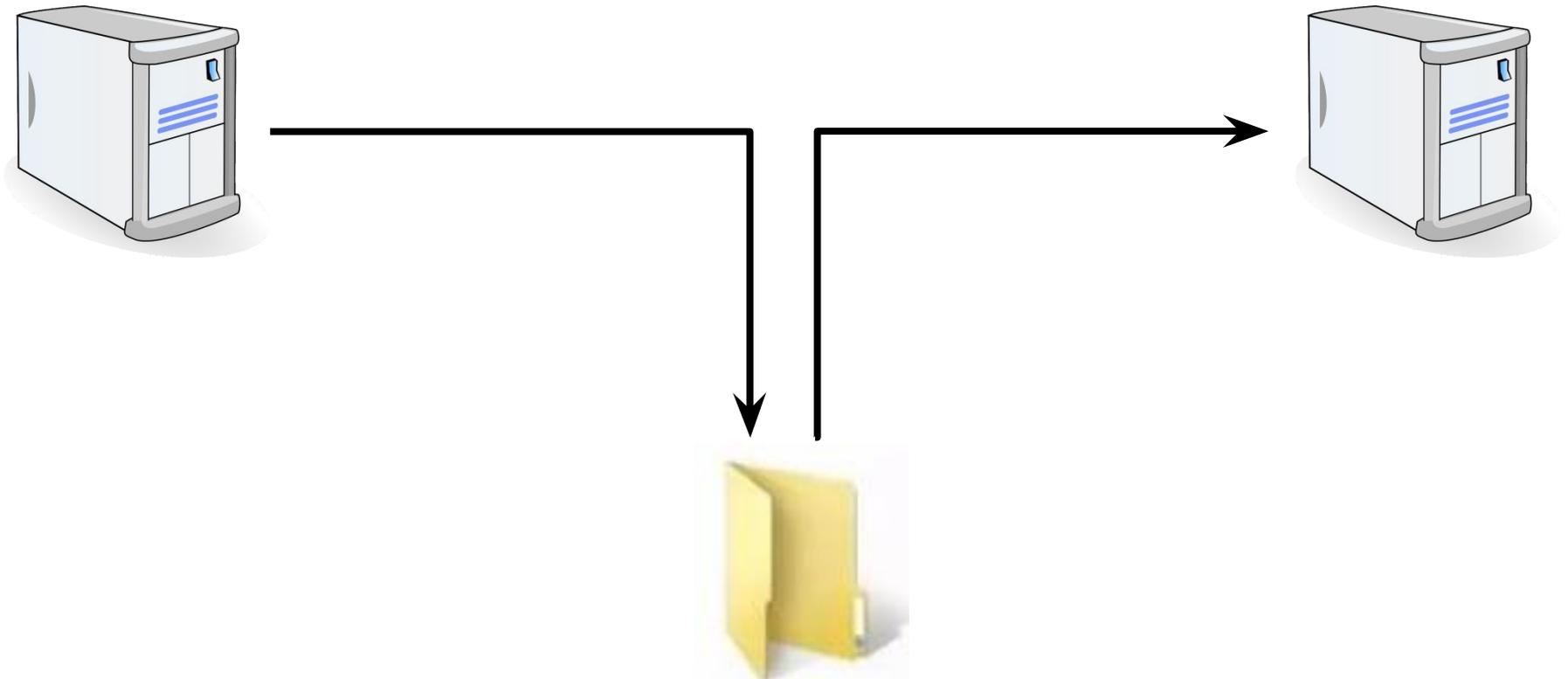
Основные подходы к интеграции

- Файловый обмен

Принцип работы файлового обмена

```
write("file.txt",  
"/root/user/docs/")
```

```
read("/root/user/docs/file.  
txt")
```



Пример из жизни (Файловый обмен)

```
foo@dev:~/ESBData> ll
```

```
итого 120
```

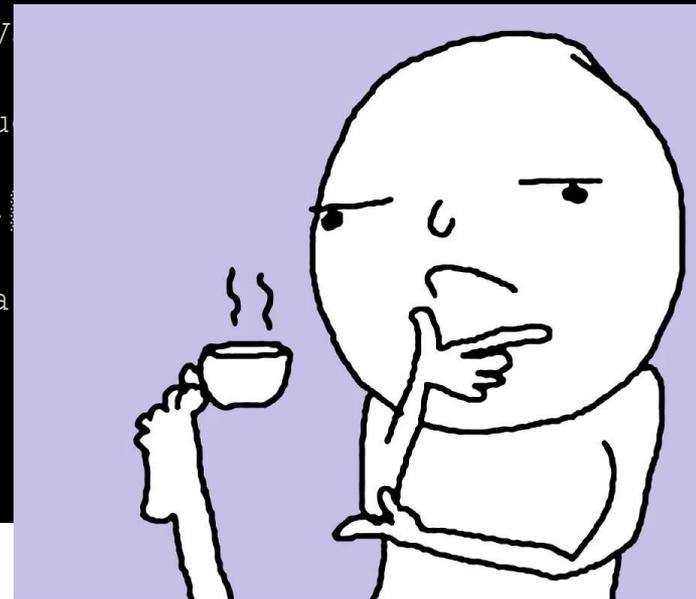
```
-rw-r--r-- 1 foo users 223 Окт 13 10:20 ats_2017.10.13_10.20.58.399.+0300.xml
-rw-r--r-- 1 foo users 131 Окт 23 12:24 ats_2017.10.23_12.24.35.144.+0300.xml
-rw-r--r-- 1 foo users 917 Окт 23 12:46 ats_2017.10.23_12.46.00.418.+0300.xml
-rw-r--r-- 1 foo users 297 Окт 23 14:45 ats_2017.10.23_14.45.00.837.+0300.xml
-rw-r--r-- 1 foo users 317 Окт 23 14:45 ats_2017.10.23_14.45.36.923.+0300.xml
-rw-r--r-- 1 foo users 280 Окт 23 14:46 ats_2017.10.23_14.46.05.834.+0300.xml
-rw-r--r-- 1 foo users 234 Окт 23 14:46 ats_2017.10.23_14.46.55.233.+0300.xml
-rw-r--r-- 1 foo users 742 Окт 27 11:43 ats_2017.10.27_11.43.09.981.+0300.xml
-rw-r--r-- 1 foo users 1041 Окт 26 09:18 dep_2017.10.26_09.18.52.104.+0300.xml
-rw-r--r-- 1 foo users 1041 Окт 26 15:55 dep_2017.10.26_15.55.52.874.+0300.xml
-rw-r--r-- 1 foo users 1041 Окт 26 16:01 dep_2017.10.26_16.01.58.526.+0300.xml
-rw-r--r-- 1 foo users 1229 Окт 31 13:50 emp_2017.10.31_13.50.02.849.+0300.xml
-rw-r--r-- 1 foo users 1229 Окт 31 13:50 emp_2017.10.31_13.50.21.729.+0300.xml
-rw-r--r-- 1 foo users 10520 Ноя 1 16:36 lda_2017.11.01_16.36.47.582.+0300.xml
-rw-r--r-- 1 foo users 10520 Ноя 2 11:35 lda_2017.11.02_11.35.25.477.+0300.xml
-rw-r--r-- 1 foo users 10520 Ноя 2 11:44 lda_2017.11.02_11.44.12.191.+0300.xml
```

Пример из жизни (Файловый обмен)

```
foo@dev:~/ESBData> cat ats_2017_10_31_09_32_08.542.+0300.xml
<?xml version="1.0" encoding="windows-1251" standalone="yes"?>
<ROOT Version="2.0">
```

encoding="windows-1251"

```
  <RecordRef Vid="██████" Kod="0">
    <Requisite Name="██████" Type="String" Value="00"/>
    <Requisite Name="██████" Type="String" Value="██████" />
    <Requisite Name="██████" Type="String" Value="██████" />
    <Requisite Name="██████" Type="String" Value="██████" />
    <Requisite Name="██████" Type="String" Value="107078, ██████ 2/1, ██████ 24"/>
    <Requisite Name="██████" Type="DateTime" Value="██████" />
  </RecordRef>
</ROOT>esbuser@esb8-dev:~/ESBData/DIRECTUM>
foo@dev:~/ESBData>
```



Файловый обмен

+ Работает
везде

+ Лучше
защищен от
потерь

+ Слабая
связанность

– Сложный контракт
взаимодействия

– Кодировки

– Сложно получать
актуальные данные

– Нужен механизм
переноса файлов
между жесткими
дисками

Основные подходы к интеграции

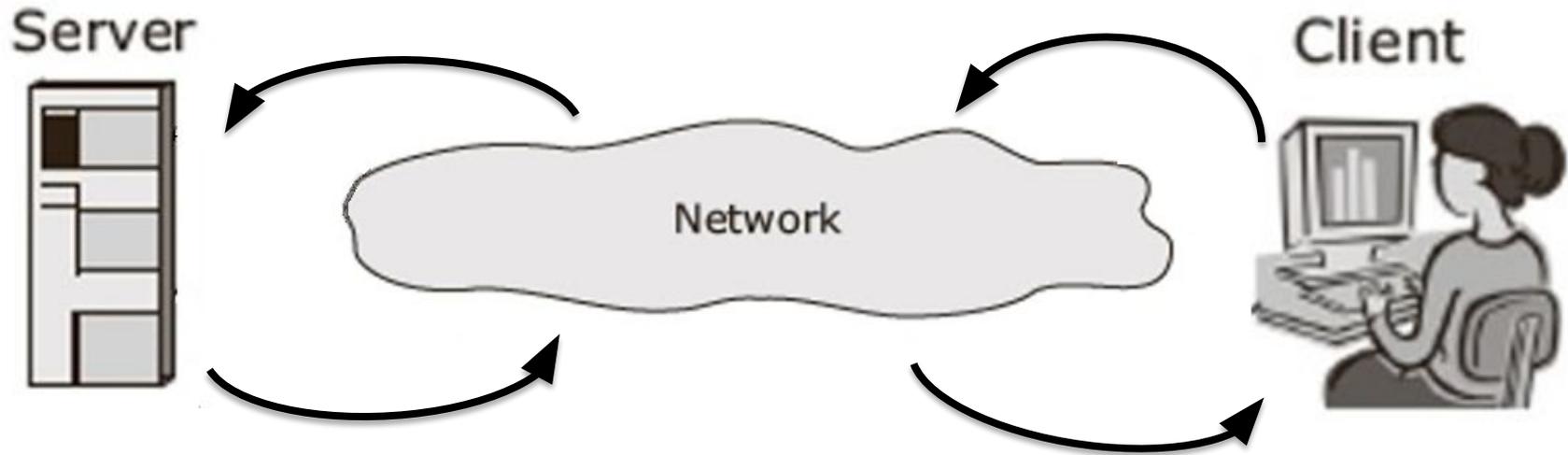
- ✓ Файловый обмен
- RPC (Remote Procedure Call)

