

VISION LABS

ОБЩАЯ ПРЕЗЕНТАЦИЯ

Alexander Khanin, Founder & CEO

ООО «АРКТИЧЕСКИЙ 12»

ОФИЦИАЛЬНЫЙ ПРЕДСТАВИТЕЛЬ НА ТЕРРИТОРИИ РЕСПУБЛИКИ БЕЛАРУСЬ

Тел: 8 029 794-49-33

E-mail: info@ais4life.com

Web: <https://ais4life.com>



VisionLabs LUNA – это платформа распознавания лиц, которая позволяет в режиме реального времени идентифицировать и верифицировать людей по изображению лица.

Полный цикл распознавания может занимать менее 2 секунд.

Применение :



Удаленное обслуживание



Борьба с мошенничеством



Повышение лояльности
клиентов



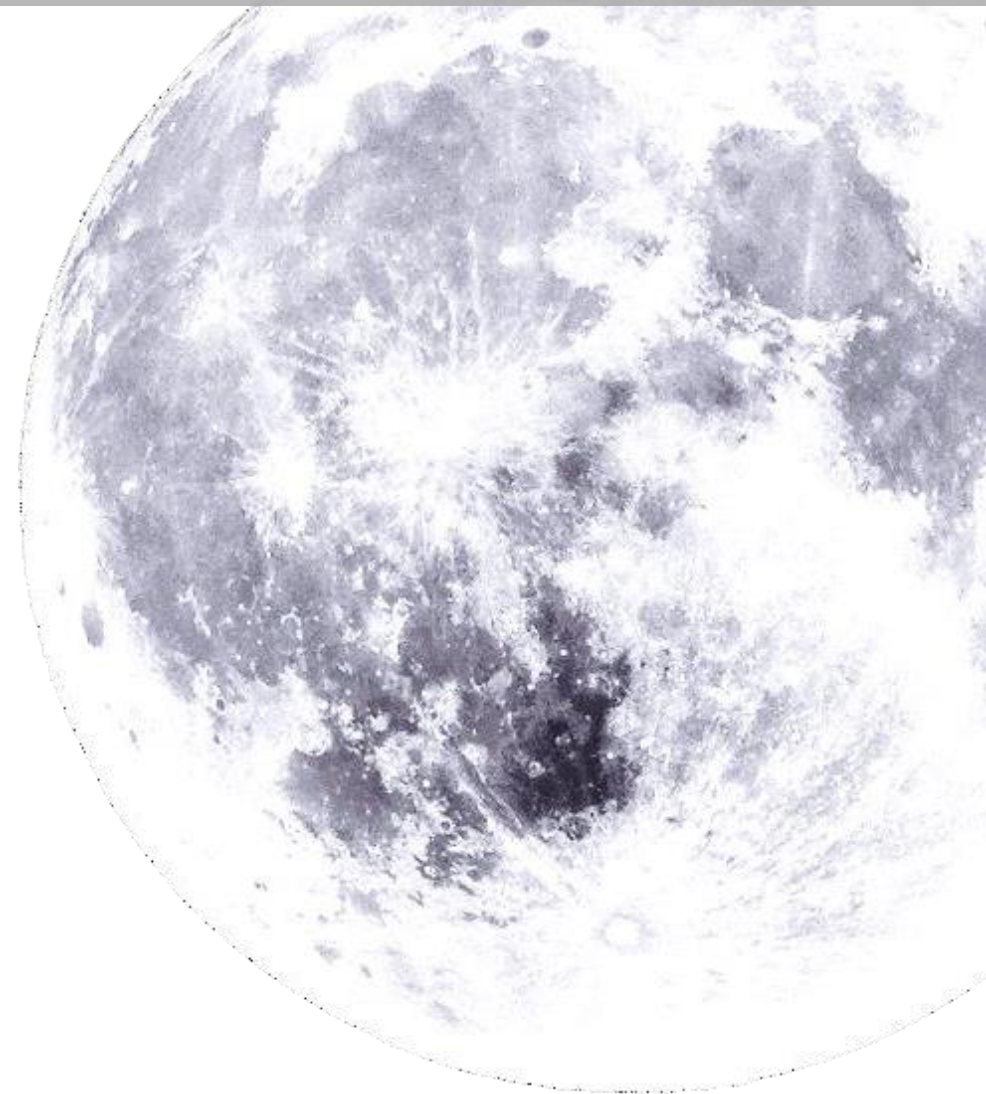
Ускорение обслуживания



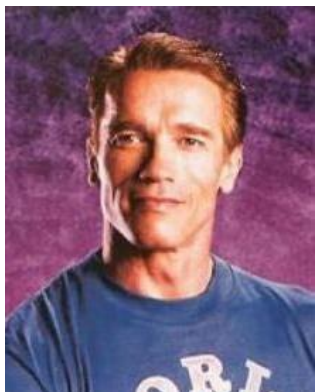
Общественная безопасность



Авторизация сотрудников



Верификация (1:1)

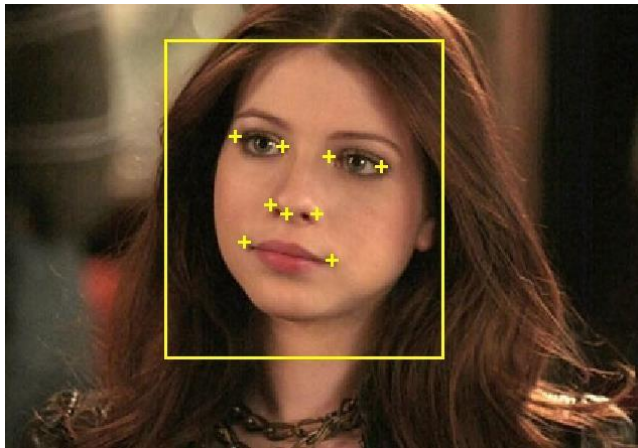


Идентификация (1:N)

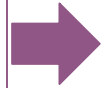


КАК ЭТО РАБОТАЕТ?

1. Детектирование лиц



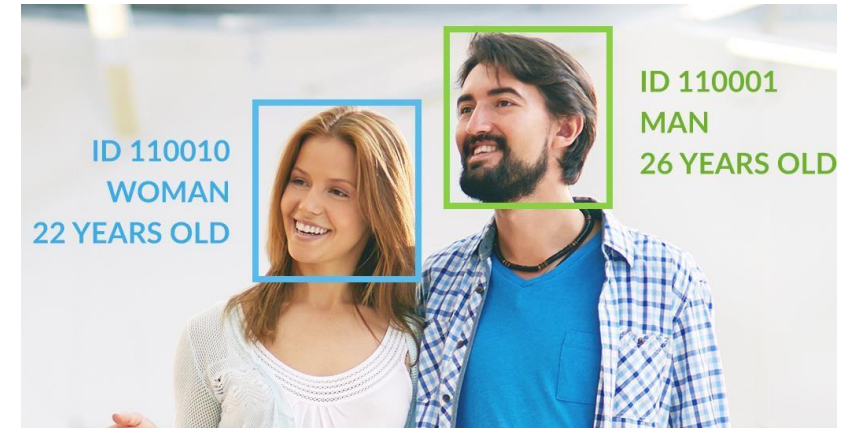
Масштаб
+ 2D поворот +
кроп



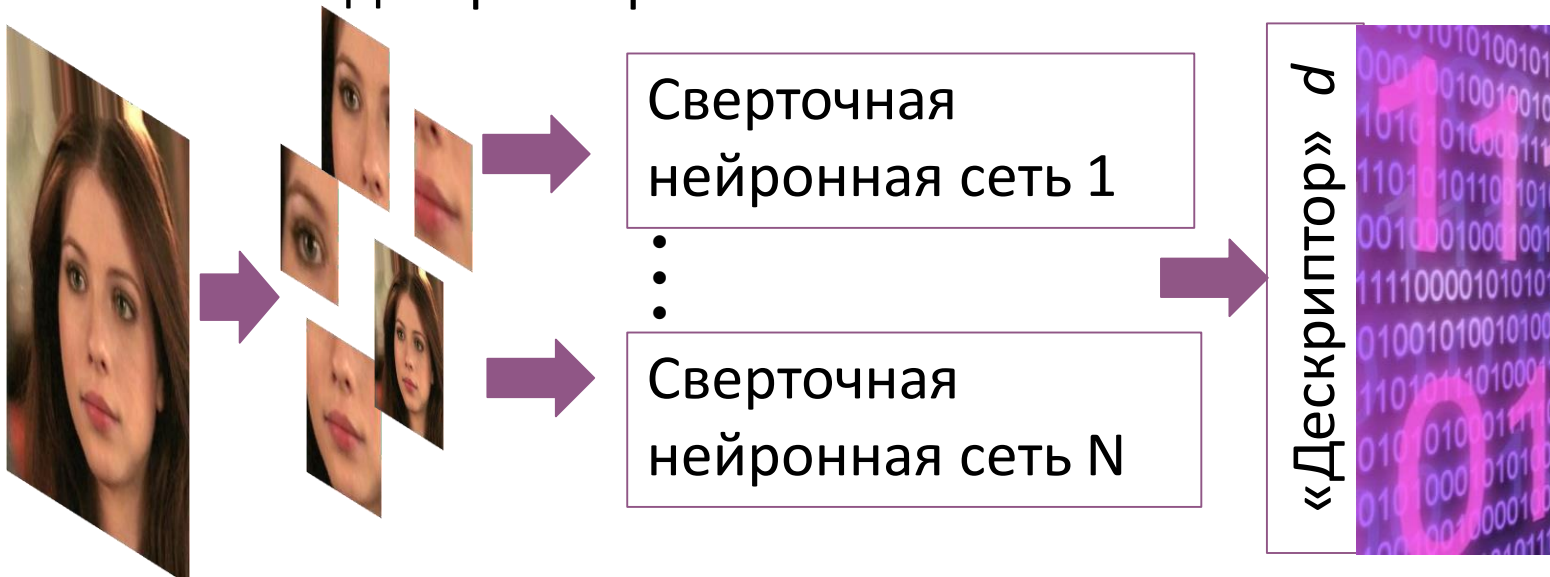
2. Предобработка изображения



3. Определение качества, оценки: пол, возраст, etc.



