

BI SOLUTIONS

MANAGING DATA

Руководство по эксплуатации

BIS Platform

Москва, 2022

Настоящий документ описывает порядок установки и настройки специального программного обеспечения для ансамблирования биометрических алгоритмов обнаружения витальности – BIS Platform. Документ предназначен для администраторов, осуществляющих установку и настройку BIS Platform.

Изготовитель	ООО «Би Ай Солюшнс»
Адрес	105122, Москва, Щёлковское шоссе, 5 строение 1
Техническая поддержка	+7 (985) 917-67-45, ataranov@bisolut.ru

ОГЛАВЛЕНИЕ

1 Общие сведения.....	3
2 Требования к аппаратному обеспечению.....	3
3 Требования к системному программному обеспечению.....	3
4 Архитектура.....	4
5 Порядок установки.....	5
6 Порядок настройки.....	6
6.1 Подключение/отключение БП.....	6
6.2 Выбор настроек ансемблирования.....	7
7 Обнаружение витальности.....	9
7.1 Примеры запросов.....	10
7.1.1 Лицо, passive-instructions.....	10
7.1.2 Видео лица с записью голоса, text-instructions.....	11
7.1.3 Голос, text-instructions.....	12
7.1.4 По 816-му приказу.....	13

1 Общие сведения

Специальное программное обеспечение BIS Platform liveness (далее – СПО) предназначено для ансамблирования биометрических алгоритмов обнаружения витальности. Алгоритмы обнаружения витальности не входят в состав СПО.

Обнаружение витальности – измерение и анализ анатомических характеристик, произвольных или произвольных реакций индивида с целью определения того, что биометрический образец получен от живого индивида (ГОСТ Р 58624.1 «Информационные технологии. Биометрия. Обнаружение атаки на биометрическое предъявление»)

СПО предоставляет клиентам HTTP API, совместимое с требованиями методических рекомендаций по подключению вендоров к Единой биометрической системе (далее – МР ЕБС) в части требований к биометрическим процессорам (далее – БП) обнаружения витальности.

2 Требования к аппаратному обеспечению

Минимальные требования для запуска 1-го экземпляра СПО:

- процессор (x86-64) – не менее 1 ядра, не менее 1 ГГц
- оперативная память – не менее 512 Мбайт
- свободное место на диске – не менее 256 Мбайт

Рекомендуемые системные требования для запуска 1-го экземпляра СПО:

- процессор (x86-64) – не менее 2 ядер, не менее 2 ГГц
- оперативная память – не менее 1 Гбайт
- свободное место на диске – не менее 512 Мбайт

3 Требования к системному программному обеспечению

Для запуска СПО требуется следующее системное программное обеспечение:

- операционная система, поддерживающая запуск Docker:
 - Linux Ubuntu 20.04 или выше
 - Linux CentOS-7 или выше
 - MS Windows 10 или выше
- [Docker](#) 20.10.10 или выше
- [Docker Compose](#) 1.29.0 или выше

4 Архитектура

СПО работает как прокси-сервер, асинхронно транслирующий запросы клиента к API обнаружения витальности в формате МР ЕБС в формат запроса конкретного БП. Далее, после получения ответов от всех задействованных в обработке запроса БП-ров и их агрегации, клиенту высылается ответ в совместимом с требованиями МР ЕБС виде.

На рисунке 1 приведён пример архитектуры информационной системы, для работы внутри которой предназначено СПО. Размещение СПО на одном с БП-рами вычислительном сервере, при достаточном количестве аппаратных ресурсов, обеспечит минимальное время обработки запроса обнаружения витальности за счёт сокращения временных задержек на передачу биометрических образцов. Тем не менее, при необходимости, БП-ры могут быть развёрнуты на удалённых серверах.