Притворимся, что удаленный
вызов на самом деле
локальный

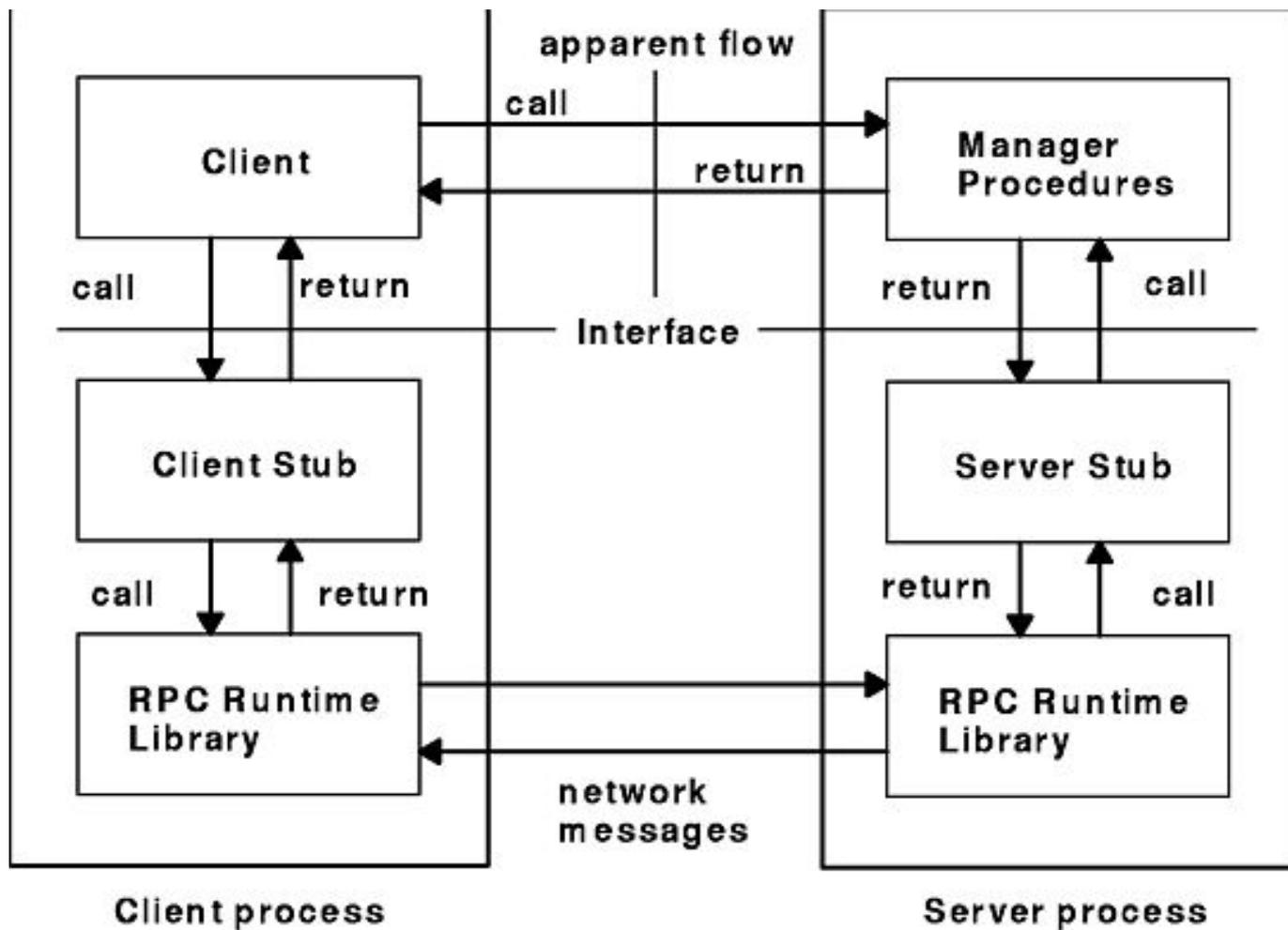
Принцип работы RPC

```
int add(int foo, int bar) {  
    return foo + bar;  
}
```

```
<..>  
int a =  
add(5, 6)  
<..>
```



Принцип работы RPC на самом деле



Remote Procedure Call Flow

RPC

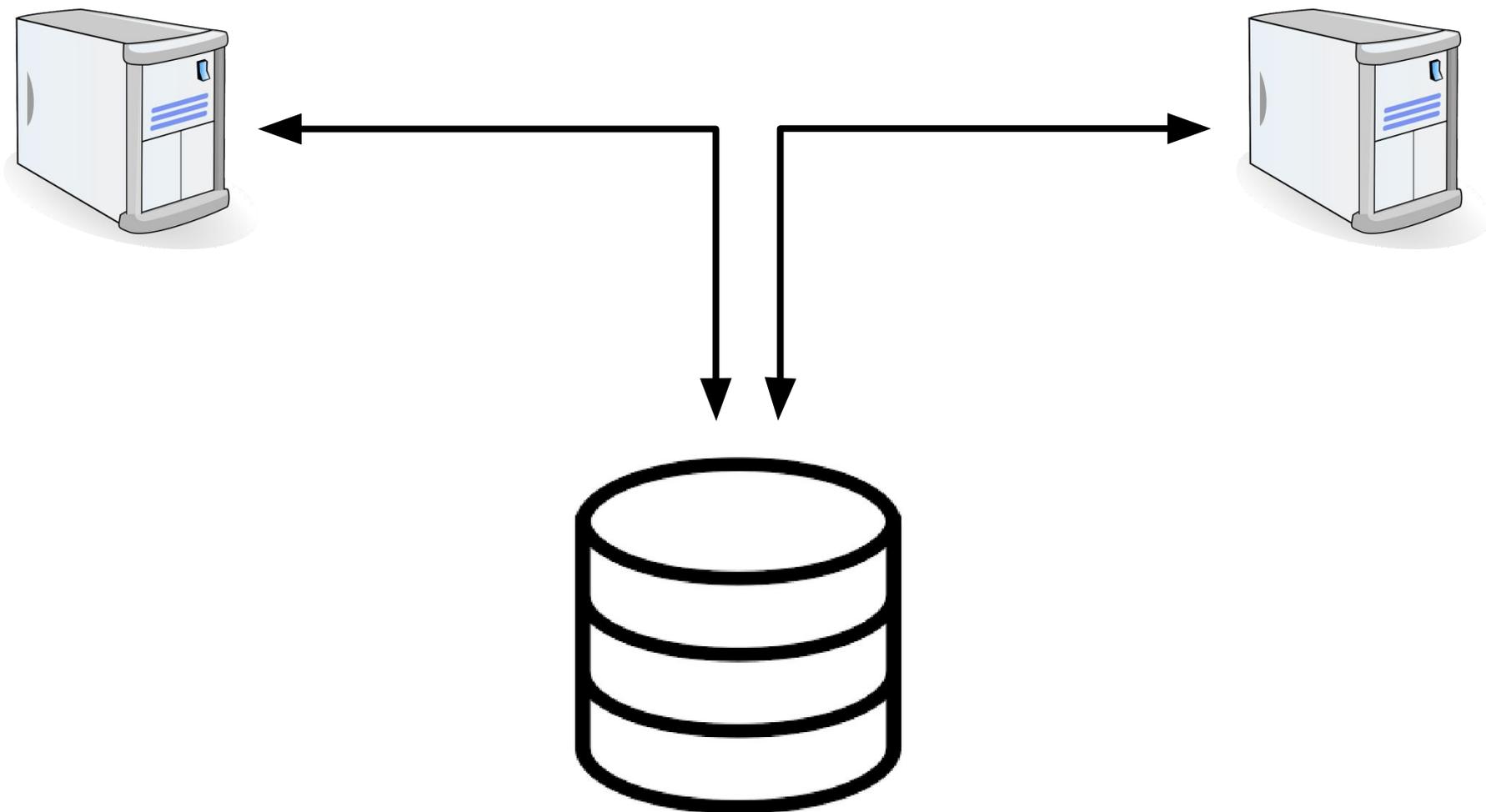
+ Простое
использование
+ Можно
притвориться,
что все
локально

– Сложная
конфигурация
– Нетривиальная
архитектура
– Сильное
связывание систем
– Проблемы с
передачей большого
объема данных

Основные подходы к интеграции

- ✓ Файловый обмен
- ✓ RPC (Remote Procedure Call)
- Общая база данных

Принцип работы общей БД



Общая БД

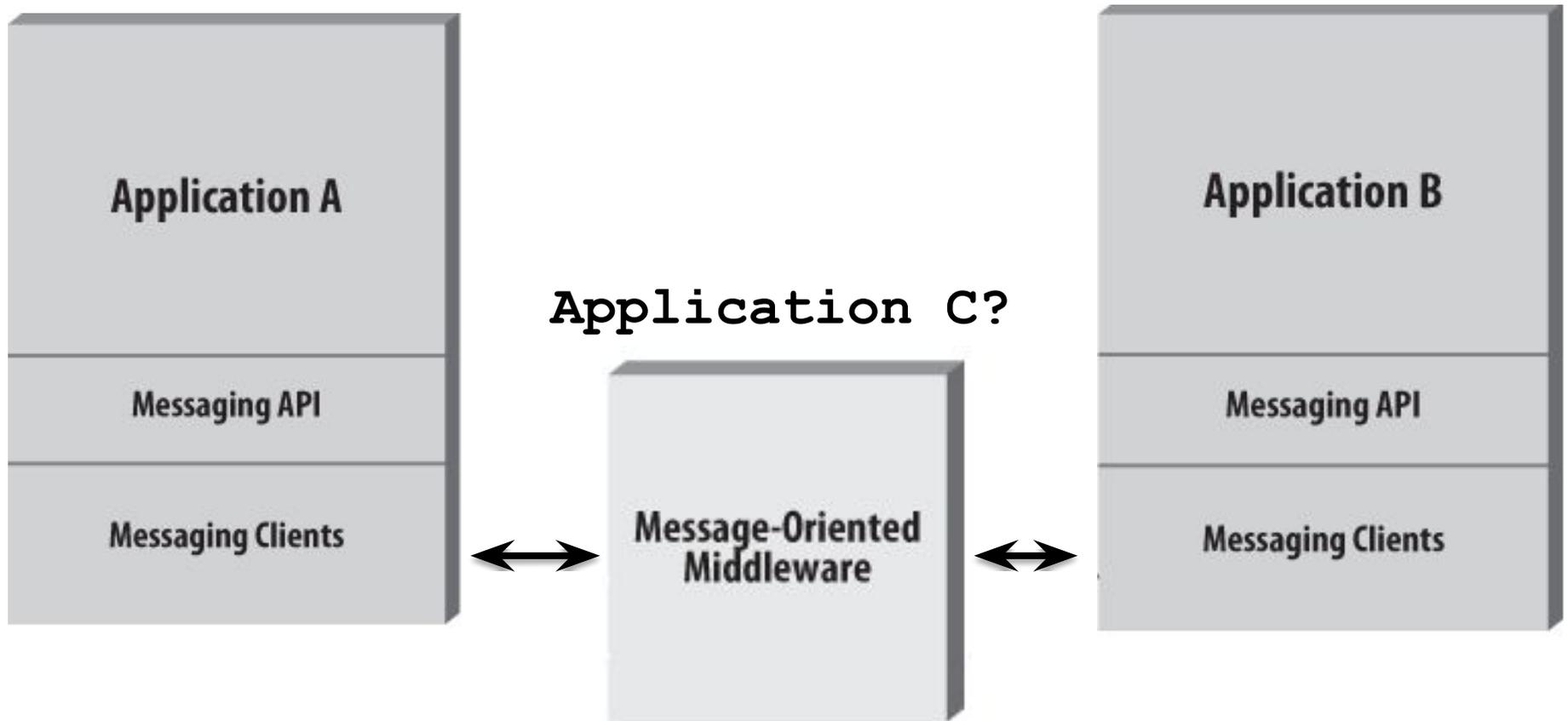
- + Транзакционность
- + Персистентность
(Сохранение данных при ошибках / катастрофах, etc)
- + Легкое представление иерархий

- Данные в одном формате в нескольких системах
- Изменять базу сложно (Надо менять всех клиентов)
- Общая точка отказа

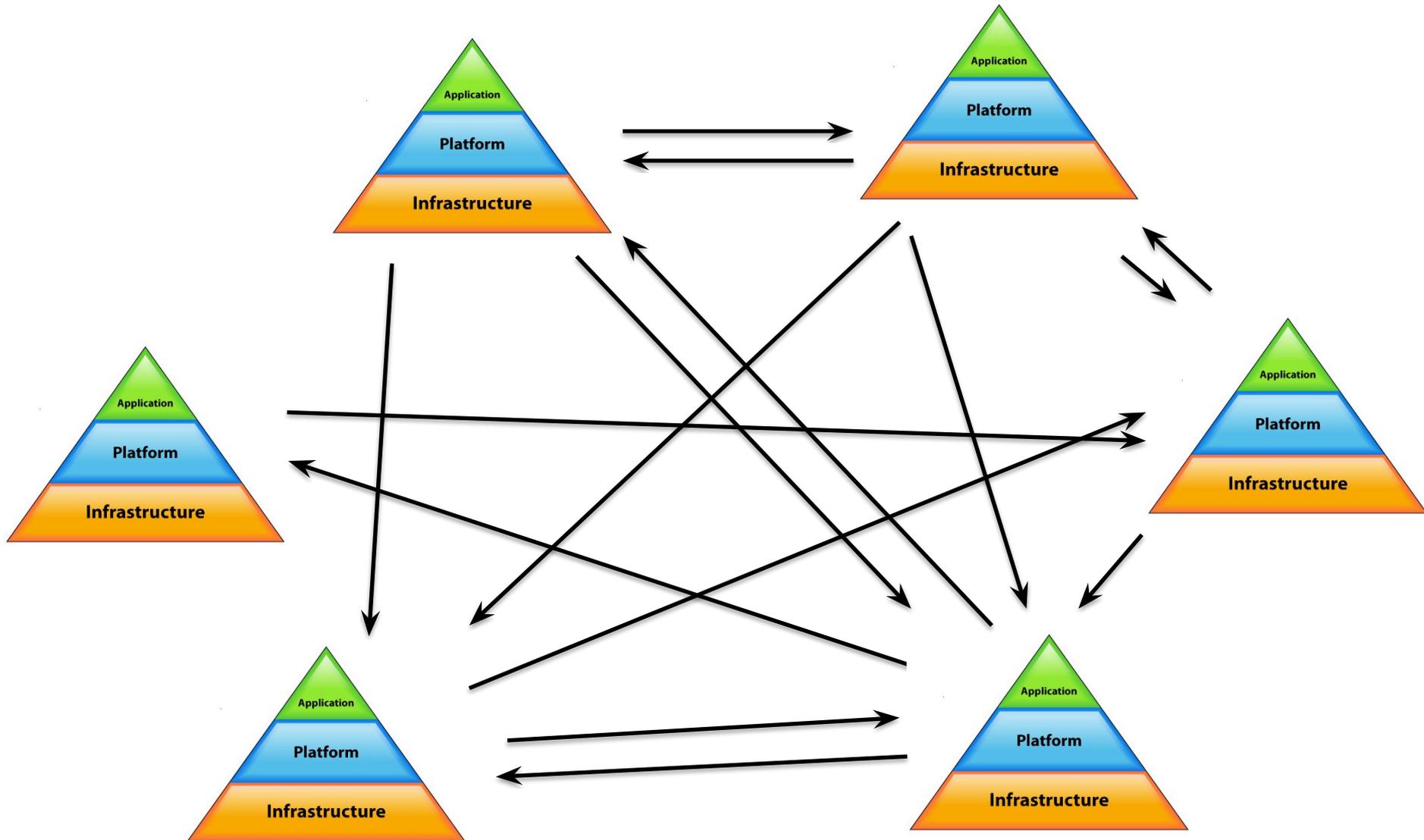
Основные подходы к интеграции

- ✓ Файловый обмен
- ✓ RPC (Remote Procedure Call)
- ✓ Общая база данных
- Системы обмена сообщениями (МОМ – message oriented middleware)