5. Извлечение «дескриптора»



5. Вычисление степени схожести дескрипторов лиц

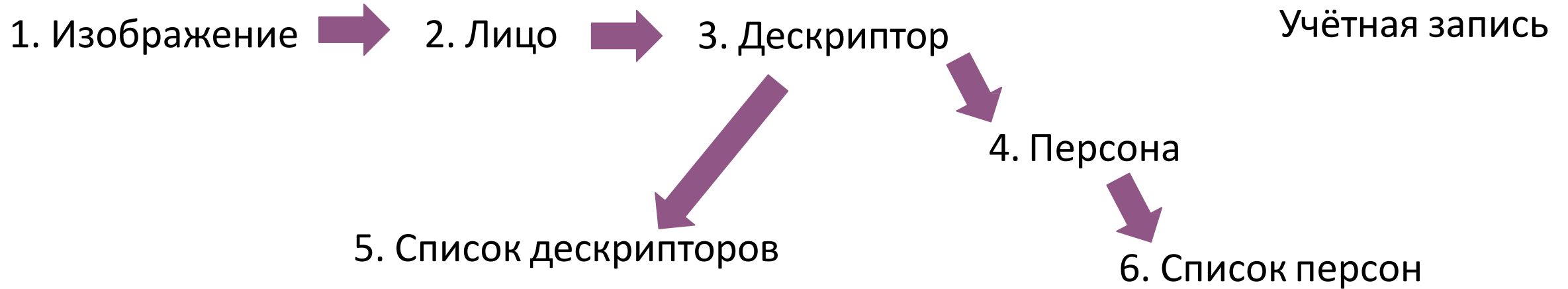
$$D_{i,j} = ||d_i - d_j||_2$$

$D(\text{Image 1}, \text{Image 2}) = 99,5\%$

$D(\text{Image 3}, \text{Image 4}) = 5\%$

LUNA - многопользовательская система:

- Пользователь системы имеет отдельную учётную запись;
- Авторизация запросов через HTTP базовую авторизацию и токены;
- Не пересекающиеся зоны видимости данных;
- Подсчёт запросов и статистические метрики использования аккаунта.



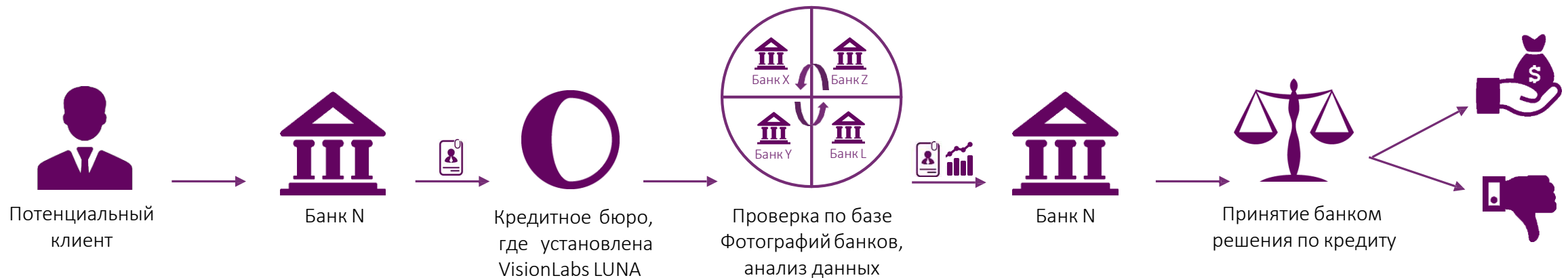
- Детекция лиц на передаваемом по API изображении;
- Определение признаков лица и оценку качества изображения лица (опционально);
- Извлечение и хранение дескриптора, представляющего лицо;
- Группирование дескрипторов лиц в следующие сущности: списки, персоны и списки персон;
- Сравнение дескрипторов друг с другом, используя высокоуровневые сущности для поиска похожих лиц.

Кейс 1: Борьба с мошенничеством

Распознавание лиц позволяет банкам выявлять мошенников и снижать кредитные риски.



Использование в кредитных бюро позволяет многим банкам использовать консолидированные результаты.



Кейс 2: Самообслуживание и дистанционное обслуживание

Распознавание лиц позволяет банку автоматизировать и упростить процесс подачи заявок на кредитные продукты и ускорить обработку информации.

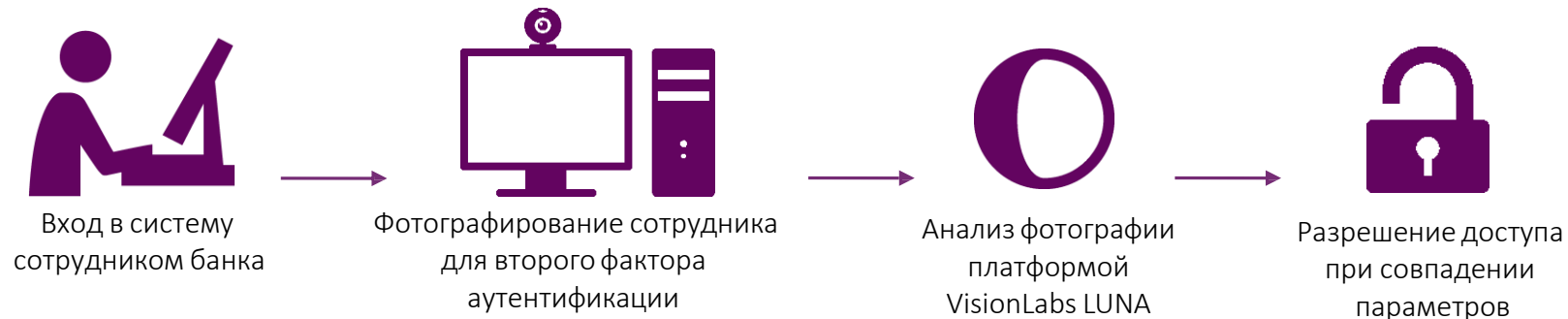


Кейс 3: удалённые каналы продаж и партнёрские программы

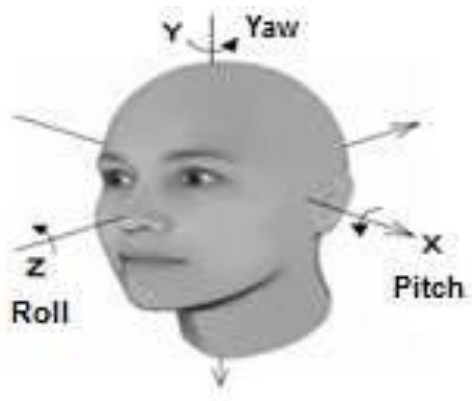
Распознавание лиц позволяет банку персонализировать предложения для клиентов, даже когда они не находятся в отделении банка.



Распознавание лиц позволяет банку защищать конфиденциальную информацию своих клиентов.