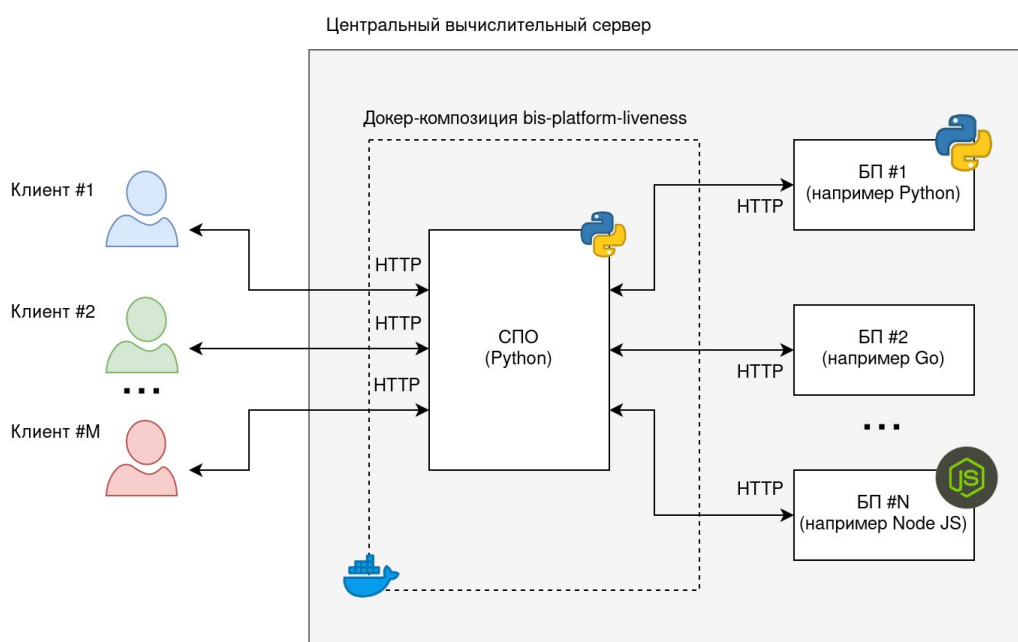


Рисунок 1 – типовая схема построения решений с применением СПО в закрытом информационном контуре

Не следует подключать к СПО несколько разных экземпляров одного БП. Хотя это не приведёт к ошибкам, но, на практике, не имеет никакого смысла. СПО выполняет функцию ансамблирования БП-ров, а не функцию распределения нагрузки. Каждый запрос на обнаружение витальности от клиента отправляется параллельно во все подключенные БП (соответствующие типу биометрического образца и биометрической модальности запроса).

СПО не умеет работать в режиме HTTPS-сервера. Поэтому, при необходимости вести обмен с клиентами по зашифрованному каналу, требуется проксировать запросы к СПО через отдельный TLS Termination прокси-сервер. Схема такой архитектуры приведена на рисунке 2.