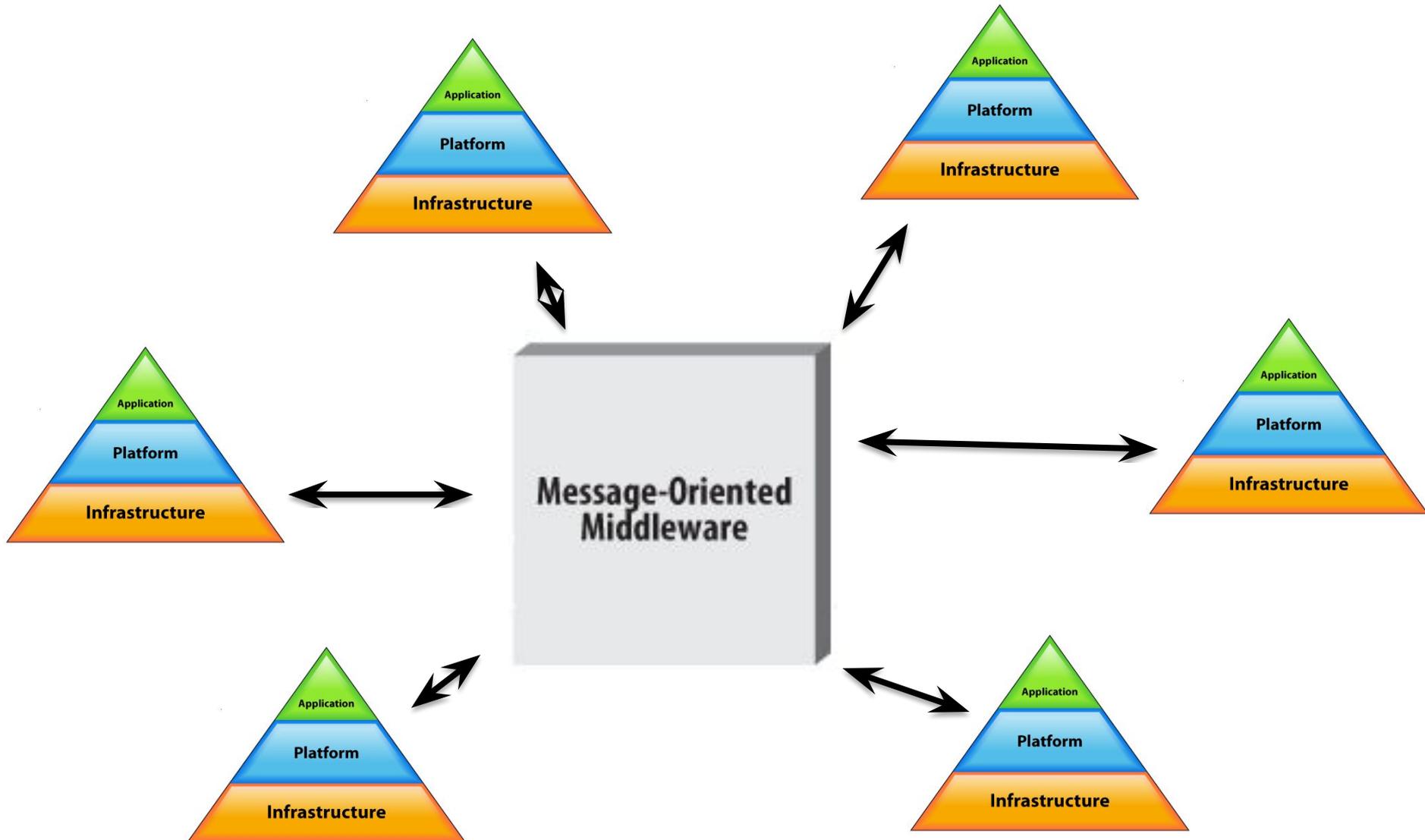
Принцип работы MOM



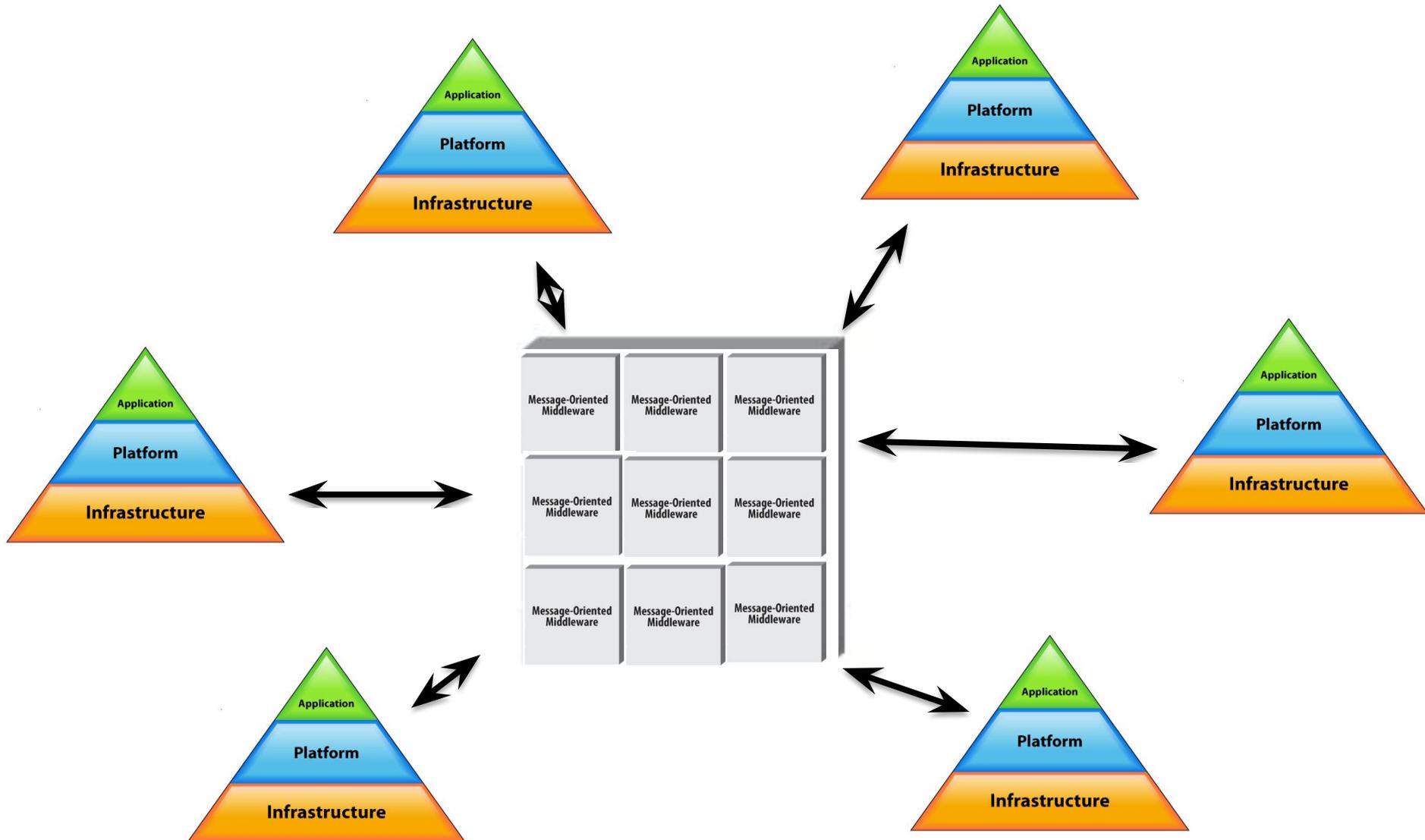
Интеграция



Интеграция



Интеграция



МОМ

- + Слабая связанность
- + Можно модифицировать отдельно от приложений
- + Сложная логика взаимодействия систем реализуется на уровне МОМ

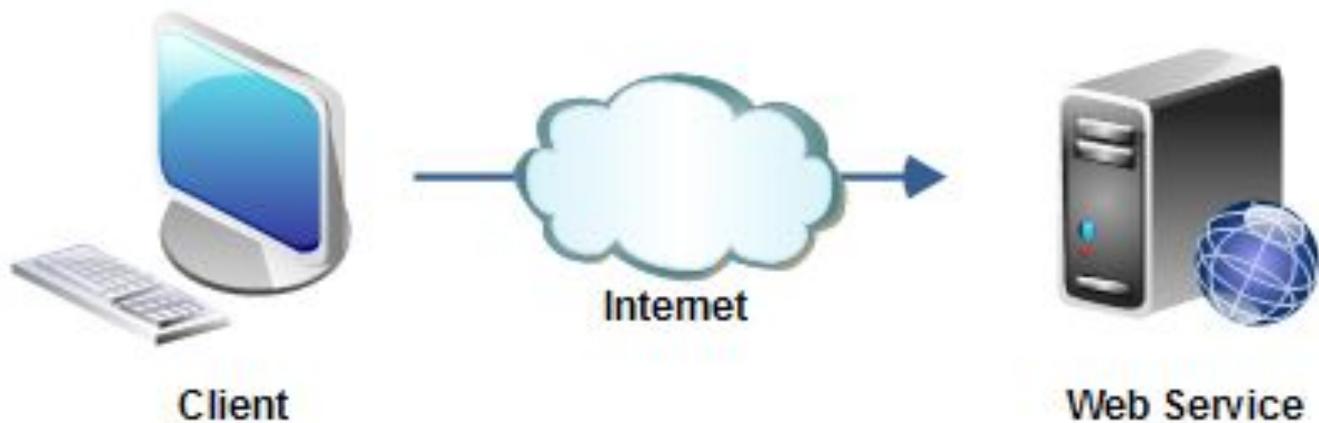
- Дополнительный компонент архитектуры (сама МОМ)
- Сложность синхронного взаимодействия
- Много поставщиков (IBMmq, RabbitMQ, Kafka, etc)

Основные подходы к интеграции

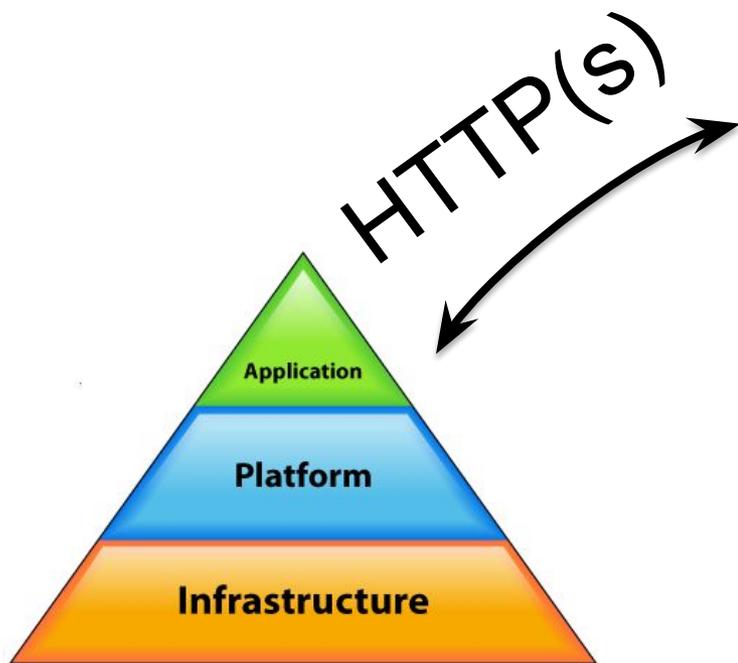
- ✓ Файловый обмен
- ✓ RPC (Remote Procedure Call)
- ✓ Общая база данных
- ✓ Системы обмена сообщениями (МООМ – message oriented middleware)
 - Web сервисы

