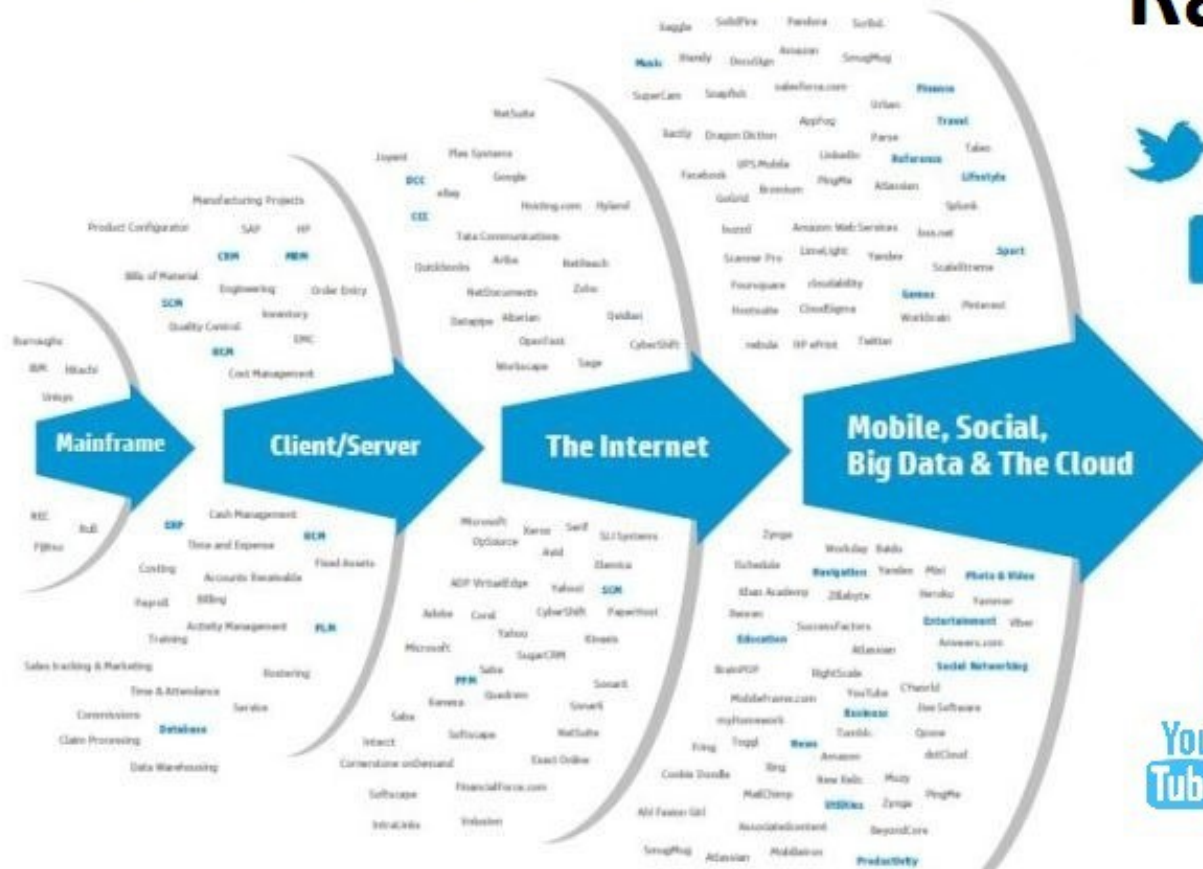


Разработка горизонтально масштабируемой системы легковесных виртуальных машин

Выполнил: Борисов В.В. - студент гр. 8ПИ-51

Руководитель: Казаков М. Г. - к.т.н., старший преподаватель

Необходимость горизонтального масштабирования



Каждые 60 секунд



98,000+ записей в twitter



695,000 обновлений статуса



11,000,000 мгновенных сообщений



698,445 поисков в Google



210,000,000+ отправленных email



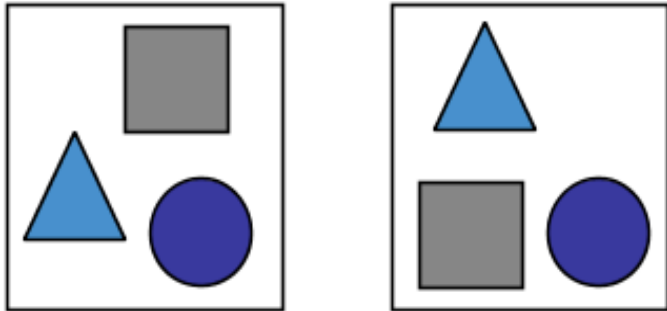
1,820TB созданных данных



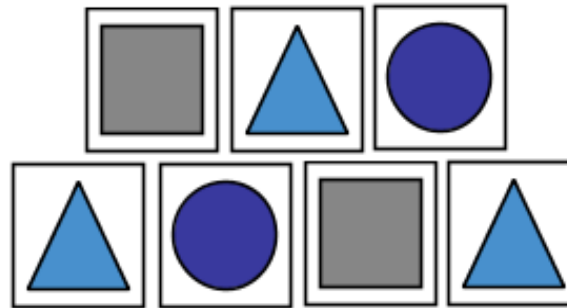