Источники изображений лиц, требования



Roll: +- 25
Pitch: +- 15
Yaw: +- 15



ГОСТ Р ИСО/МЭК 19794-5

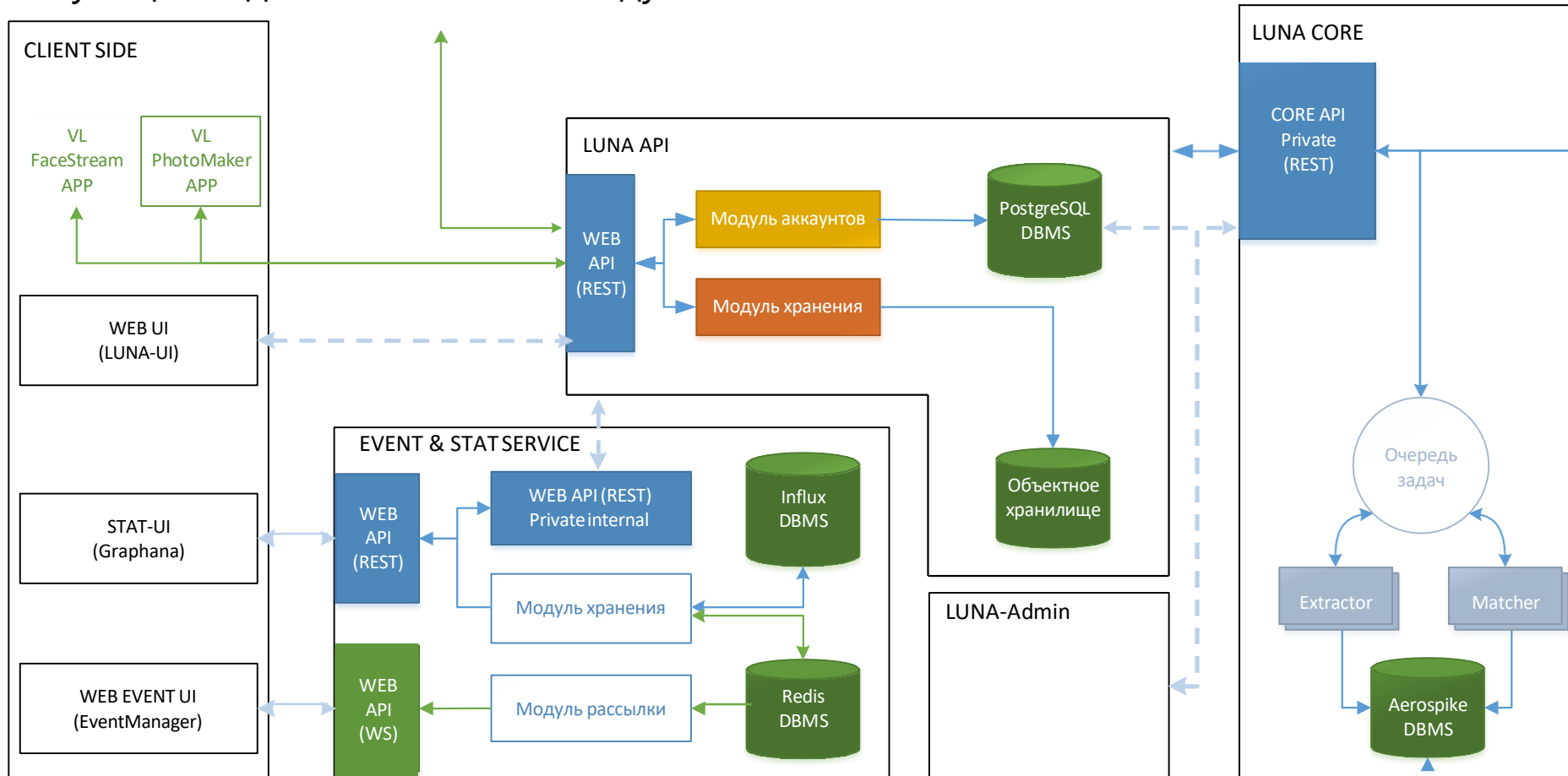
Варианты получения фотоизображения лица

- Кооперативный, когда человек знает, что его снимают и предпринимает усилия для того, чтобы фотопортрет получился как можно качественней.
- Не кооперативный, работа которого подразумевает фоновый режим, человек может не взаимодействовать с камерой, фотопортрет выбирается из тех кадров, которые удалось получить таким образом.

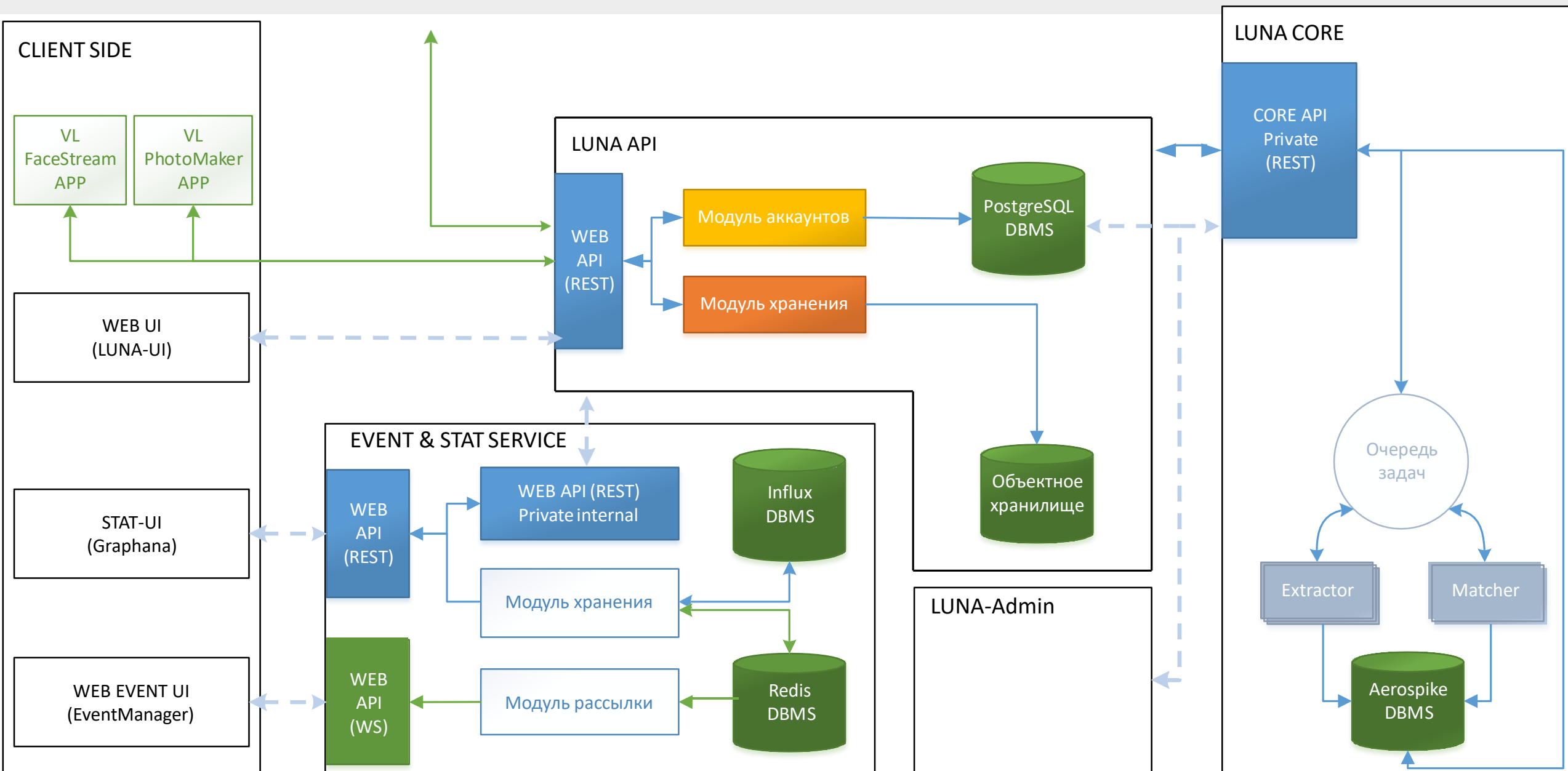
ПЛАТФОРМА LUNA

Архитектура LUNA Platform

- LUNA-API – обеспечивает высокоуровневый публичный интерфейс взаимодействия (RESTful API) и разграничение доступа к данным;
- LUNA-CORE – реализует алгоритмы извлечения и сравнения дескрипторов лиц, функции хранения дескрипторов;
- Сопутствующие и дополнительные модули: Events&Stats, LUNA-UI, LUNA-Admin, etc.