Принцип работы Web сервисов

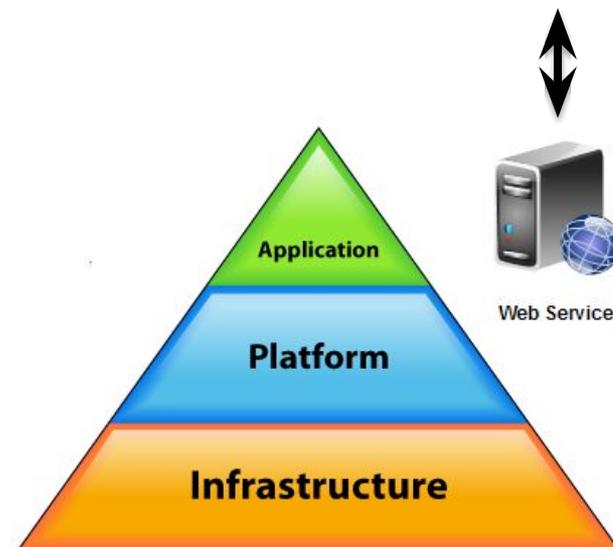
HTTP(S) – ЭТО ЖИЗНЬ



Принцип работы Web сервисов



URL: `http://my_company/dostuff`



Внутренняя сеть

Web сервисы

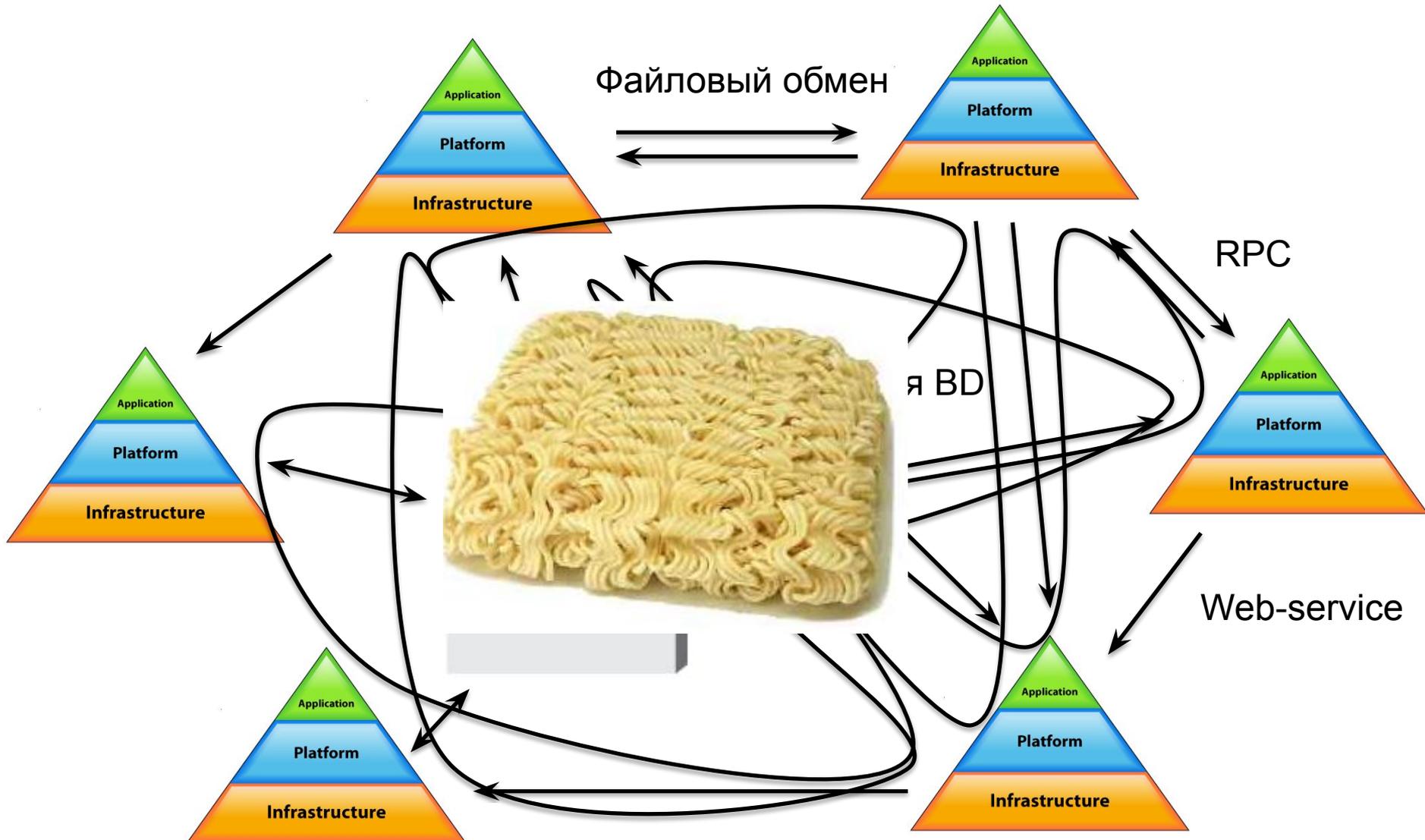
- + Синхронны по своей природе
- + Просты в реализации
- + Является общепринятым стандартом (Реализованы балансировка, авторизация и т.д.)

- Сложности с асинхронным взаимодействием
- +/- Довольно сложный контракт
- Сложно передавать много данных

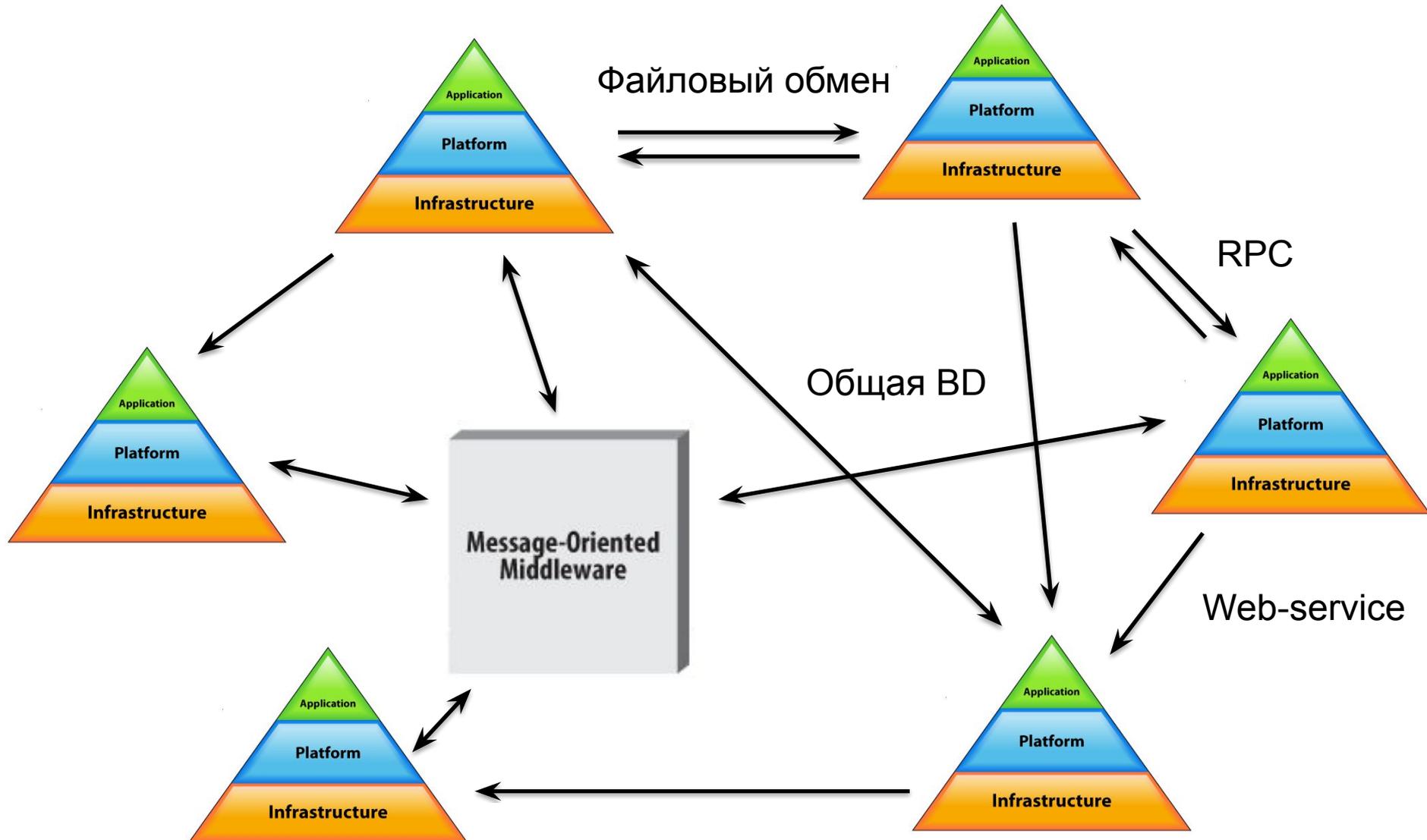
Основные подходы к интеграции

- ✓ Файловый обмен
- ✓ RPC (Remote Procedure Call)
- ✓ Общая база данных
- ✓ Системы обмена сообщениями (МООМ – message oriented middleware)
- ✓ Web сервисы