50 часов нового видео на YouTube

Микросервисы - единица масштабирования

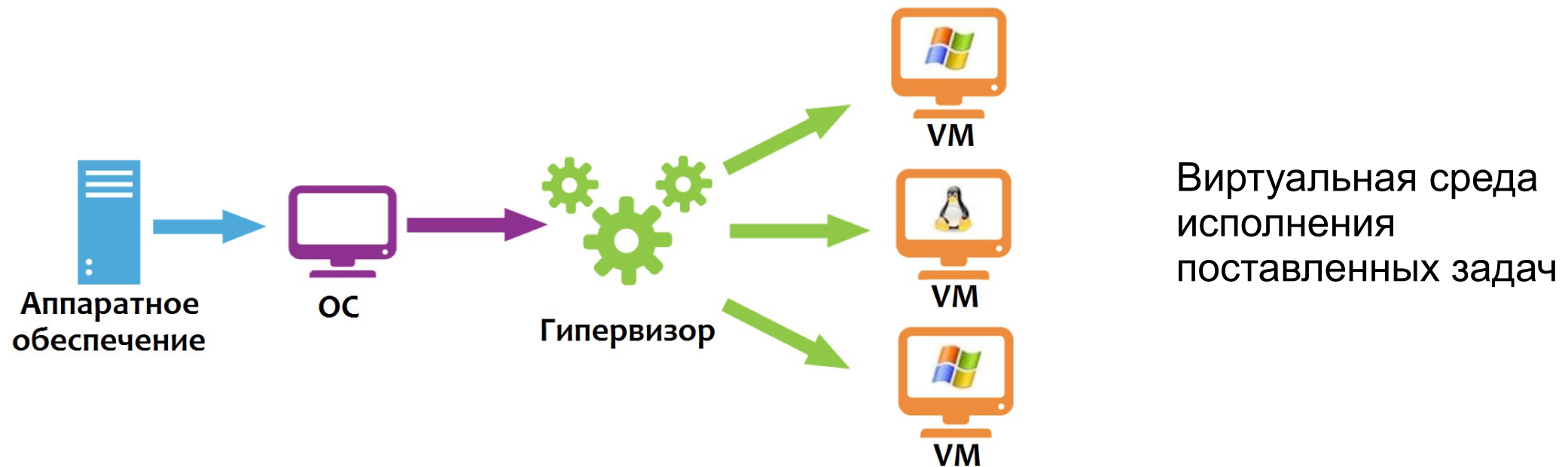
Монолитная архитектура систем



Микросервисная архитектура систем



Логическое представление системы



unikernel

- подход к компоновке ОС,
идеально подходящий для
выполнения однотипных задач

Цель работы

Целями данной работы является:

- Проектирование системы горизонтального масштабирования виртуальных машин и диспетчеризации запросов к исполняемым на них микросервисам.
- Реализация компонентов системы в соответствии с разработанной моделью взаимодействия.