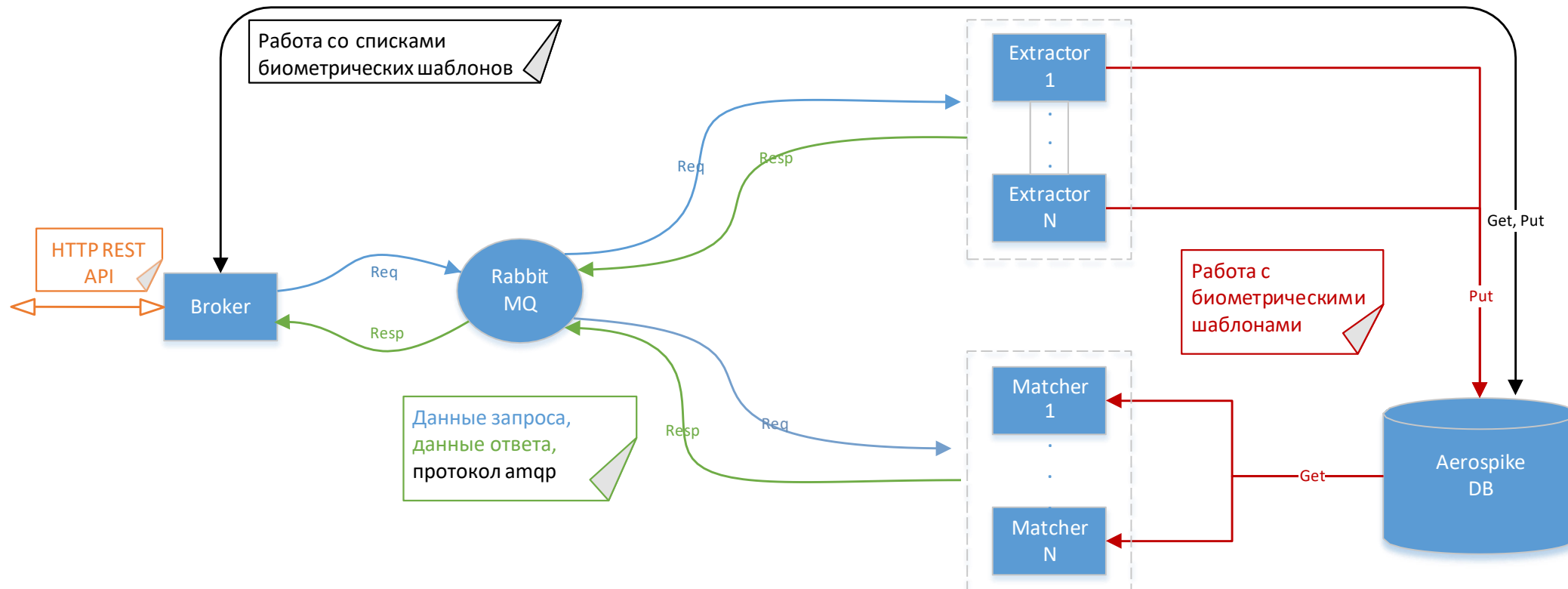


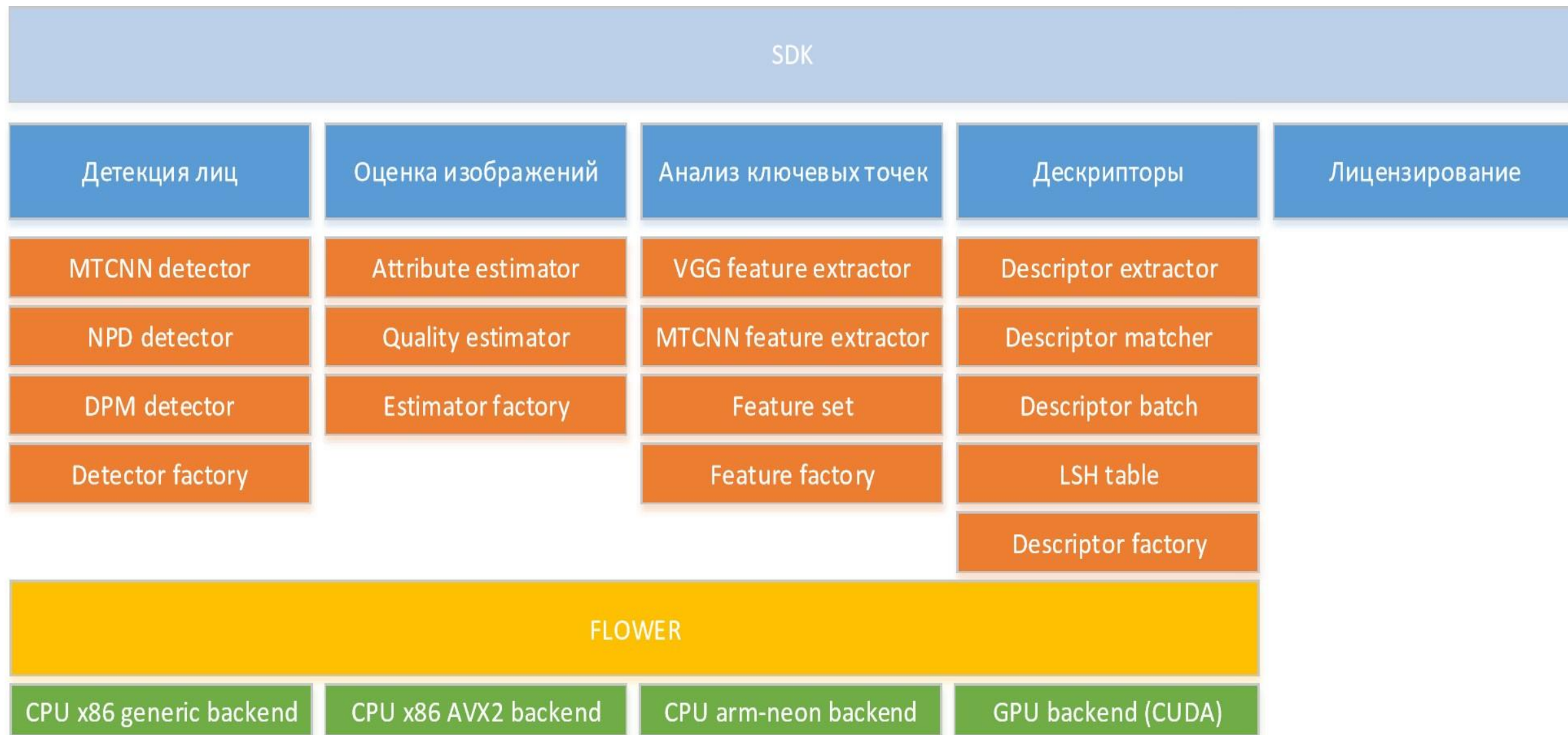
Общая схема LUNA

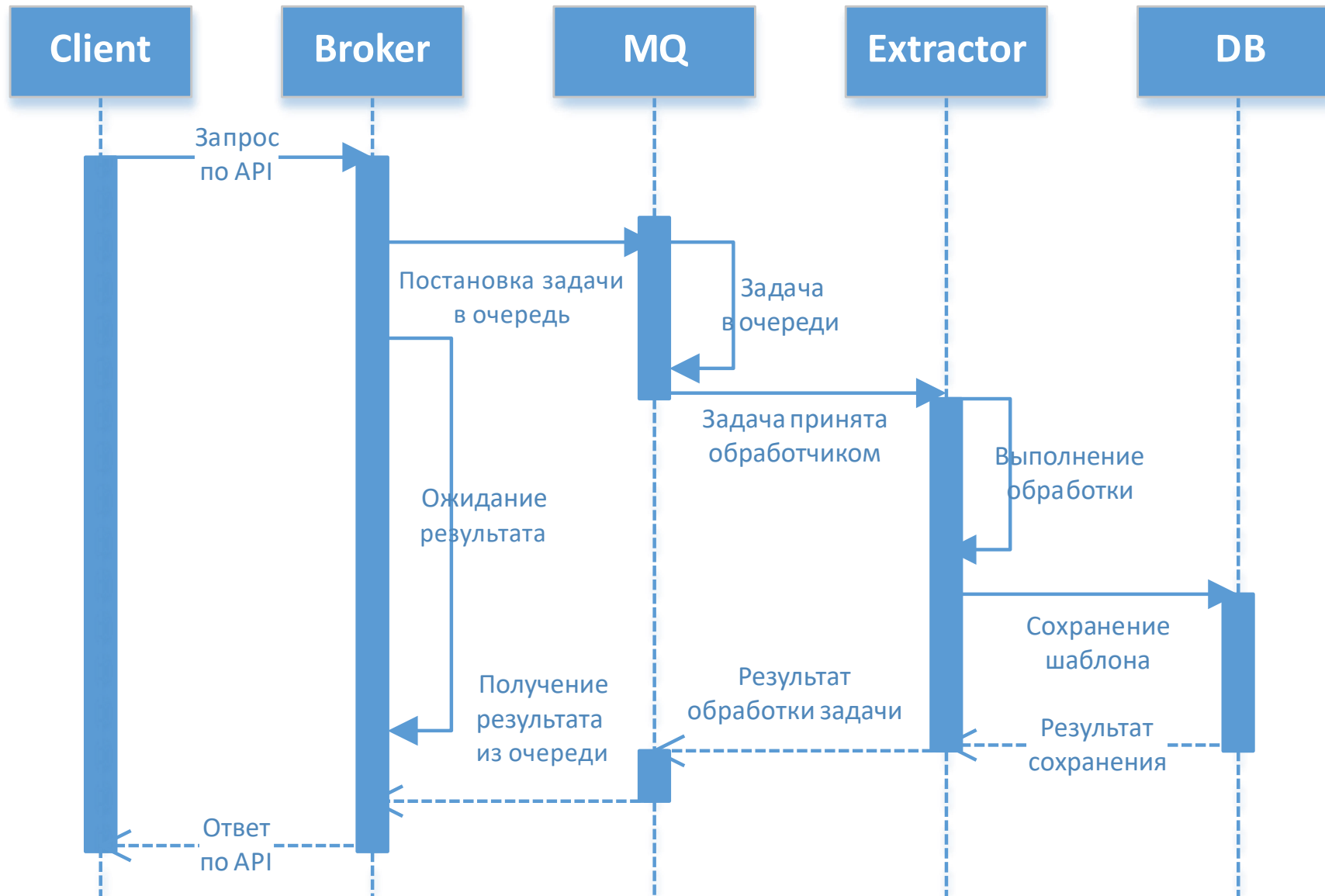


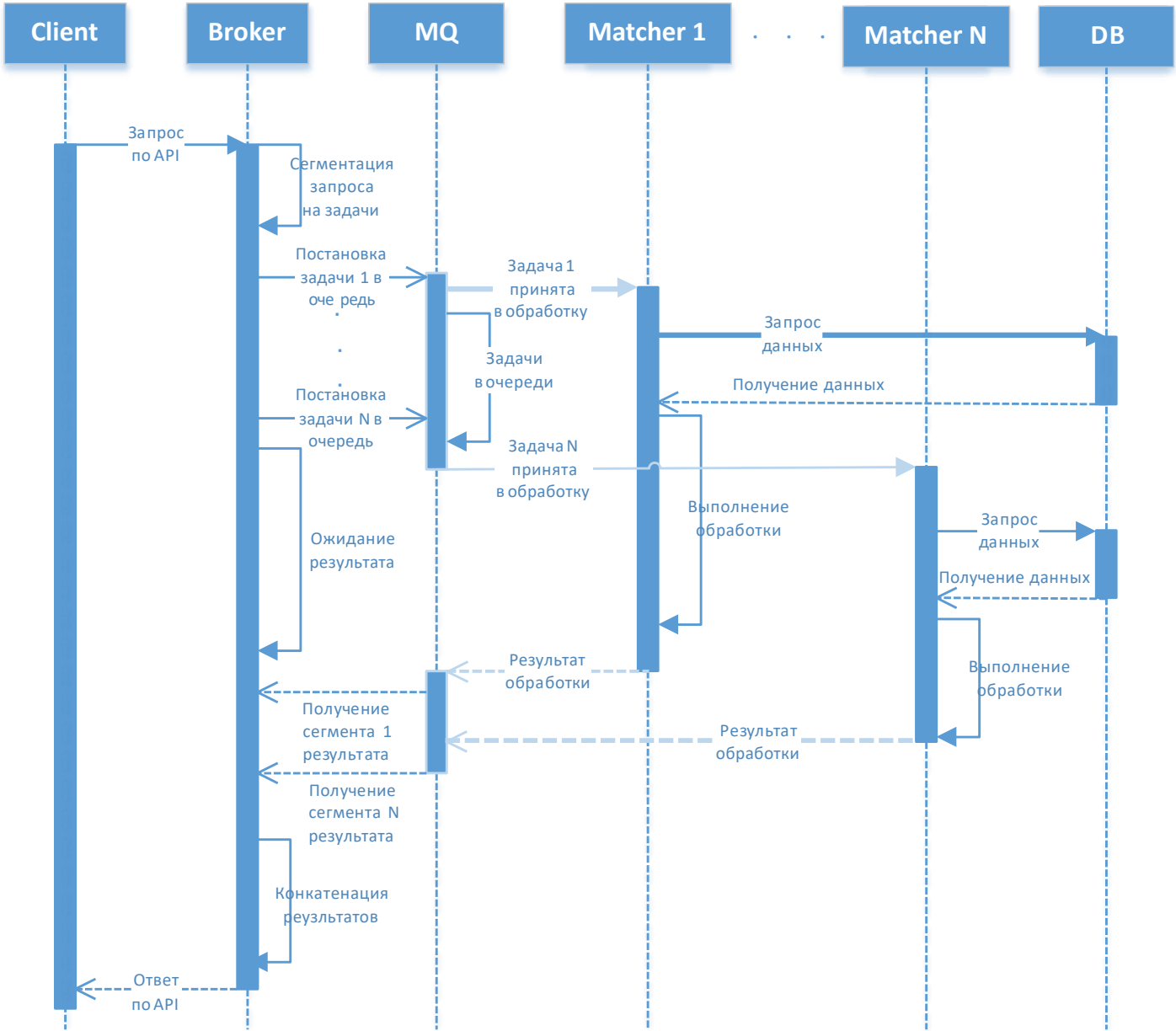
LUNA состоит из следующих компонентов:

- Broker – модуль, реализующий API;
- Extractor – обработчик, реализующий функцию извлечения биометрического шаблона из изображения;
- Matcher – обработчик, реализующий функцию сравнения биометрических шаблонов.
- RabbitMQ - Менеджер очередей сообщений (MQ);
- Aerospike - Система управления базами данных класса noSQL Key-value in memory DB.









Требования LUNA состоит из следующих компонентов:

- Центральный процессор
 - Наиболее затратная операция - извлечение дескриптора.
 - Сравнение дескрипторов не затратная операция.
- Требования к объёму оперативной памяти (RAM)
 - in memory база данных Aerospike (~4 млн записей в БД на гигабайт оперативной памяти).
 - объём загружаемых в Matcher дескрипторов для сравнения, параметры кэширования списков.
- Требования к подсистеме хранения
 - требований к объёму дискового пространства для постоянного хранения БД.
 - требований к быстродействию в случае in SSD DB.

HW sizing – количество запросов на extract исходя из нормативного времени от запроса до ответа 1 секунда.

Очередь обработки запросов. Запросы, пришедшие во время полной загрузки системы ставятся в очередь на исполнение => повышается время обработки запроса.

В стандартной инсталляции используется ОС Red Hat Enterprise Linux \ ОС CentOS версии > 7.1

Для лицензирования LUNA предусмотрены следующие возможности:

- Установка аппаратного USB ключа
 - на каждый HW сервер, (в случае с VM необходим "проброс" USB устройства в VM).
- Использование программного ключа
 - на каждый HW сервер с привязкой к:
 - ID процессора
 - ID чипсета материнской платы
 - MAC-адрес сетевой карты
 - на каждый VM сервер с привязкой к:
 - ID физического процессора хостовой машины
 - UUID виртуальной машины
 - MAC-адрес виртуальной сетевой карты
- Сетевой программный ключ с выделенным сервером лицензирования
 - привязка сервера лицензирования по варианту программного или аппаратного ключа.
 - сетевая связность, одна подсеть с broadcast.

Стандартная установка LUNA состоит из следующих шагов:

- ➊ Установка общесистемных пакетов – зависимостей
 - epel-release, yum install
- ➋ Установка СУБД Aerospike, с++ клиента
 - install aerospike-server-community
 - edit config
- ➌ Установка менеджера сообщений RabbitMQ
 - add user
 - set permissions
 - declare exchanges
 - declare queues
- ➍ Установка шины HASP Runtime
- ➎ Установка ПО LUNA
 - untar \ copy
 - edit configs
 - systemd *.service ; systemctl enable \ start

Стандартная настройка namespaces СУБД Aerospike

- ➊ Определить максимально планируемое количество хранимых в RAM БД дескрипторов
 - базово из расчёта ~4 млн записей дескрипторов (256 bytes) в БД на гигабайт оперативной памяти
 - прибавить 40% на нужды репликации\ дефрагментации (значение по умолчанию)
- ➋ Определить место хранения и размер файла постоянного хранения БД
 - размер файла должен быть минимум в 2 раза больше объёма выделенной RAM
- ➌ Определить время жизни данных (если применимо)
 - размер файла default-ttl, max-ttl
- ➍ Определить replication-factor (для работы в кластере БД)
 - количество копий данных в кластере

```
namespace luna_temporary {  
    replication-factor 1  
    memory-size 1G  
    default-ttl 1D  
    max-ttl 7D  
    storage-engine memory  
}
```

```
namespace luna_persistent {  
    replication-factor 1  
    memory-size 4G  
    default-ttl 0  
    storage-engine memory  
    storage-engine device {  
        file /opt/aerospike/data/luna.dat  
        filesize 8G  
        data-in-memory true  
    }  
}
```


Стандартная настройка менеджера сообщений RabbitMQ

➤ Определить пользователя, задать права и пароль LUNA