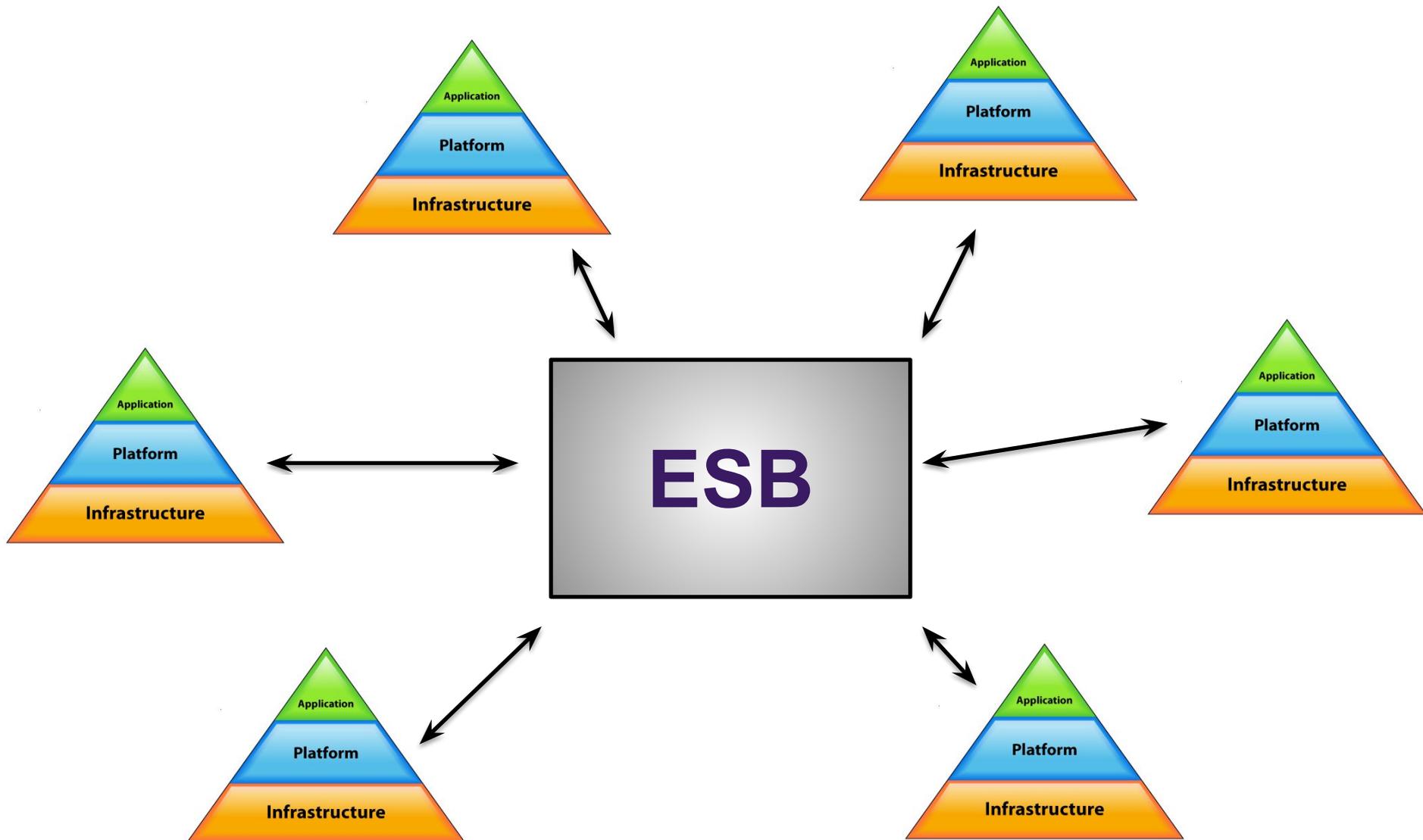
Наивный подход к Интеграции 2



Наивный подход к Интеграции 2



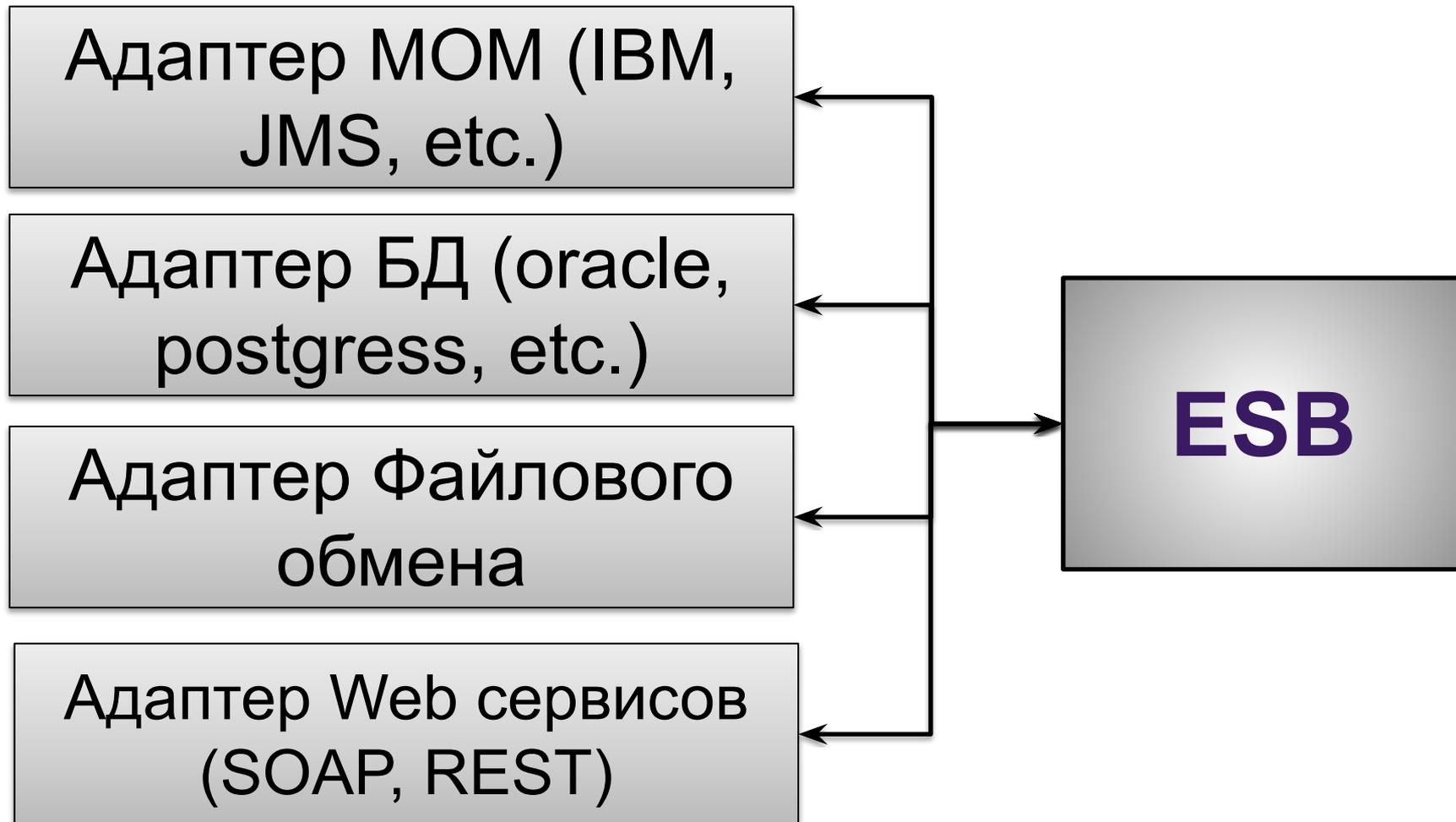
Шина интеграции (ESB)



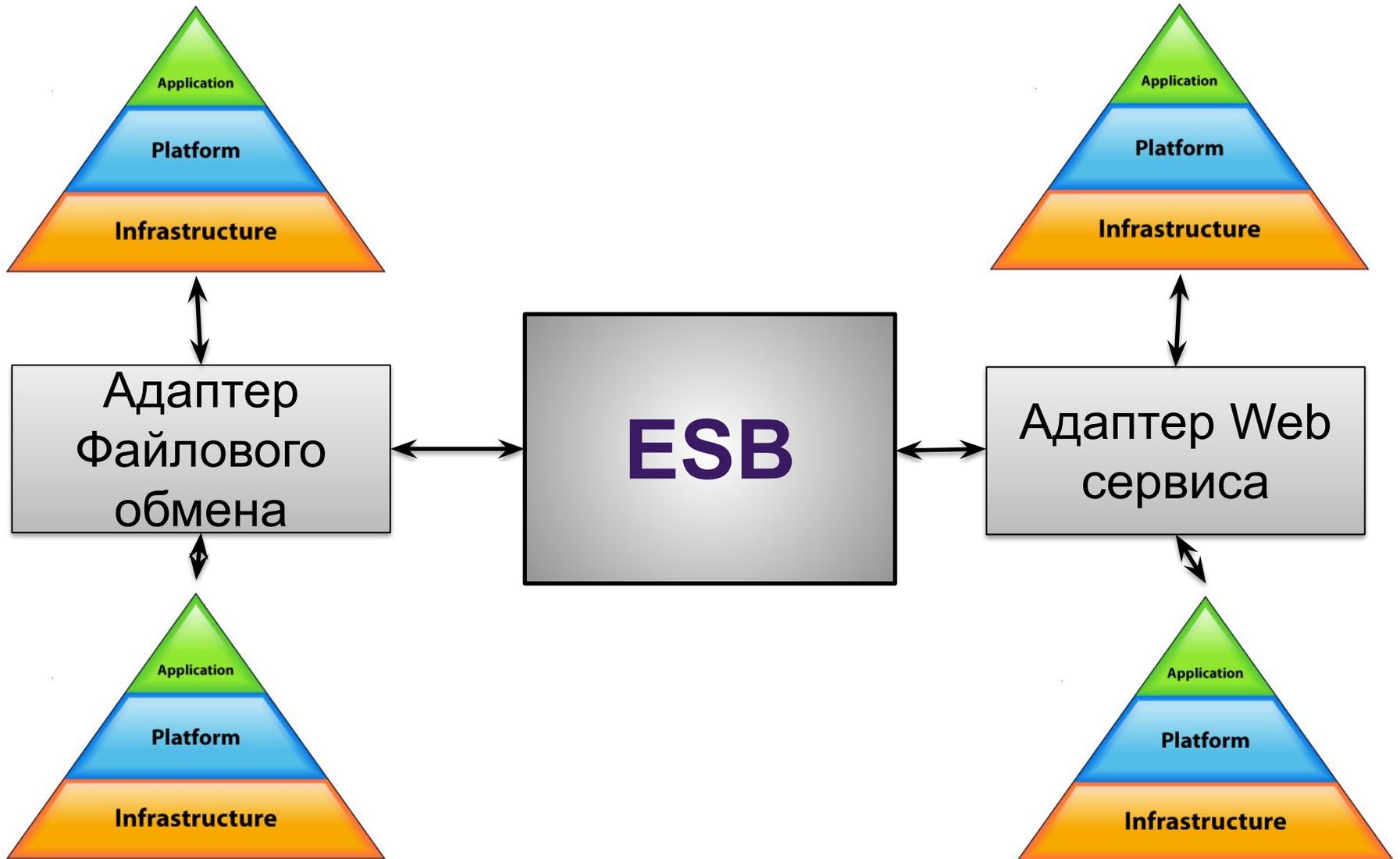
Шина интеграции (ESB)

Промежуточное ПО,
централизирующие задачи
обмена данными между
системами

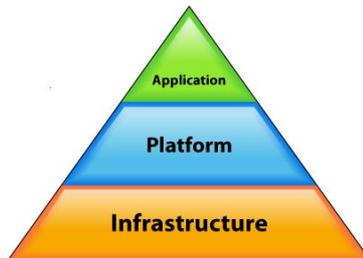
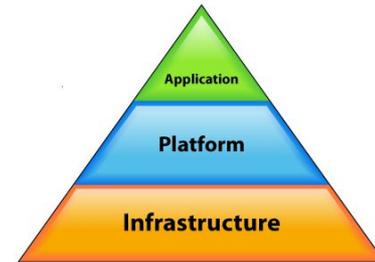
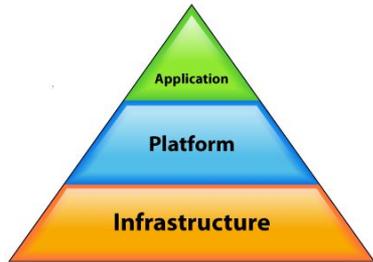
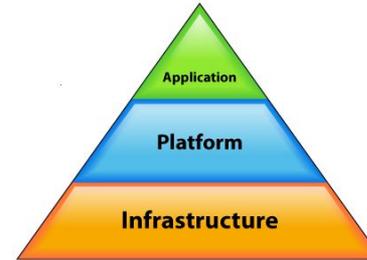
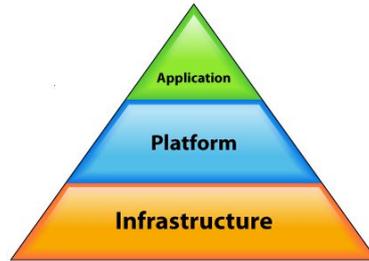
Как «присоединиться» к шине