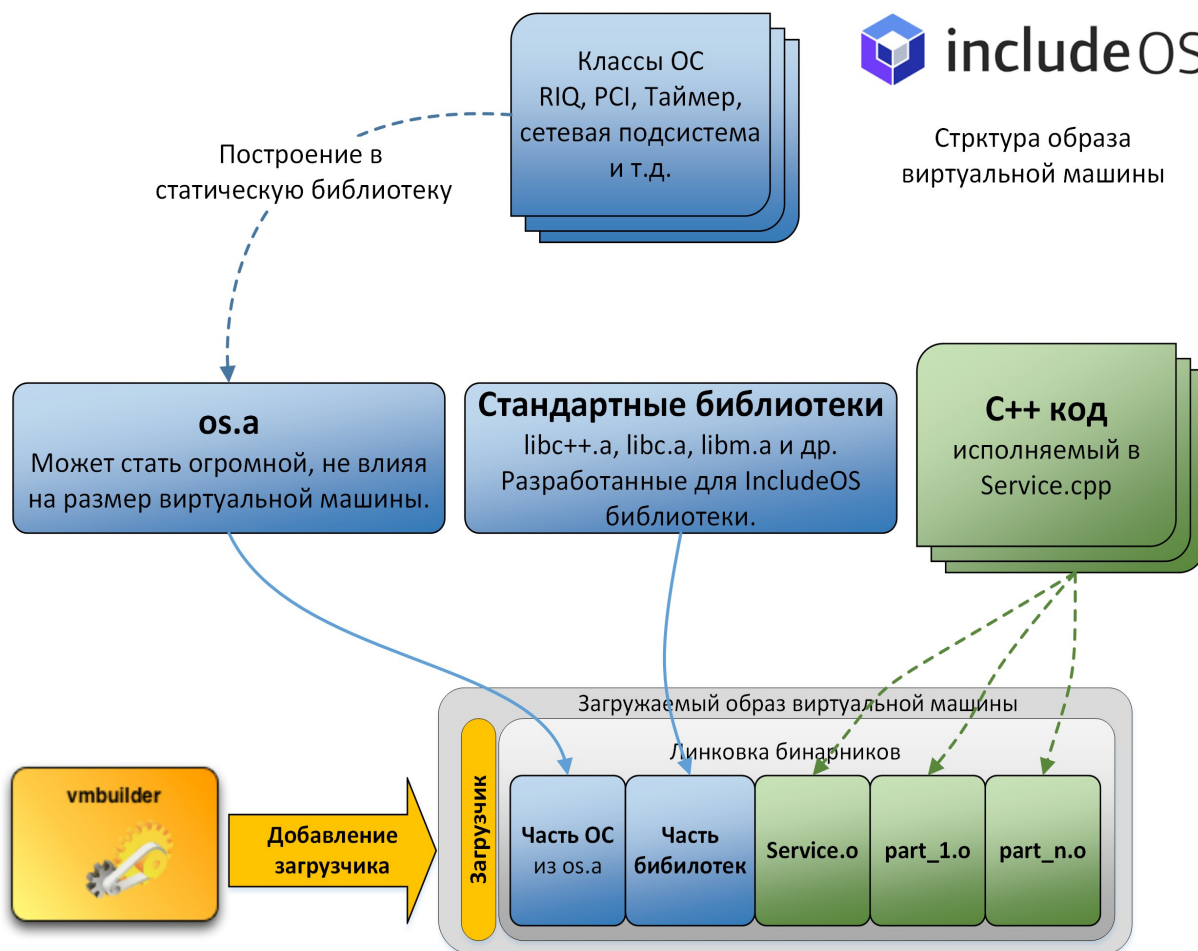
Unikernel ОС на примере IncludeOS



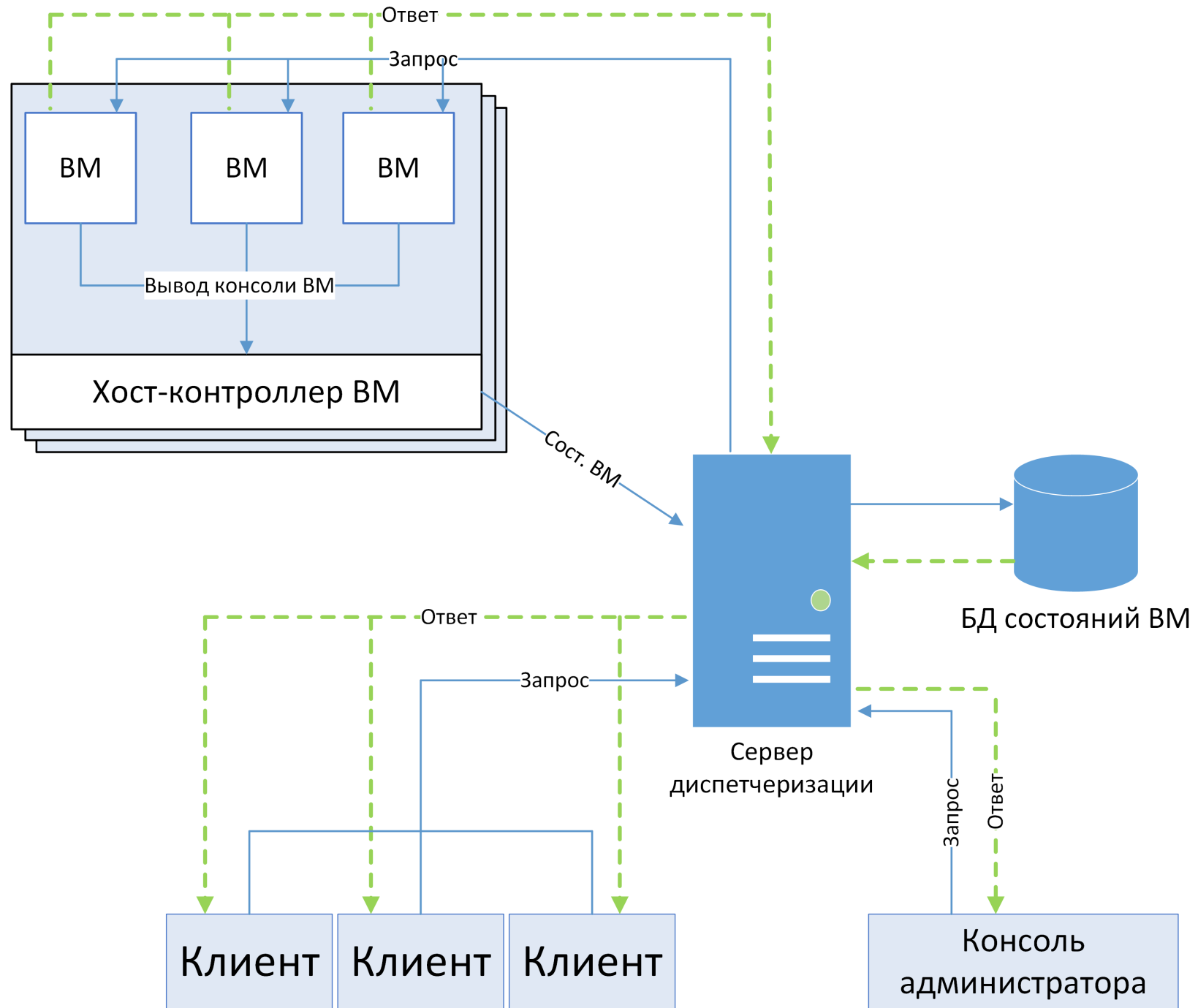
Структура образа
виртуальной машины

Характеристики:

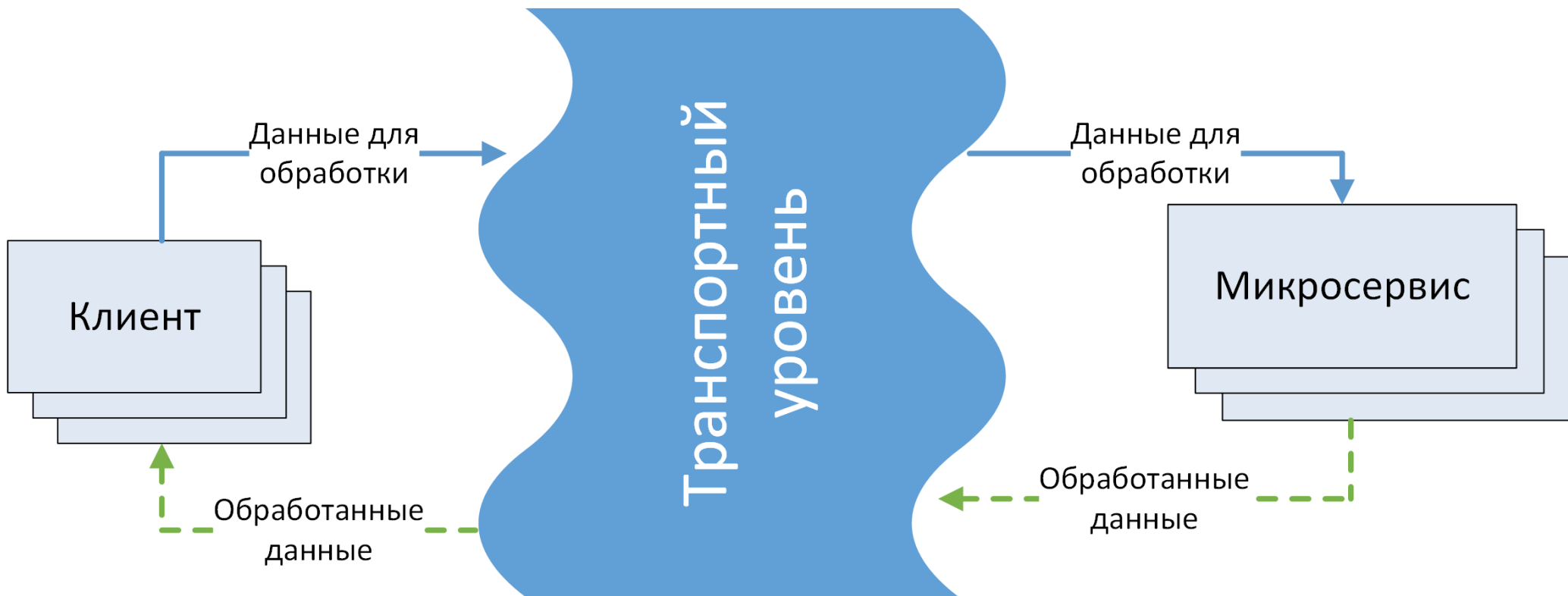
- максимально эффективное использование ресурсов (3,5Mb RAM при простое системы);
- мгновенный запуск (вплоть до 200 мс);
- малый размер загрузочного образа (около 1,5Mb);
- не требует установки для работы;
- минимальные энергозатраты для поддержания работы одного образа;
- исполняет задачу на нулевом кольце защиты;
- ориентирована на работу с web-сервисами.