- `rabbitmqctl add_user luna luna`
- `rabbitmqctl set_permissions -p / luna ".*" ".*" ".*"`
- `rabbitmqctl set_user_tags luna administrator`

➤ Декларировать точки обмена и очереди, необходимые для работы компонентов LUNA

- Включить плагин управления:

`rabbitmq-plugins enable rabbitmq_management`

- Получить утилиту администрирования:

`wget http://127.0.0.1:15672/cli/rabbitmqadmin`

- настроить точки обмена и очереди:

`rabbitmqadmin -u luna -p luna declare exchange name=luna.extract type=direct`

`rabbitmqadmin -u luna -p luna declare exchange name=luna.match type=direct`

`rabbitmqadmin -u luna -p luna declare queue name=extractor durable=false auto_delete=true`

`rabbitmqadmin -u luna -p luna declare queue name=matcher durable=false auto_delete=true`

Настройка шины лицензирования HASP



Стандартная настройка шины лицензирования HASP

- Директория продукта /var/hasplm/
 - файл конфигурации hasplm.ini
 - файл лицензии .v2c
 - библиотека вендора haspvlib_111186.so
 - исполняемый файл hasp_update

Инсталлированные программные лицензии находятся в
/var/hasplm/installed/111186

systemctl status aksusbd.service

Команды:

./LicenseAssist fingerprint node_name.c2v

./hasp_update u License.v2c

```
hasplm.ini: [SERVER]
name = test-stand
pagerefresh = 3
linesperpage = 20
ACCremote = 1
enabledetach = 0
reservedseats = 0
reservedpercent = 0
detachmaxdays = 14
commuter_delete_days = 7
requestlog = 0
loglocal = 0
logremote = 0
logadmin = 0
errorlog = 0
rotatelog = 0
access_log_maxsize = 0 ;kB
error_log_maxsize = 0 ;kB
zip_logs_days = 0
delete_logs_days = 0
pidfile = 0
passacc = 0
accessfromremote = 1
accesstoremote = 1
bind_local_only = 0 ;1=localhost only
[REMOTE]
broadcastsearch = 1
aggressive = 0
serversearchinterval = 30
serveraddr = 127.0.0.1
[ACCESS]
allow=all
```

Пояснение по составу ПО

● Нейронная сеть CNN

- алгоритмы, позволяющие извлекать вектора признаков из лиц
- спроектированная, обученная и скомпилированная сеть
- директория /data , номер сети задаётся в faceengine.conf

● Библиотеки FSDK

- программная реализация методов работы с CNN
- программная реализация детектора лиц, оценки качества, etc.
- директория luna , пороги в faceengine.conf

● ПО LUNA

- логика работы приложения
- программная реализация внешних методов взаимодействия (API)
- программная реализация внутренних методов взаимодействия

faceengine.conf :

```
<?xml version="1.0"?>
<settings>
  <section name="system">
    <param name="cpuClass" type="Value::String" text="auto" />
  </section>
  <section name="flower">
    <param name="deviceClass" type="Value::String" text="cpu" />
    <param name="numThreads" type="Value::Int1" x="4" />
    <param name="verboseLogging" type="Value::Int1" x="0" />
    <param name="numComputeStreams" type="Value::Int1" x="1" />
  </section>
  <section name="DescriptorFactory::Settings">
    <param name="model" type="Value::Int1" x="46" />
    <param name="distance" type="Value::String" text="L2" />
  </section>
  <section name="MTCNNDetector::Settings">
    <param name="PNetThreshold" type="Value::Float1" x="0.6"/>
    <param name="RNetThreshold" type="Value::Float1" x="0.7"/>
    <param name="ONetThreshold" type="Value::Float1" x="0.93"/>
    ...
  </section>
  <section name="LSH::Settings"> ... </section>
  <section name="QualityEstimator::Settings"> ... </section>

</settings>
```

LUNA-CORE broker

"port": <порт> - tcp порт, на котором сервис будет ожидать http соединения;

"listenGlobally" <true> разрешает запуск на всех сетевых интерфейсах, иначе необходимо указать IP адрес на котором сервис будет ожидать http соединения;

"imageConstraints" определяет минимальные и максимальные размеры изображений в пикселях по каждой стороне;

"resultLimit": <Количество_результатов> - глобальная настройка количества кандидатов, возвращаемых как лучшие при сравнении дескрипторов.