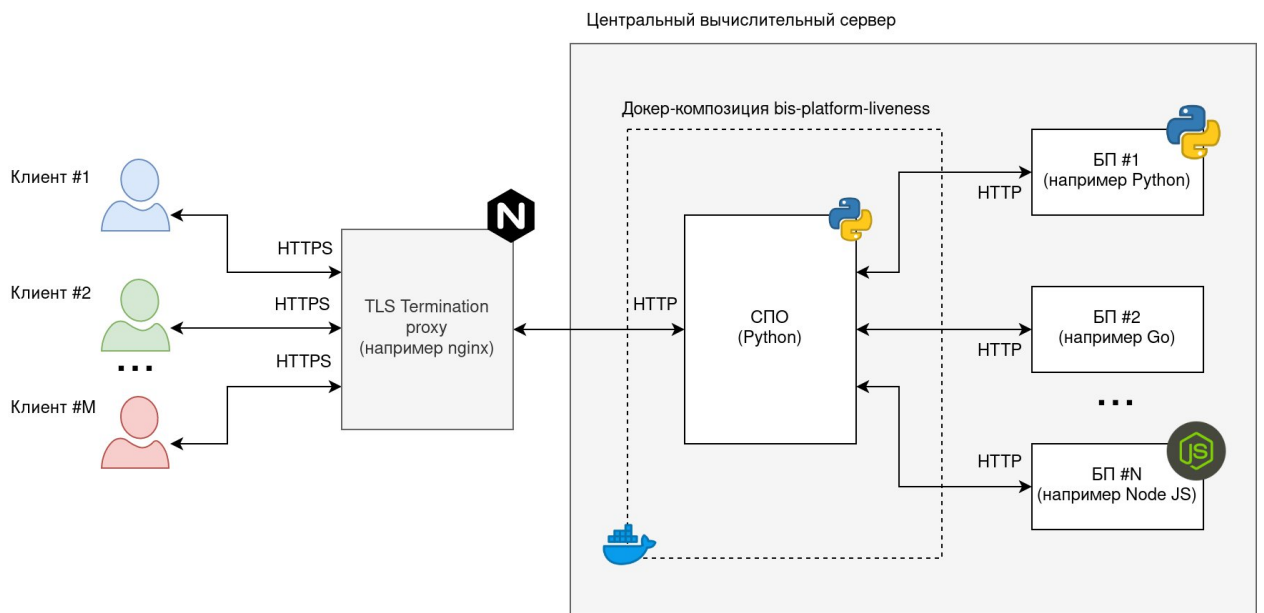


Рисунок 2 – типовая схема построения решений с применением СПО в открытом информационном контуре

При необходимости увеличения средней скорости обработки запросов, следует применять балансировщики нагрузки. Все экземпляры одного БП следует разместить за одним балансировщиком. Затем, подключить балансировщик к СПО вместо соответствующего БП. Схема такой архитектуры приведена на рисунке 3.

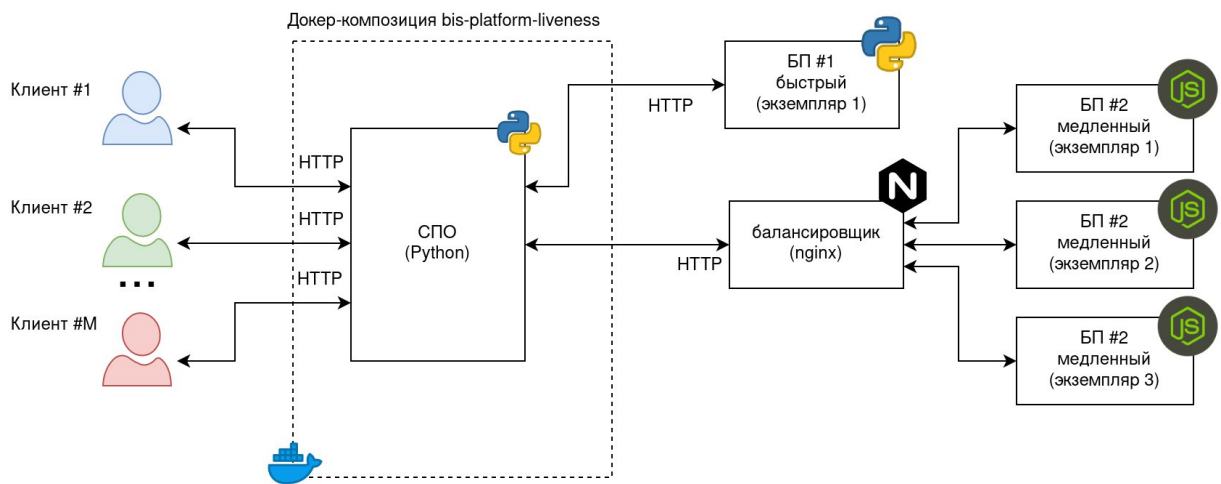


Рисунок 3 – схема правильного способа увеличения средней скорости обработки запросов