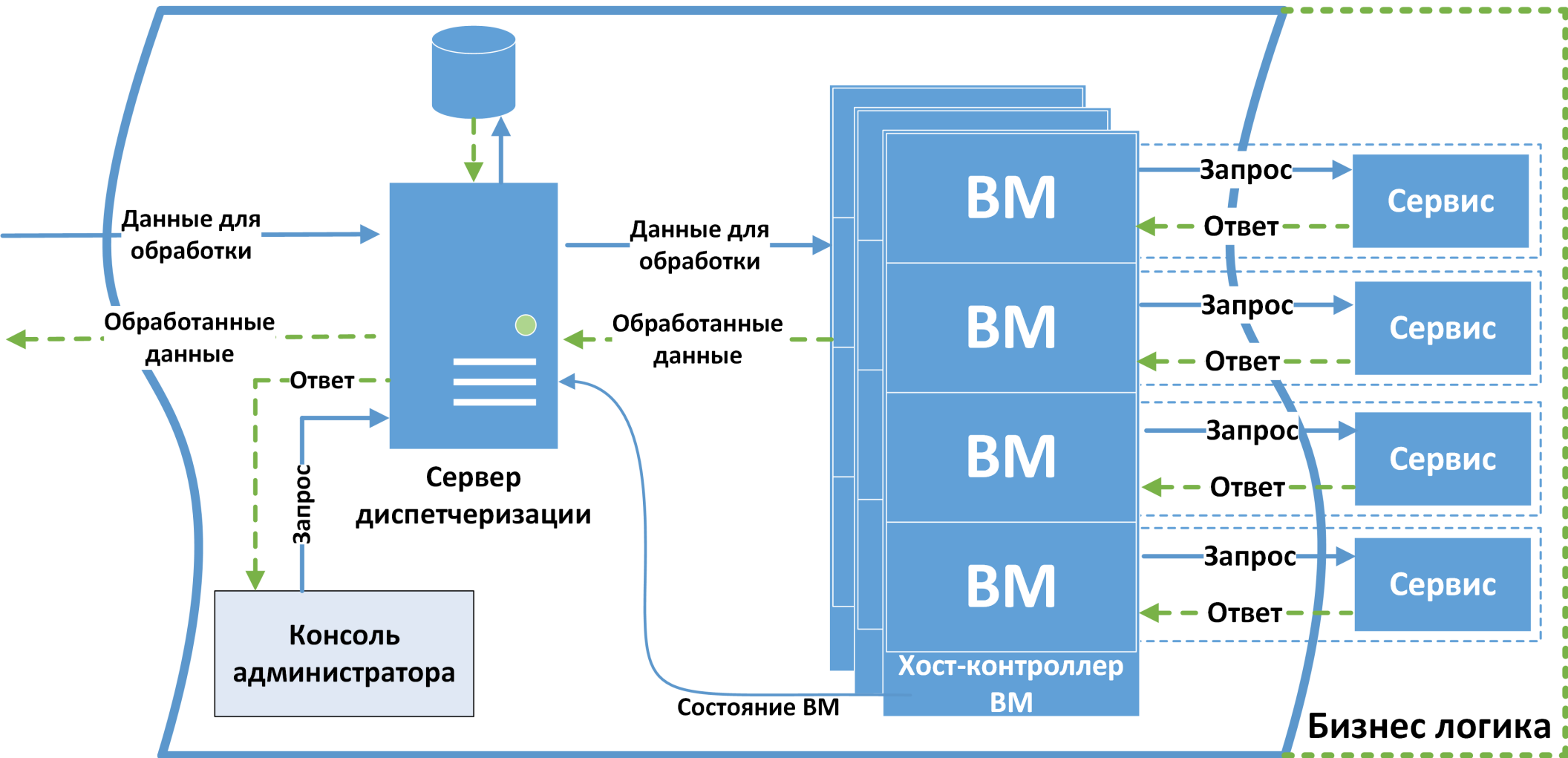
Общая схема разработанной системы