"staticListPartitionSize": <количество списков> - настройка количества батчей списков сравнений, которое максимально может получить один процесс matcher в рамках одной задачи сравнения.

"individualPartitionSize": - настройка количества сравнений, которое максимально может получить один процесс matcher в рамках одной задачи сравнения без использования статических списков.

/var/lib/luna/conf/broker.conf:

```
{
  "include" : ["aerospike.conf"],
  "core" : {
    "port": 8083,
    "listenGlobally": true
  },
  "extractor" : {
    "include" : ["rabbitmq-extractor.conf"],
    "imageConstraints" : {
      "minSize": { "width": 96, "height": 96 },
      "maxSize": { "width": 4096, "height": 4096 }
    },
    "routes" : { "routingKey": "extractor" }
  },
  "matcher" : {
    "include" : ["rabbitmq-matcher.conf"],
    "routes" : { "routingKey": "matcher" },
    "resultLimit": 150,
    "staticListPartitionSize": 112,
    "individualPartitionSize": 15000
  }
}
```


LUNA-CORE extractor

/var/lib/luna/current/conf/extractor.conf :

```
{
  "include": ["aerospike.conf", "rabbitmq-extractor.conf"],
  "worker" : { "routingKey": "extractor" },

  "fsdk" : {
    "configPath": "/var/lib/luna/current/data/faceengine.conf",
    "dataPath": "/var/lib/luna/current/data"
  }
},
```

/var/lib/luna/current/conf/aerospike.conf :

```
{
  "db" : {
    "host": "localhost",
    "port": 3000,
    "user": "",
    "passwd": "",

    "namespaces" : {
      "persistent": "luna_persistent",
      "temporary": "luna_temporary"
    }
  }
}
```

/var/lib/luna/current/conf/rabbitmq-extractor.conf :

```
{
  "amqp" : {
    "host": "127.0.0.1",
    "version": [0,9,1],
    "exchange": "luna.extract",
    "user": "luna",
    "pass": "luna"
  }
}
```

LUNA-CORE matcher

/var/lib/luna/current/conf/matcher.conf :

```
{
  "include" : ["aerospike.conf", "rabbitmq-matcher.conf"],
  "worker" : { "routingKey": "matcher" },

  "fsdk" : {
    "configPath": "/var/lib/luna/current/data/faceengine.conf",
    "dataPath": "/var/lib/luna/current/data"
  },

  "cache" : { "capacity" : 2000  },

  "useLSH" : false
}
```

Cache capacity - ёмкость RAM, доступной matcher для кэширования списков
LSH - Locality-sensitive hashing - метод понижения размерности данных.

/var/lib/luna/current/conf/aerospike.conf :

```
{
  "db" : {
    "host": "localhost",
    "port": 3000,
    "user": "",
    "passwd": "",

    "namespaces" : {
      "persistent": "luna_persistent",
      "temporary": "luna_temporary"
    }
  }
}
```

/var/lib/luna/current/conf/rabbitmq-matcher.conf :

```
{
  "msg" : {
    "host": "localhost",
    "driver": "msgrabbitmqdrv",
    "exchange": "luna.match",
    "user": "luna",
    "pass": "luna"
  }
}
```

Стандартная установка LUNA содержит скрипт запуска компонент Extractor и Matcher

```
luna-extractor --config-path ../conf/extractor.conf --log-severity $VERBOSE --log-file /logs/extractor-$(i).log
```

```
luna-matcher --config-path ../conf/matcher.conf --num-threads $TH --log-severity $VERBOSE --log-file /logs/matcher-$(i).log
```

А также стандартные юнит файлы сервисов для systemd:

luna-api.service

luna-broker.service

luna-extractor.service

luna-matcher.service

luna-stat.service

luna-ui.service

[Unit]

Description=Luna Extractor workers

After=network.target aerospike.service rabbitmq-server.service

Requires=aerospike.service rabbitmq-server.service

[Service]

Type=forking

Environment=VERBOSE=2

Environment=WORKERS=6

SyslogIdentifier=luna-extractor

WorkingDirectory=/var/lib/luna/current/

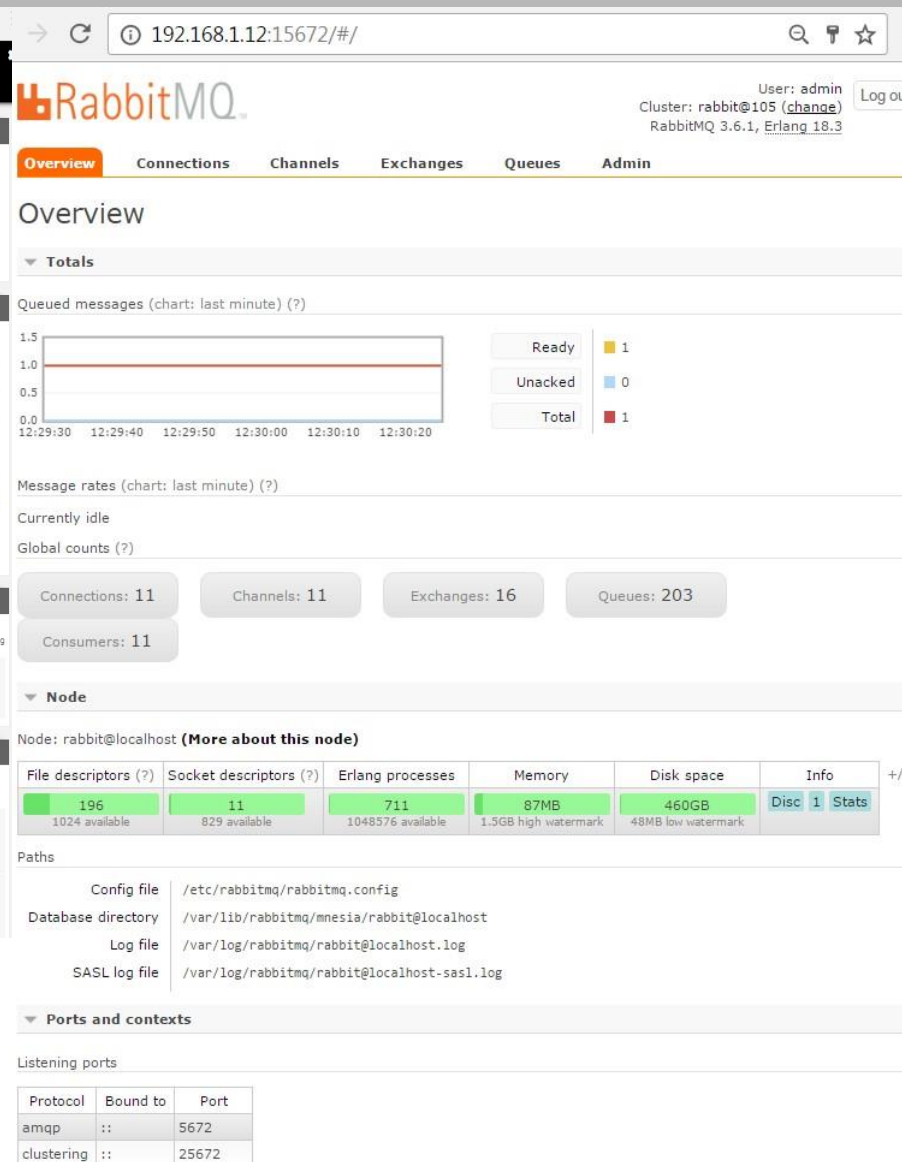
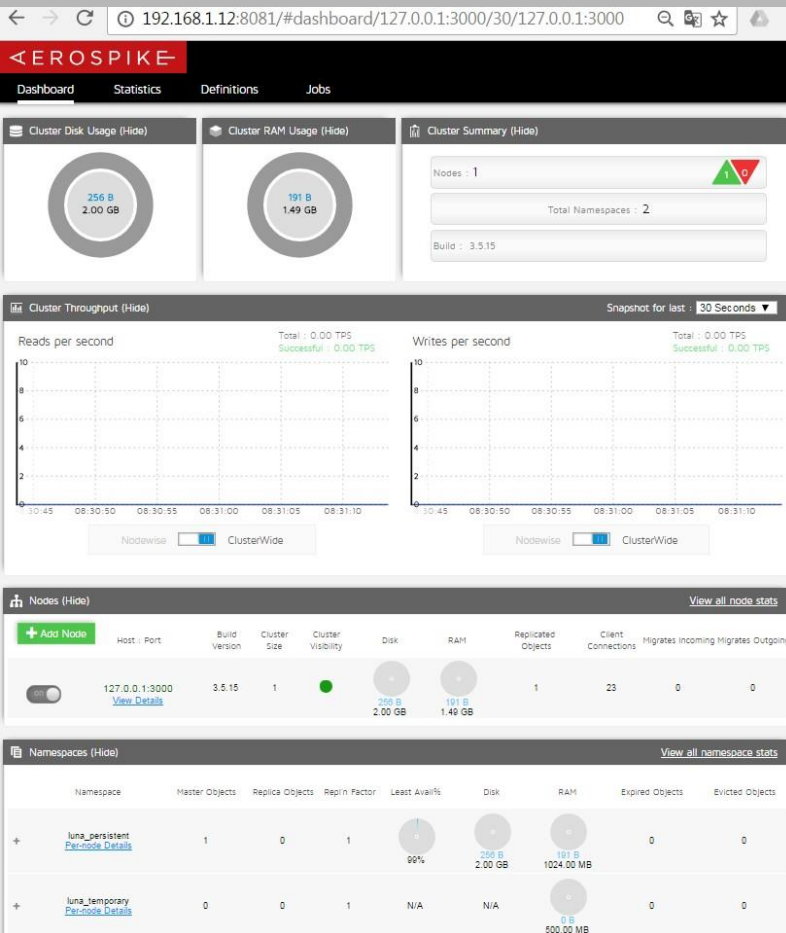
ExecStart=/var/lib/luna/current/bin/run-extractors.sh