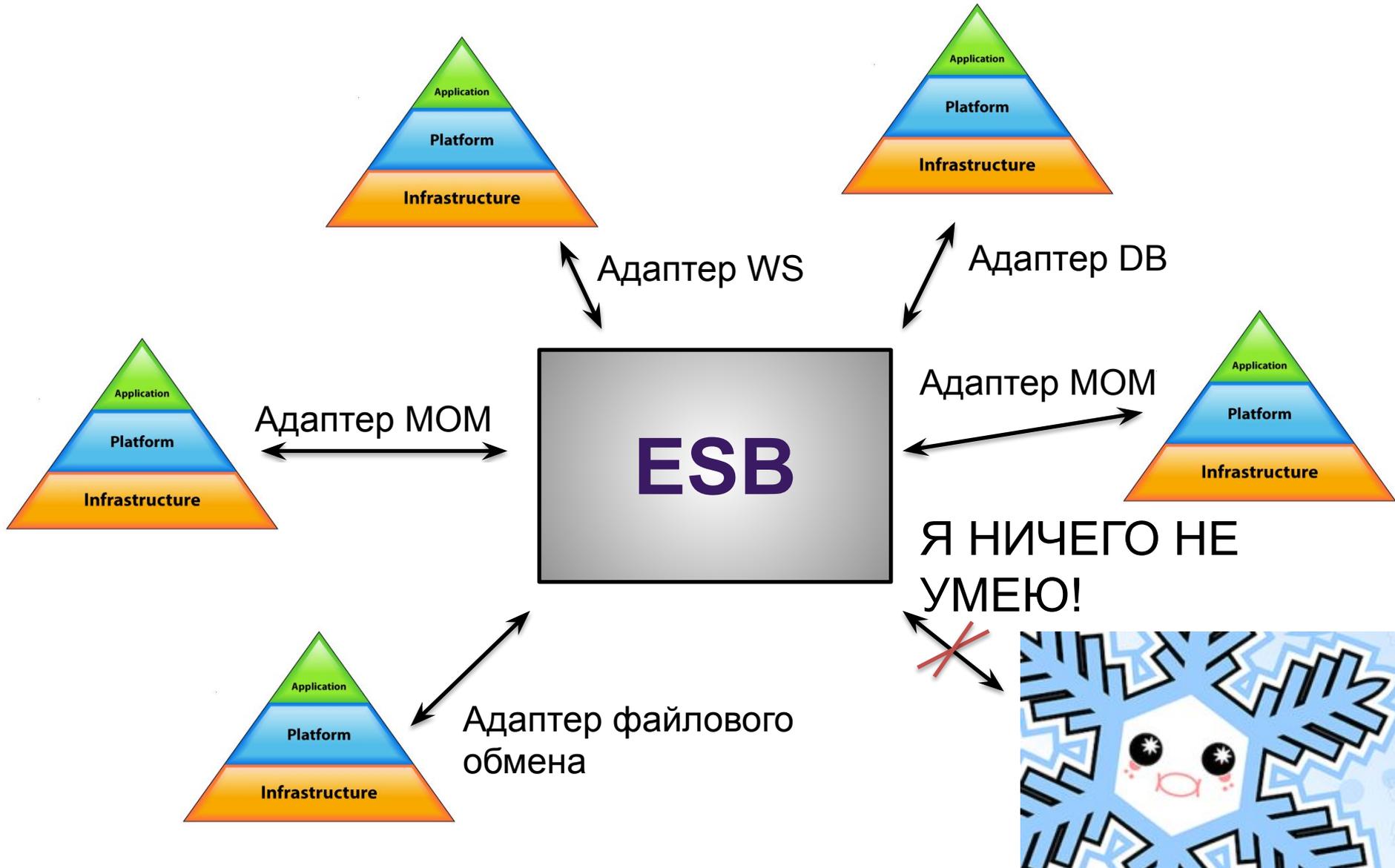
Пример



Иногда этого не хватает...



Иногда этого не хватает...



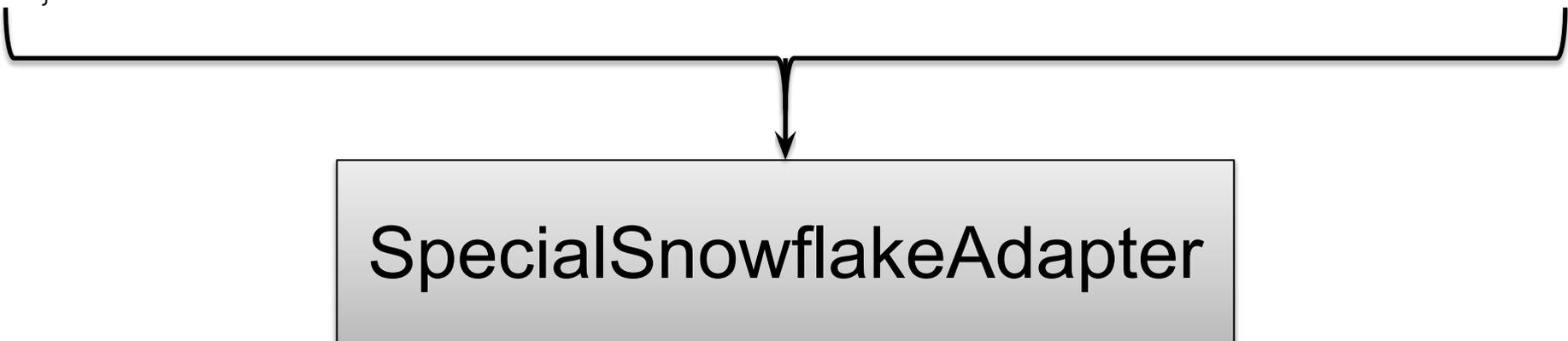
...Иногда появляется система -
особая снежинка, которая
ничего не умеет

И для нее надо делать особый
Адаптер

... При помощи API шины!

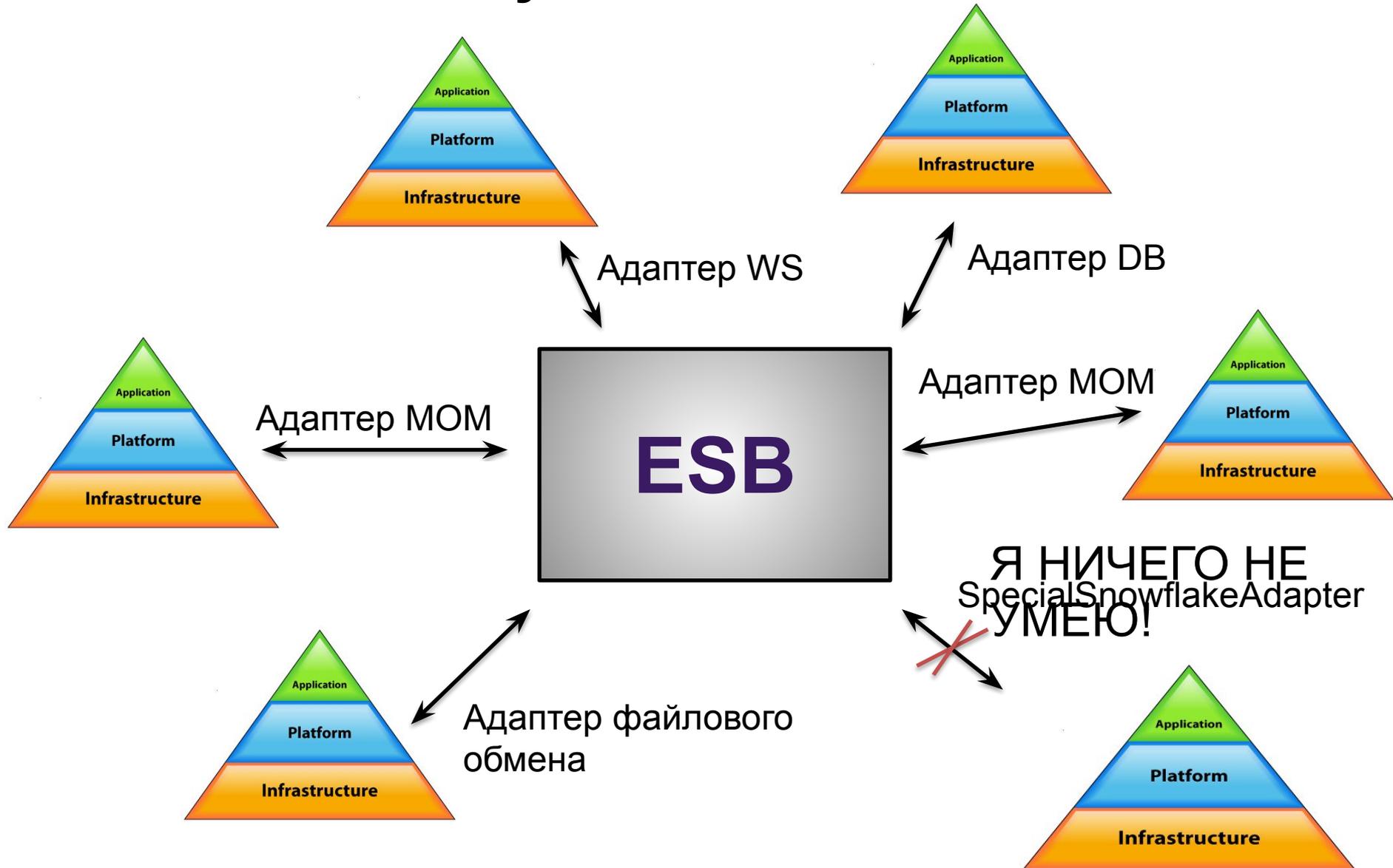
API шины интеграции SonicESB

```
import com.sonicsw.xq.XQServiceEx;  
import com.sonicsw.xq.XQInitContext;  
  
class SpecialSnowflakeAdapter implements XQServiceEx {  
  
    @Override public void init(XQInitContext initialContext)  
    @Override public void service(XQServiceContext ctx)  
    @Override public void start()  
    @Override public void stop()  
    @Override public void destroy()  
  
}
```