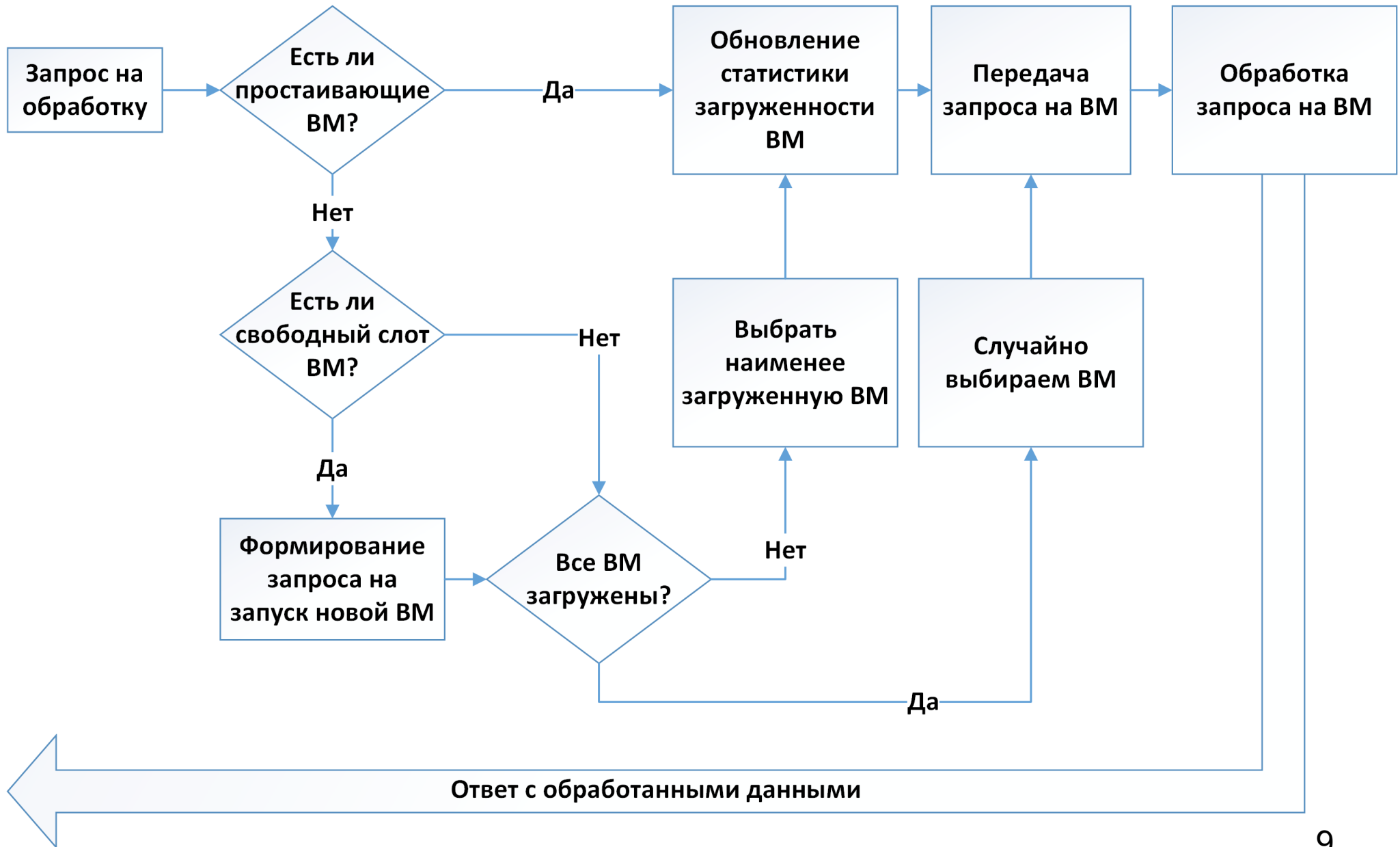
Обработка запросов с точки зрения бизнес логики