\$WORKERS \$VERBOSE

[Install]

WantedBy=multi-user.target

Мониторинг компонентов LUNA: GUI интерфейсы



192.168.1.12:1947/_int_/config.htmr

SafeNet Sentinel Admin Control Center

Configuration for Sentinel License Manager on stand-0609

Basic Settings Users Access to Remote License Managers Access from Remote Clients Detachable Licenses Network

Machine Name stand-0609

Allow Remote Access to ACC ☒

Display Refresh Time 3 (seconds)

Table Rows per Page 20 (5 to 100)

Write an Access Log File ☐ Size Limit (KB): 0 (0: No limit) Edit Log Parameters

Include Local Requests ☐

Include Remote Requests ☐

Include Administration Requests ☐

Write an Error Log File ☐ Size Limit (KB): 0 (0: No limit)

Write Log Files Daily ☐

Days Before Compressing Log Files 0 (0: Never compress)

Days Before Deleting Log Files 0 (0: Never delete)

Write a Process ID (.pid) File ☐

Password Protection ☒ Configuration Pages ☐ All ACC Pages Change Password

Submit Cancel Set Defaults

rabbitmq-plugins enable rabbitmq_management

```
rabbitmqctl status  
{alarms,[____]},
```

```
/var/log/rabbitmq/rabbit@hostname.log
```

```
rabbitmqadmin -H 127.0.0.1 -V / list exchanges
```

```
journalctl -u LunaExtractor.service
```

```
journalctl -u aksusbd.service --since="2017-01-12 17:17:00" --until="2017-01-12 17:17:06"
```

--since="2017-01-12 17:17:00" дата и время, начиная с которого выводить информацию
--until="2017-01-13 18:17:06" дата и время, до которого выводить информацию

```
asinfo -v 'get-config '      asadmin
```

```
asmonitor -e "asinfo -v 'set-config:context=namespace;id=luna_persistent;memory-size=8G'"
```

Admin> info node

```
2 hosts in cluster: 10.17.78.102:3000,10.17.78.106:3000
```

```
Aerospike Interactive Shell, version 3.7.3
```

```
Monitor> info node
```

```
===NODES===
```

```
2017-05-12 22:13:47.347809
```

```
Sorting by IP, in Ascending order:
```

ip:port	Build	Cluster	Cluster	Free	Free	Migrates	Node	Principal	Replicated	Sys
	.	Size	Visibility	Disk	Mem	.	ID	ID	Objects	Free
	.	.	.	pct	pct	Mem
vislabs-node3-test.hq.bc:3000	3.7.3	2	true	73	50	(0,0)	BB9310F81565000	BB9642381565000	77005094	48
vislabs-node4-test.hq.bc:3000	3.7.3	2	true	73	50	(0,0)	BB9642381565000	BB9642381565000	77005094	54

Number of nodes displayed: 2

```
===NAMESPACE===
```

```
Total (unique) objects in cluster for luna_persistent : 77005094
```

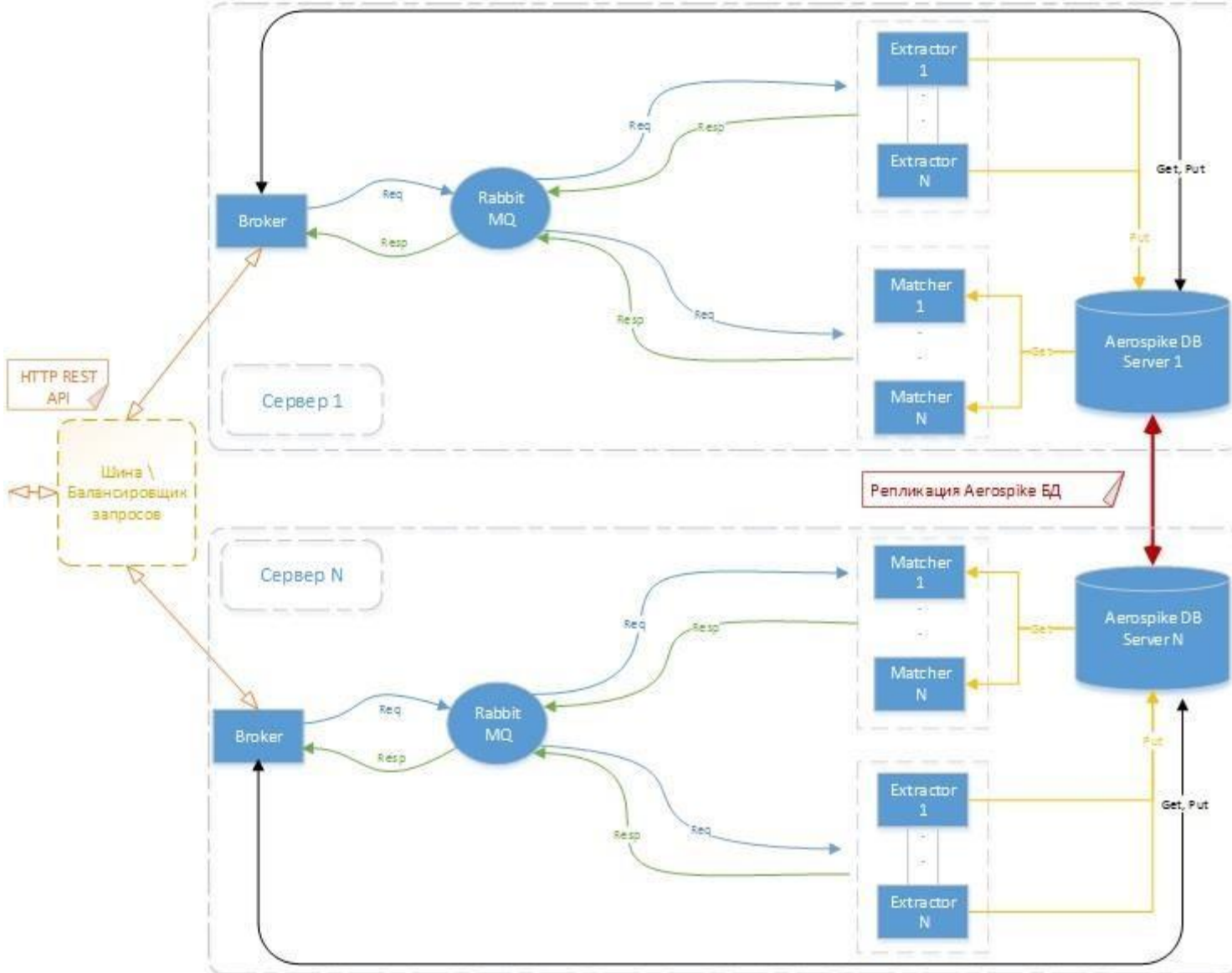
```
Total (unique) objects in cluster for luna_temporary : 0
```

```
Note: Total (unique) objects is an under estimate if migrations are in progress.
```

ip/namespace	Avail	Evicted	Master	Repl	Stop	Used	Used	Used	Used	hwm	hwm
	Pct	Objects	Objects	Factor	Writes	Disk	Disk	Mem	Mem	Disk	Mem
	%	.	%	.	.
vislabs-node3-test.hq.bc/luna_persistent	73	0	39555375	2	false	55.08 G	27	52.42 G	50	90	90
vislabs-node4-test.hq.bc/luna_persistent	73	0	37449719	2	false	55.08 G	27	52.42 G	50	90	90
vislabs-node3-test.hq.bc/luna_temporary	n/a	0	0	1	false	n/a	n/a	0.00 B	0	50	60
vislabs-node4-test.hq.bc/luna_temporary	n/a	0	0	1	false	n/a	n/a	0.00 B	0	50	60