5 Порядок установки

Распаковать архив с дистрибутивом СПО на локальный диск:

```
unzip bis-platform-liveness.zip
```

Загрузить докер-образ в локальное хранилище:

```
docker load -i bis-platform-liveness.tar.gz
```

Опционально, удалить архивы для увеличения свободного пространства на диске:

```
rm bis-platform-liveness*
```

Запустить докер-композицию:

```
docker-compose up -d
```

Дождаться завершения запуска композиции и проверить работу сервера:

```
curl -v http://localhost:8888/health
```

В ответ должен прийти HTTP статус 200 OK и json объект, содержимое которого зависит от настройки БП-ров.

В качестве дополнительной проверки работы СПО, можно перейти в браузере по ссылке <http://localhost:8888/docs> (если СПО развёрнуто локально и проброс портов в файле `bis-platform-liveness/docker-compose.yml` не редактировался). При успешном старте СПО, по ссылке должна открыться интерактивная html-страница с документацией API СПО в формате swagger. Для рендеринга html-страницы требуется доступ к сети интернет.

6 Порядок настройки

Настройка СПО заключается в указании URL-адресов и весов БП-ров и выборе настроек ансамблирования.

6.1 Подключение/отключение БП

Запрос списка известных СПО БП-ров:

```
curl -v http://localhost:8888/processors
```

В полученном списке, по значениям параметров «vendor», «version» и «modality» следует найти кодовое обозначение «name» БП-ра, который требуется подключить. Например, пассивный лицевой лайвнесс от Би Ай Солюшнс (не входит в состав СПО), имеет кодовое обозначение «ares». Предположим, экземпляр соответствующего БП-ра развёрнут на сервере с адресом 192.168.0.20 и его API работает на порте 5000 с префиксом /liveness. Для подключения этого БП-ра к СПО требуется в API СПО отправить запрос:

```
curl -X PATCH http://localhost:8888/processors?  
name=ares&weight=1.0&url=http://192.168.0.20:5000/liveness
```

После успешного подключения БП-ра, в json-объекте ответа, значение параметра «enabled» должно быть установлено в true. После подключения, запросы GET /health и POST /detect к API СПО начнут проксироваться к соответствующим эндпоинтам БП-ра. И так для всех подключённых БП-ров.

Для отключения БП-ра следует в запросе PATCH /processors указать, для отключаемого БП, значение «weight» равным нулю, или значение «url» равное пустой строке:

```
curl -X PATCH http://localhost:8888/processors?
name=ares&weight=0&url=
```

6.2 Выбор настроек ансамблирования

Запрос настроек ансамблирования:

```
curl -v http://localhost:8888/settings
```

В ответ СПО пришлёт json-объект со следующими параметрами:

Таблица 1 – Перечень настроек ансамблирования

№	Параметр	Назначение	Допустимые значения	Расшифровка
1	attack_instrument	инструмент атаки, на борьбу с которым настроено СПО	all	все инструменты атаки
			print	фото напечатанное на бумаге
			replay	изображение лица/ проигрывание видео на экране
2	work_mode	режим выбора БП-ров	multiple_weighted	отправлять запрос обнаружения витальности всем подходящим БП
			single_fastest	отправлять запрос обнаружения витальности одному, самому быстрому БП
			single_most_accurate	отправлять запрос обнаружения витальности одному, самому точному БП
3	aggregation_type	метод взвешивания скоров	arithmetic	арифметическое среднее
			geometric	геометрическое среднее
			harmonic	гармоническое среднее
			median	медиана
			max	максимум
			min	минимум