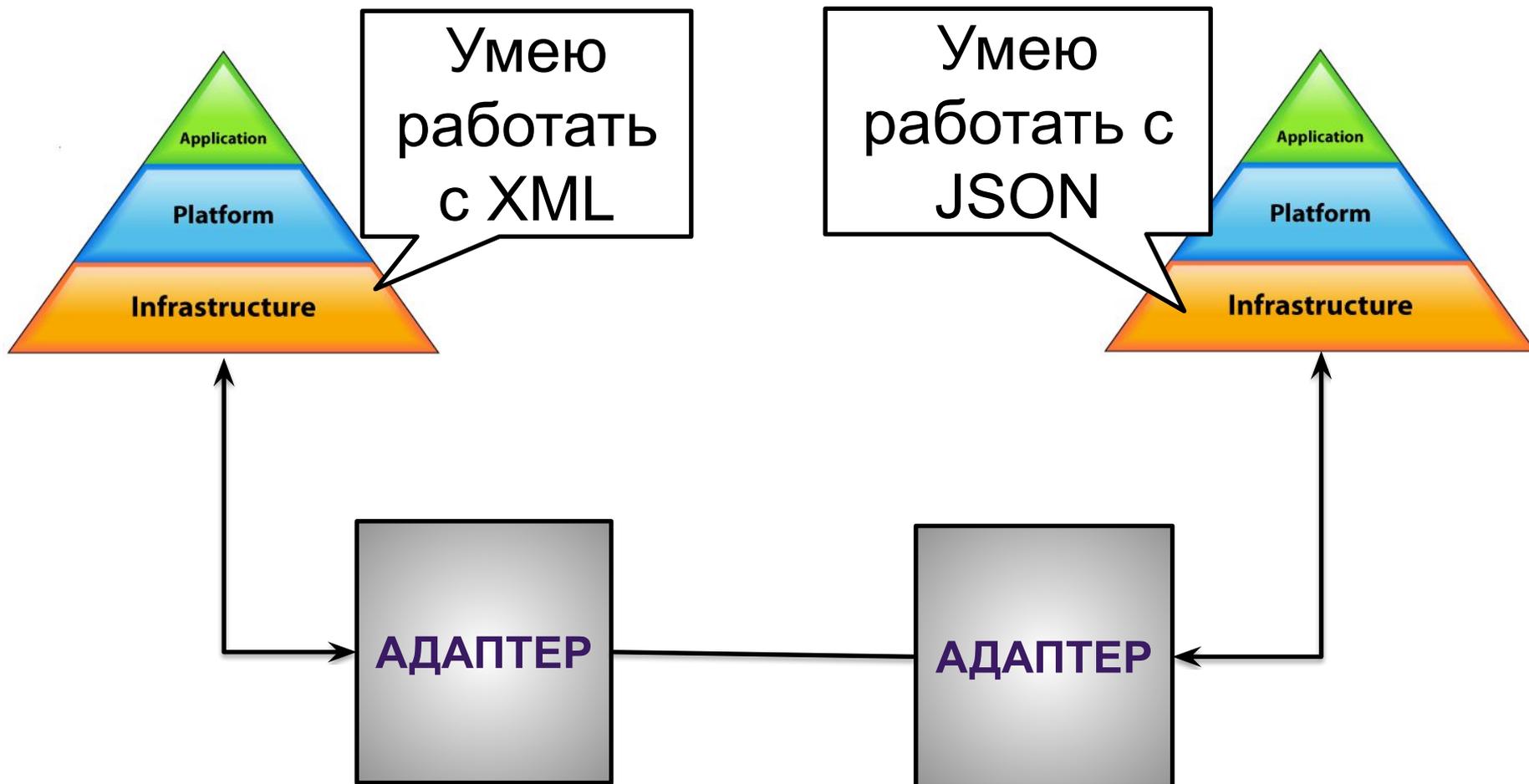


SpecialSnowflakeAdapter

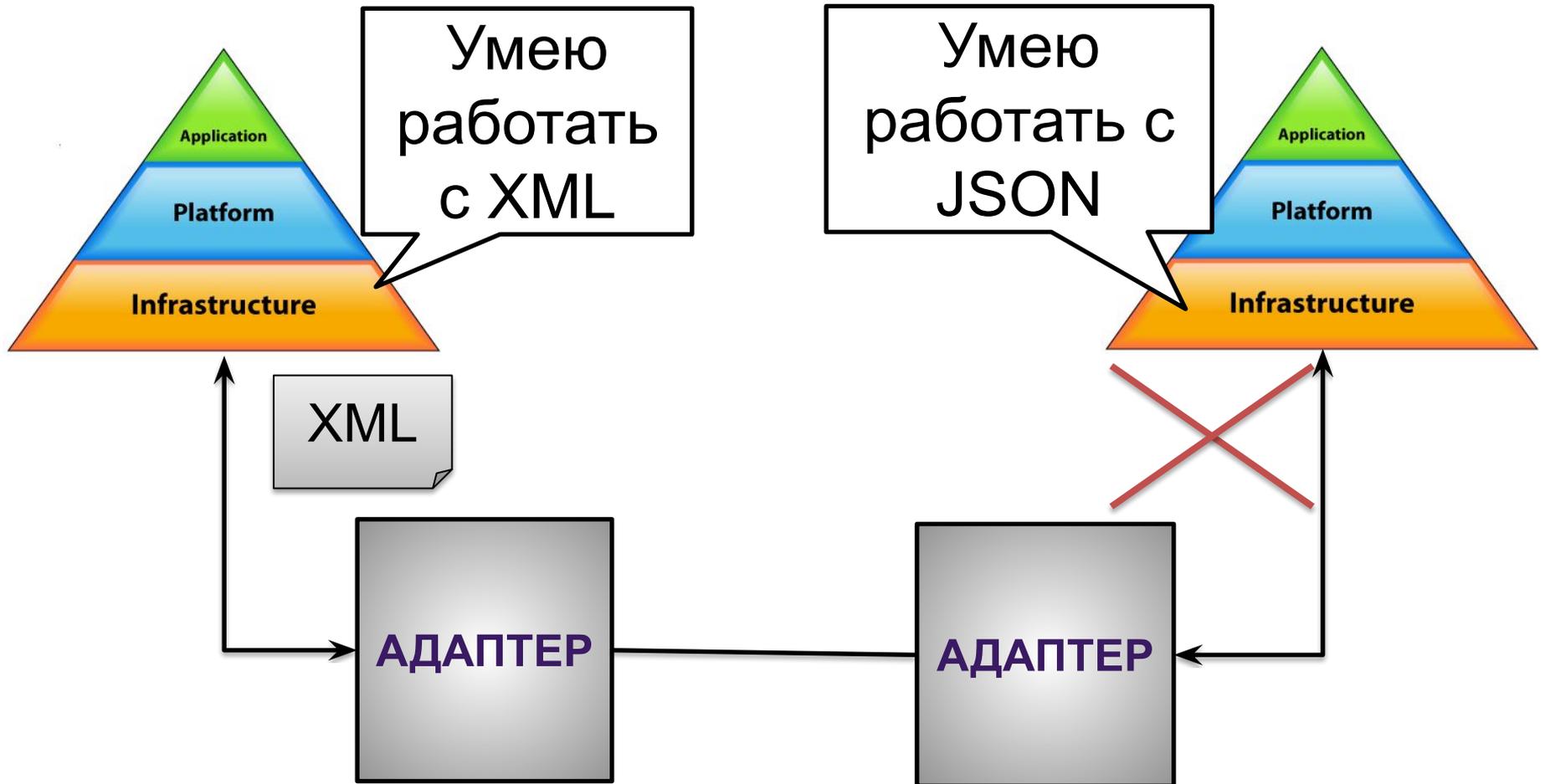
Обычно возможности обратиться на шину достаточно



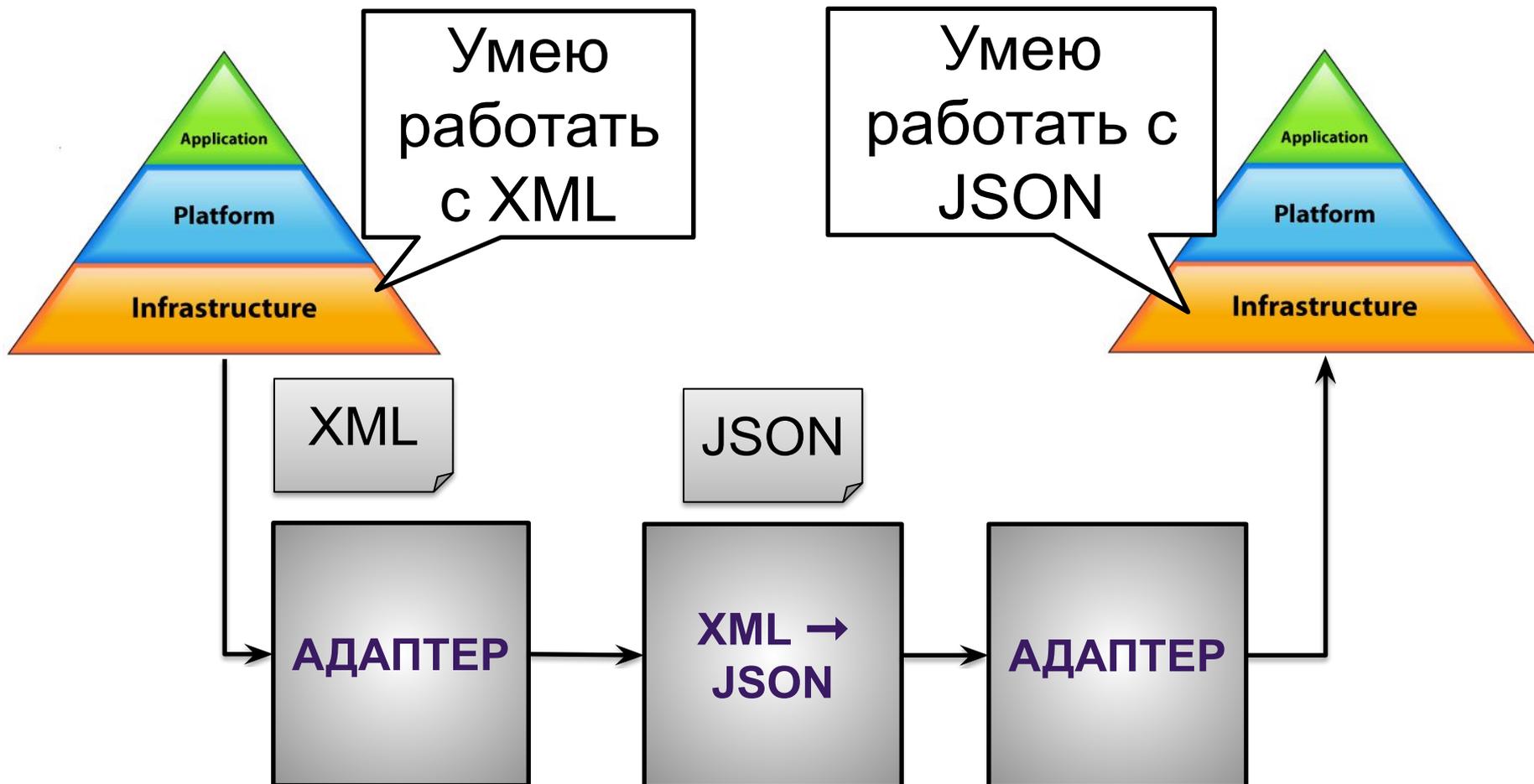
... Но иногда не достаточно
ТОЛЬКО попасть на шину



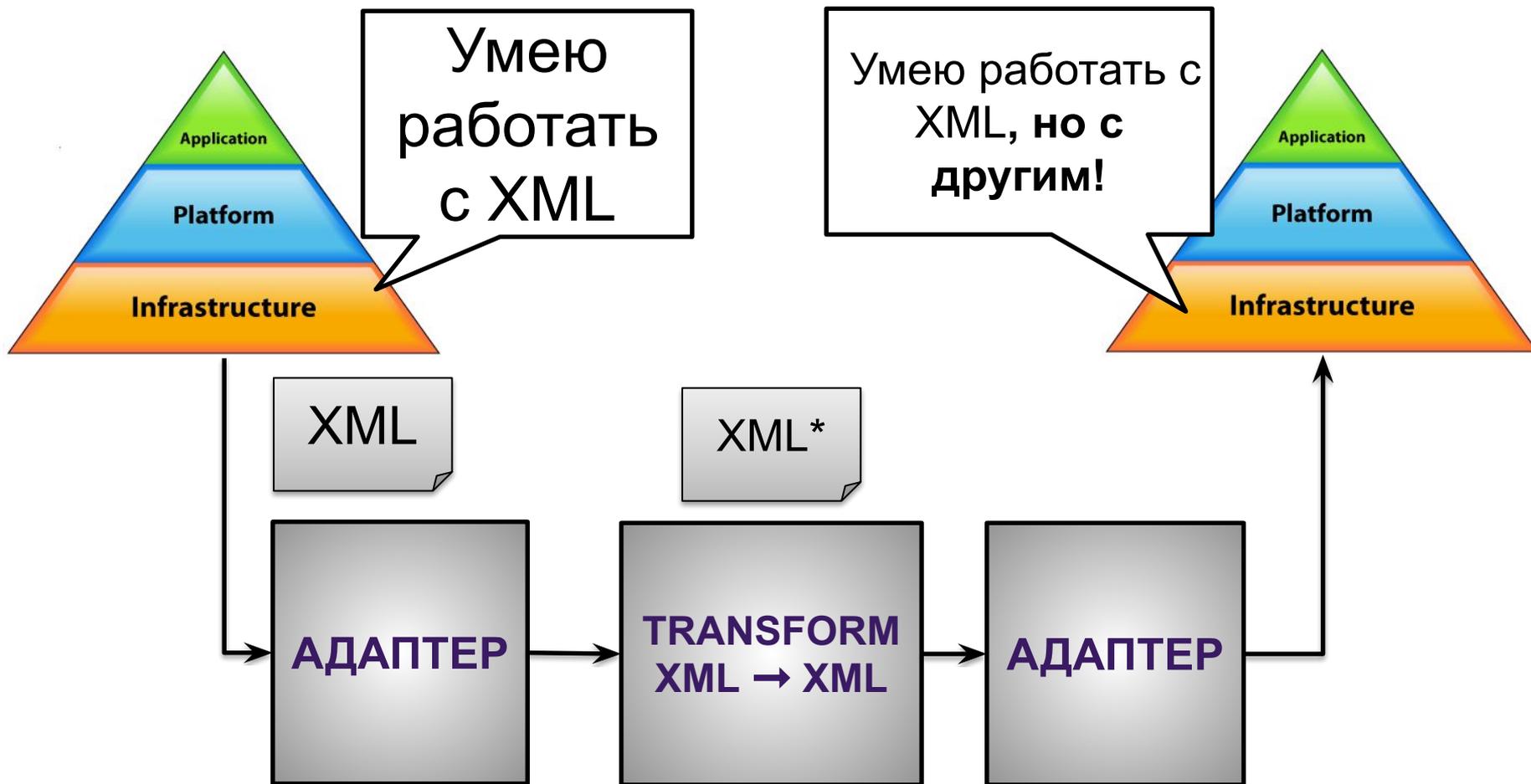
... Но иногда не достаточно
ТОЛЬКО попасть на шину