Number of rows displayed: 4

Масштабирование ПО LUNA и репликация данных осуществляется на уровне СУБД Aerospike.



```
heartbeat { mode multicast
address 239.1.99.222
port 9918
interval 150
timeout 10 }
```

```
=====
heartbeat { mode mesh
address _SERVER1_IP
port 3002
```

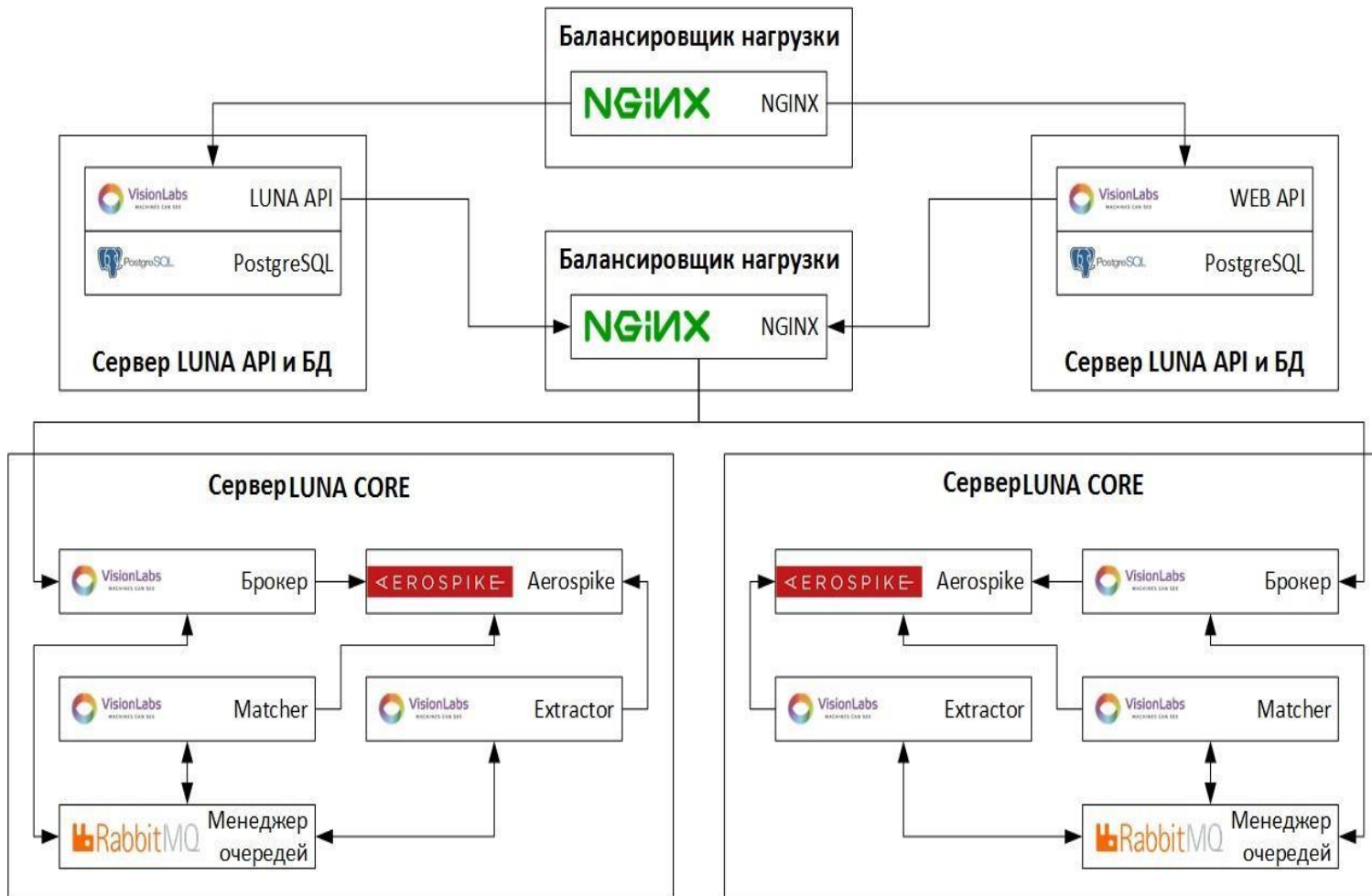
```
mesh-seed-address-port SERVER1_IP 3002
mesh-seed-address-port SERVER2_IP 3002
interval 150
timeout 10
}
```

- Определить планируемое количество нод БД
 - определить количество копий данных (replication-factor)
 - работающий по умолчанию sharding делит все данные на N нод
- Определить размер данных в RAM на каждой из нод
 - исходя из объёма данных, количества нод, количества копий, возможных аварий.
- Определить какой размер кластера считается критически малым (paxos-single-replica-limit)
 - исходя из какого количества нод сократить до единственной количество копий данных.
- Определить и настроить сетевое взаимодействие нод
 - определить тип сетевого взаимодействия (multicast\mesh)
 - задать адреса и порты взаимодействия

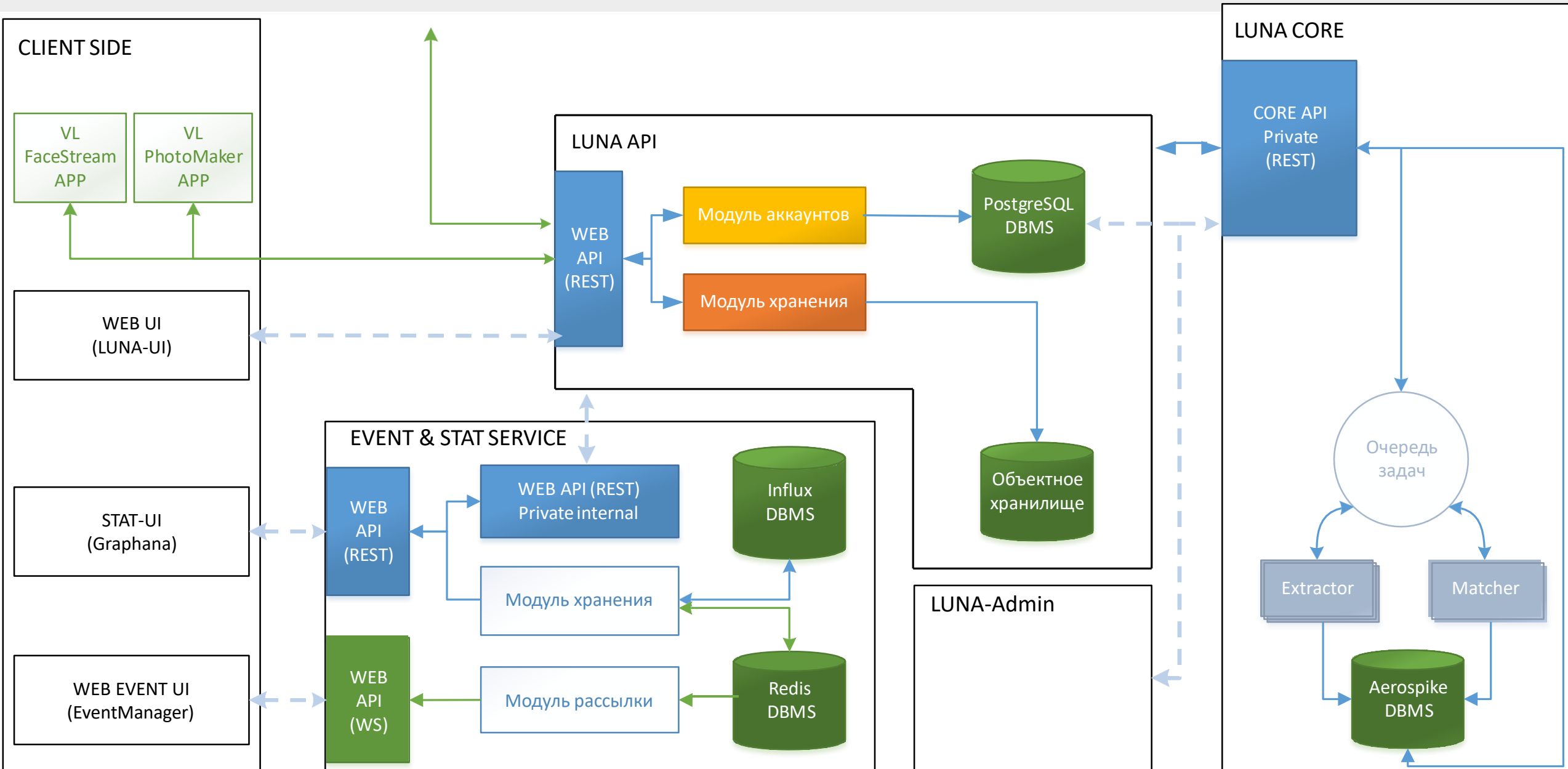
FailSafe конфигурация LUNA API

FailSafe конфигурация LUNA API подразумевает наличие двух нод LUNA API, которые работают с внешней по отношению к ним реляционной БД Postgres.

В случае падения одной из нод внешний балансировщик (например nginx) переключат поток данных на вторую ноду.



Общая схема LUNA



Тел: 8 029 794-49-33

E-mail: info@ais4life.com

Web: <https://ais4life.com>