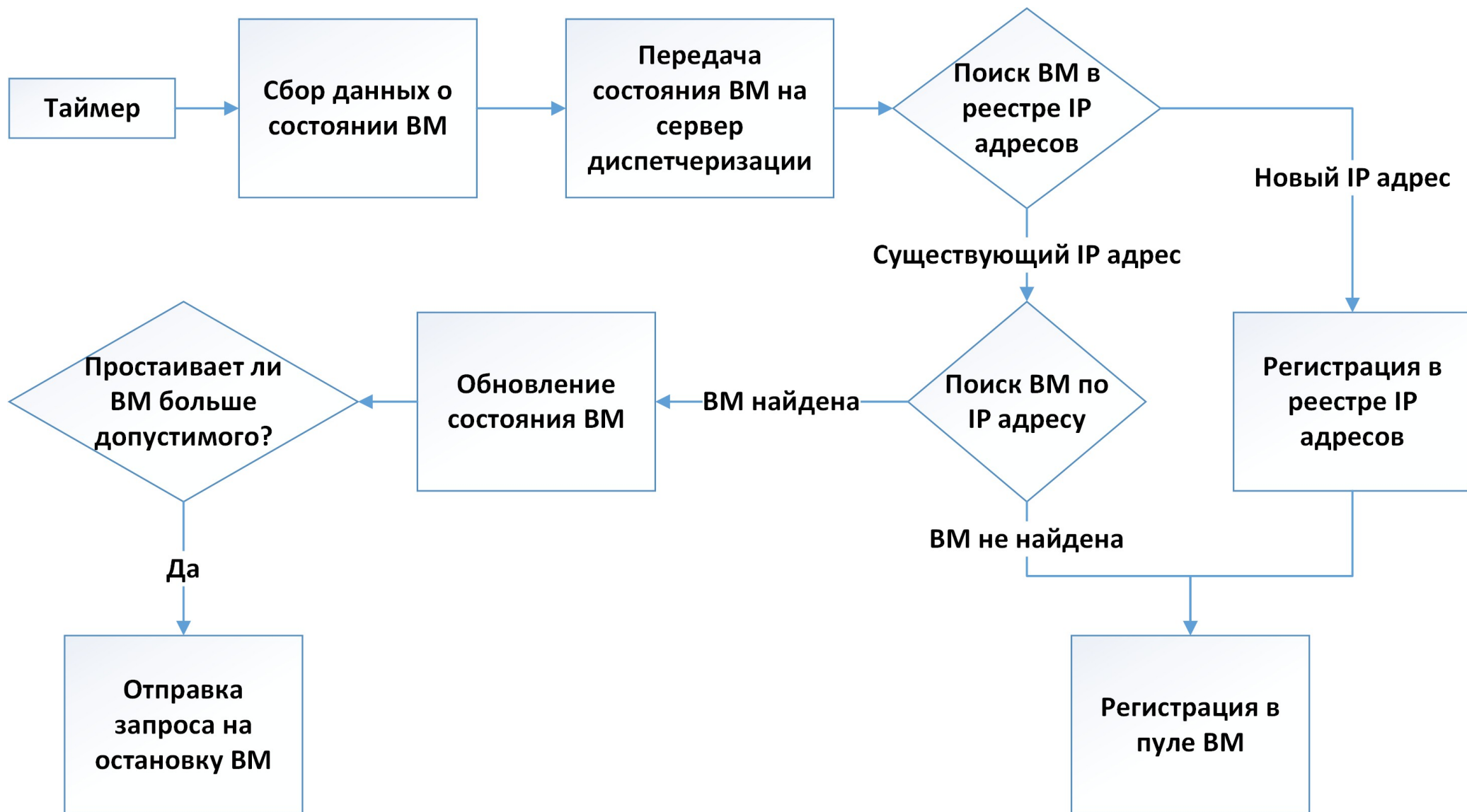
Транспортный уровень системы



Процесс обработки входящего запроса



Процесс мониторинга виртуальных машин



Структура базы данных состояния VM

system_params	
key	
value	



	key	value
1	free_count	5
2	last_connect_ms	600000
3	ip_reserve_ms	60000
4	min_vm_count	2