... Необходимо еще и преобразовать формат



... или трансформировать сообщение внутри формата



Проблемы не заканчиваются

Некоторые системы
представляют одинаковые
данные в разных форматах

Пол	Система 1	Система 2	Система 3
Мужской	М	0	муж
Женский	Ж	1	жен

Предоставление единого формата на ESB

Пол	Система 1	Система 2	Система 3
Мужской	М	0	муж
Женский	Ж	1	жен

**УНИФИКАТОР
ФОРМАТА**

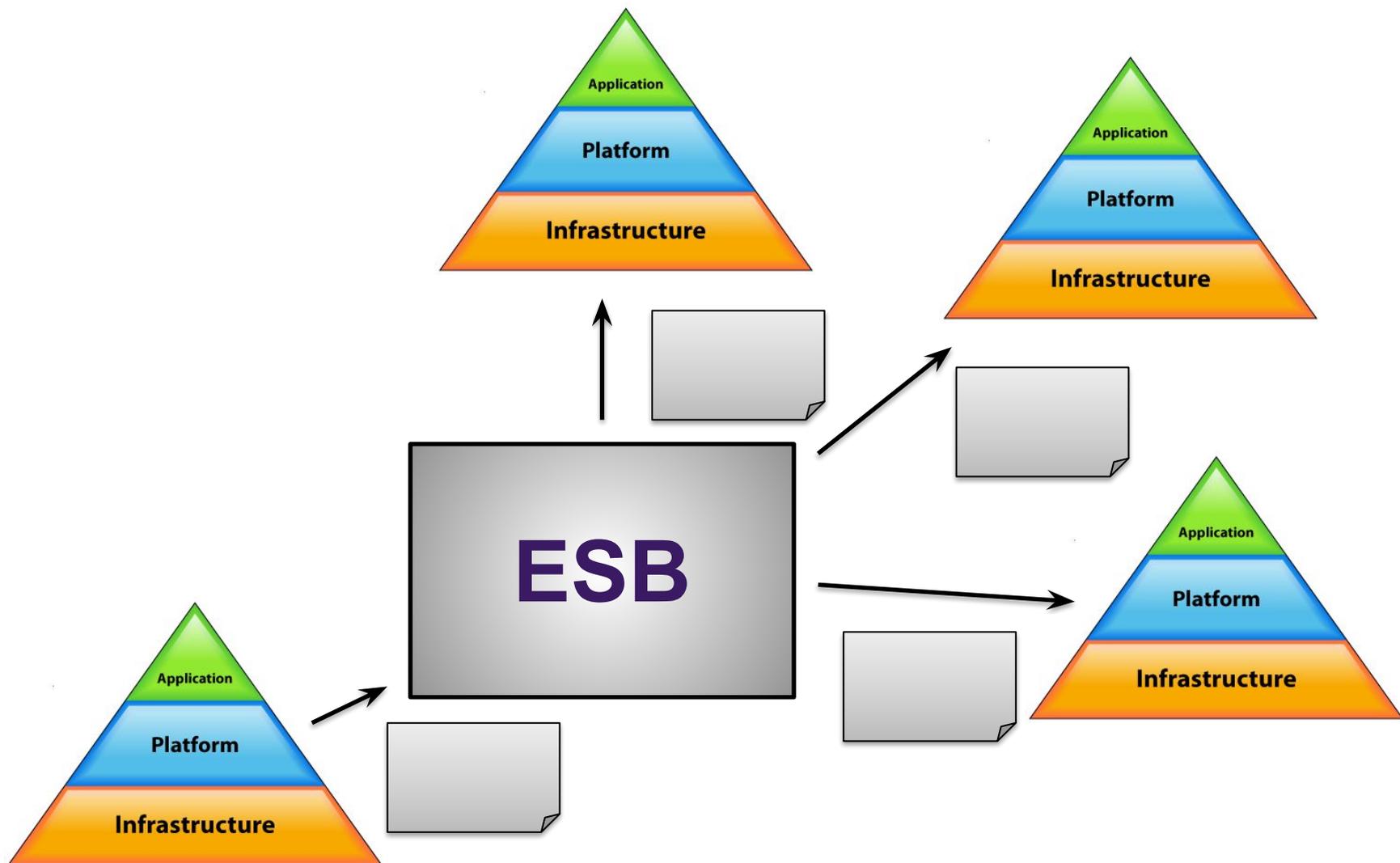
```
graph TD; S1[Система 1] --> U[УНИФИКАТОР ФОРМАТА]; S2[Система 2] --> U; S3[Система 3] --> U; U --> U2[Общий формат];
```

Пол	Общий формат
Мужской	М
Женский	F

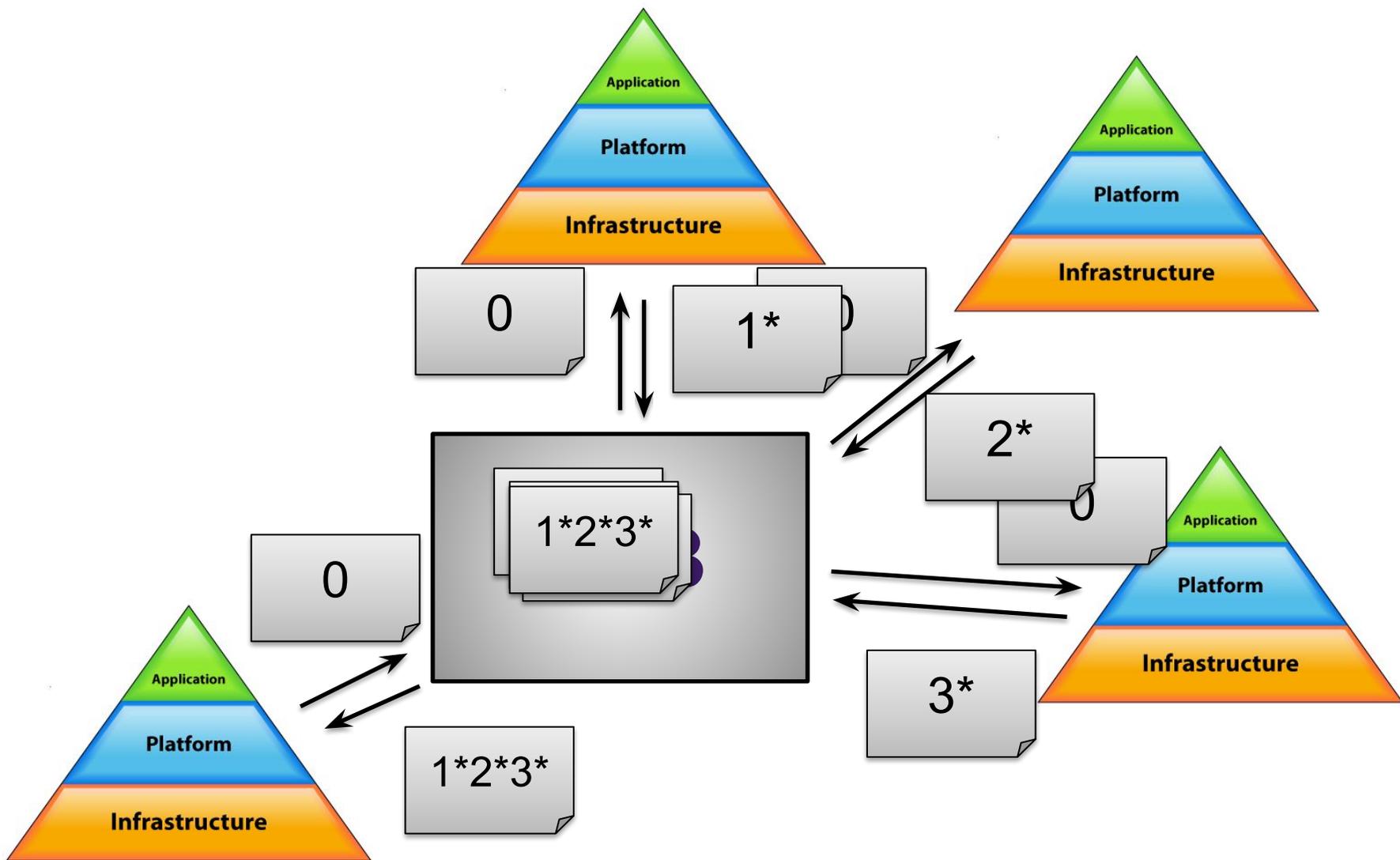
Репликация данных

Некоторые данные нужно
отправить в несколько систем
при этом сделать только 1
ВЫЗОВ

Репликация данных



А иногда и дожждаться ответа от всех



Возможности шины интеграции

- Присоединение систем к единой платформе обмена (Посредством адаптеров)
- Преобразования форматов (e.g. XML → JSON)
- Трансформации форматов (e.g. XML → XML)
- Предоставление единого формата полей
- Распространение (репликация) данных
- etc...

Особенности коммерческих ESB



- Предоставляет MOM, Файловый обмен, БД, Web сервисы, **единый API**
- Единая точка конфигурации
- Единая точка мониторинга
- Единый подход к разработке, развертыванию и обновлениям

Среда разработки

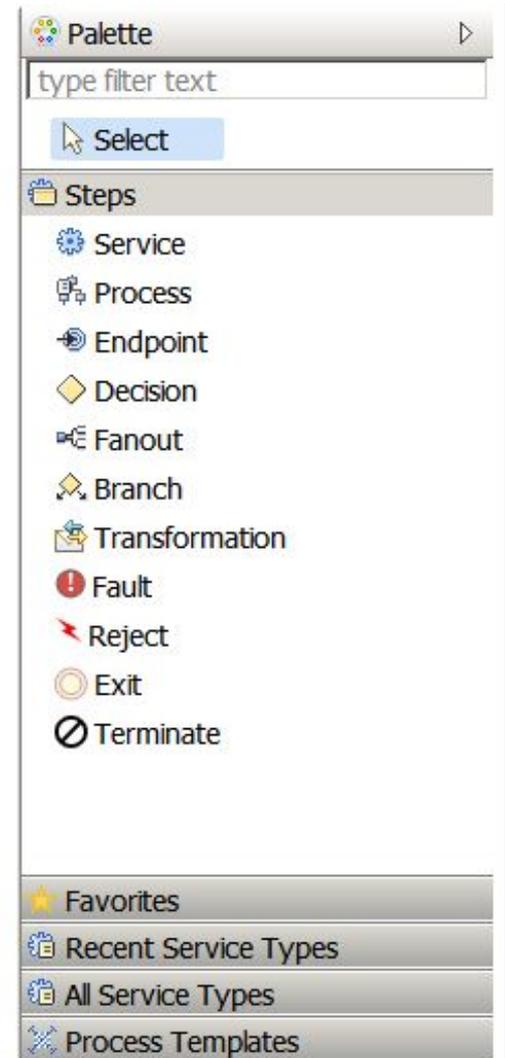
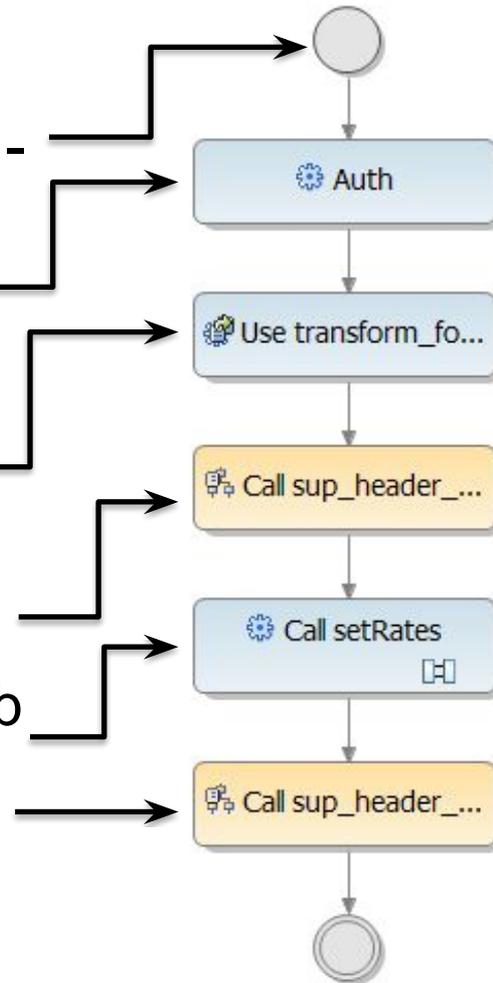
Любое взаимодействие между системами можно представить в виде последовательности действий на шине

Пример взаимодействия

1. Взять сообщение из базы данных
2. Добавить заголовки basic authentication
3. Преобразовать xml → xml (По правилам преобразования)
4. Записать сообщение в лог
5. Вызвать систем по Web сервису
6. Записать сообщение в лог

Пример взаимодействия

1. Сообщение берется из БД особым адаптером - Gateway
2. Добавить заголовки basic authentication
3. Преобразовать xml → xml (По правилам преобразования)
4. Записать сообщение в лог
5. Вызвать систем по Web сервису
6. Записать сообщение в лог



Сегодня мы узнали

Что интегрировать
приложение между собой
сложно, но возможно

Основные подходы к интеграции

- ✓ Файловый обмен
- ✓ RPC (Remote Procedure Call)
- ✓ Общая база данных
- ✓ Системы обмена сообщениями (МООМ – message oriented middleware)
- ✓ Web сервисы

Особенности коммерческих ESB



- Предоставляет MOM, Файловый обмен, БД, Web сервисы, **единый API**
- Единая точка конфигурации
- Единая точка мониторинга
- Единый подход к разработке, развертыванию и обновлениям

Тема ESB и интеграции на
этом не исчерпана

Более подробно можно
узнать, попав к нам на
стажировку 😊

Или почитав умные книжки

Список литературы

- 1 . Шаблоны интеграции корпоративных приложений. : Пер. с англ. М.:ООО “И.Д. Вильямс”.
- 2 . Мартин Фаулер. Шаблоны корпоративных приложений. : Пер. с англ. М.:ООО “И.Д. Вильямс”.

IT – интеграция Современный подход в банках

Автор: Бухматов Павел

vk: vk.com/buhmatov
email: buhmatov@gmail.com
github: github.com/aquatir