vmachine	
id_vmachine	PK
id_ip	
CPU	
RAM	
free_count	
last_connect	

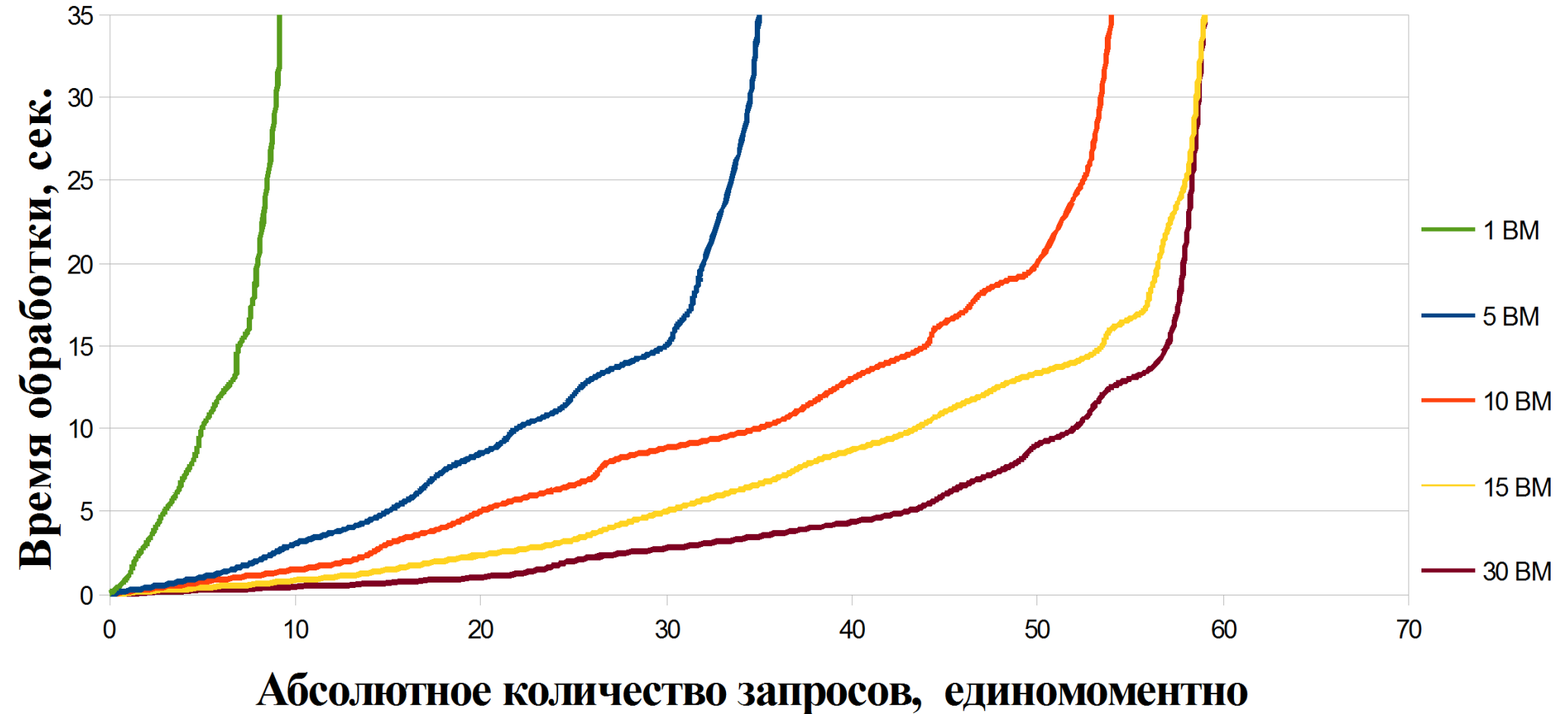


ip_pool	
id_ip	PK
IP	
PORT	
reserve	
last_reserve	

Элементы научной новизны работы

- Создана новая модель горизонтально масштабируемой распределенной обработки запросов на основе легковесных виртуальных машин.
- Показана применимость виртуальных машин с ядром IncludeOS для создания горизонтально масштабируемых систем.

Пример масштабирования тестового сервиса



Наивное вычисление количества простых чисел до заданного N

Пример работы системы

The screenshot displays a system interface with several components:

- Клиент (Client):** A window titled "Настройка" (Configuration) showing a table of tasks.
- Журнал сервера (Server Log):** A window titled "super_server.log" showing log entries.
- Контроллер (Controller):** A window showing system boot logs.
- Консоль администратора (Administrator Console):** A window showing VM configuration details.

Клиент (Client) Table:

Название задачи	Состояние
test1	выполнена
test2	выполнена
test3	выполнена
test4	выполнена
test5	выполнена
test6	выполнена

Журнал сервера (Server Log) Excerpt:

```
20:54:04,527 INFO http-thread-pool-9999(1) getFreeIP:308 - Free Ip = 10.0.0.46:80 was found
20:54:04,527 INFO http-thread-pool-9999(1) sendingStartKillRequest:46 - Sending 'GET' request
(command = start_new!10.0.0.46) to URL : http://10.0.0.45
20:54:04,540 INFO http-thread-pool-9999(1) send:73 - Response Code : 200
20:54:04,542 INFO http-thread-pool-9999(1) processAddJob:138 - VM ip: 10.0.0.46 starting
successfully
20:54:04,542 INFO http-thread-pool-9999(1) updateIPState:354 - Update IP(10.0.0.46) state = true
20:54:04,560 INFO http-thread-pool-9999(1) getFistFreeVM:140 - VM with free_count>0 not found
20:54:04,560 INFO http-thread-pool-9999(1) resetVMfreeCount:449 - Update vm with id_vmachine = 75
```

Контроллер (Controller) Excerpt:

```
[ CMOS ] RTC is 24 hour format, BATT mode
[ RTC ] Enabling regular clock sync with CMOS
[ Kernel ] Initializing plugins
[ Kernel ] Starting IncludeOS minimal
Start params: IncludeOS_example !10.0.0.46
[ Super stack ] Constructing stacks
[ Super stack ] Creating stack for Nic et
[ Inet4 ] Bringing up a IPv4 stack
[ Inet4 ] Negotiating DHCP...
[ DHCPv4 ] Negotiating IP-address (x
[ Inet4 ] Network configured
IP: 10.0.0.46
Netmask: 255.255.255.0
Gateway: 10.0.0.1
DNS Server: 10.0.0.1
*** SERVICE STARTED ***
IncludeOS v0.10.0-rc.2-2-gc1e9172
+--> Running [ IncludeOS minimal example
```

Консоль администратора (Administrator Console) Table:

ID	CPU	Ram	IP	Port
87	59	3259712	10.0.0.48	80
92	26	3467774	10.0.0.45	80
89	100	3251712	10.0.0.49	80
90	90	3267904	10.0.0.47	80
85	98	3362927	10.0.0.51	80

Сведения о разработанной системе

Объём написанного кода: около 3800 строк.

Используемые языки программирования:

- Java
- C++
- SQL

Используемые ОС:

- Windows
- GNU/Linux
- IncludeOS

Используемые технологии:

- NetBeans
- PostgreSQL
- PgAdmin
- GlassFish
- JfreeChart
- Ccmake
- Библиотека Gson
- Библиотека HttpCore от Apache
- Библиотека HttpClient от Apache
- Библиотека Log4j от Apache

Заключение

При выполнении данной работы были получены следующие результаты:

- Спроектирована и реализована система горизонтального масштабирования виртуальных машин и диспетчеризации запросов к исполняемым на них микросервисам.
- Реализованы компоненты системы в соответствии с разработанной моделью взаимодействия.

Результаты работы были продемонстрированы в XIV всероссийской научно-технической конференции «Наука и молодёжь - 2017»