№	Параметр	Назначение	Допустимые значения	Расшифровка
4	passing_policy	способ принятия решения	score_threshold	сравнение взвешенного сора с пороговым значением, – если скор выше порога, то предъявление признаётся подлинным
			simple_voting_on_decisions	простое голосование по бинарным решениям БП-ров, – если больше половины считают предъявление подлинным, то предъявление признаётся подлинным
			solid_voting_on_decisions	единогласное голосование по бинарным решениям БП-ров, – предъявление признаётся подлинным только если все БП-ры считают предъявление подлинным
			probs_comparison	сравнение взвешенных оценок ВОКПА и ВОКПБП, – если ВОКПБП больше ВОКПА, то предъявление признаётся подлинным
			simple_voting_on_threshold	простое голосование по результатам сравнения скоров с порогом, – если больше половины БП-ров возвращают скор выше порога, то предъявление признаётся подлинным
			solid_voting_on_threshold	удиногласное голосование по результатам сравнения скоров с порогом, – если все БП-ры возвращают скор выше порога, то предъявление признаётся подлинным
5	score_threshold	пороговое значение сора	0.0 .. 1.0	величина с которой сравнивается скор для принятия решения по обнаружению витальности

Примечания:

ВОКПА – вероятность ошибки классификации предъявления атаки

ВОКПБП – вероятность ошибки классификации подлинного биометрического предъявления

Запрос PATCH /settings предназначен для установки желаемых настроек:

```
curl -X PATCH http://localhost:8888/settings?
    attack_instrument=all&
    work_mode=multiple_weighted&
    aggregation_type=arithmetic&
    passing_policy=solid_voting_on_threshold&
    score_threshold=0.5
```

Значения параметров, указанных в запросе выше, являются рекомендованными для продуктовой среды. Принятие решения методом единогласного голосования обеспечивает минимально возможные значения ВОКПА для ансамбля, а управление величиной «score_threshold» позволяет регулировать величину ВОКППП.

7 Обнаружение витальности

Для обнаружения витальности биометрического образца «bio_sample» с метаданными «metadata» предназначен запрос POST /detect:

```
curl -X POST -F bio_sample=@photo.jpg
    -F metadata=@passive_instructions.json
    http://localhost:8888/detect
```

На основании Content-Type части «bio_sample» и содержимого «metadata» СПО выбирает из списка подключённых БП-ров подходящие для обработки запроса. Формат входных данных и формат ответа на запрос соответствуют требованиям методических рекомендаций по подключению вендоров к Единой биометрической системе (далее – МР ЕБС) в части требований к биометрическим процессорам (далее – БП) обнаружения витальности.

Если все подключённые БП-ры, отобранные СПО для обнаружения витальности, отвечают статусом 200 ОК, то формируется агрегированный ответ со статусом 200 ОК и json-объектом с полями, назначение которых указано в таблице 2. Если хотя бы один из БП-ров пришлёт статус отличный от 200 ОК или не ответит на запрос, то СПО пришлёт в ответ статус 400 BAD_REQUEST или 500 SERVER_ERROR и json-объект с описанием ошибки. Также, если СПО не удастся найти среди подключённых БП-ров необходимых для обработки запроса, СПО также ответит статусом 400 BAD_REQUEST и json-объектом с справочной информацией о причине отказа.

Таблица 2 – выходные параметры метода POST /detect

№	Наименование	Тип данных	Описание
1	passed	bool	результат проверки bio_sample на витальность

№	Наименование	Тип данных	Описание
2	score	float	количественная оценка результата проверки bio_sample на витальность (0.0 - подделка, 1.0 - подлинное предъявление)
3	details	json-массив	результаты проверки на витальность по каждому из участвовавших в проверке БП
4	settings	json-объект	копия настроек СПО, которые применялись при выполнении запроса

7.1 Примеры запросов

7.1.1 Лицо, passive-instructions

```

POST /detect HTTP/1.1
Host: localhost:8888
Content-Length: ...
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW

----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="bio_sample"; filename="photo.jpg"
Content-Type: image/jpeg

(data)
----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="metadata"; filename="passive-instructions.json"
Content-Type: application/json

{
  "mnemonic": "passive-instructions",
  "actions": [
    {
      "type": "photo-type",
      "duration": 0,
      "message": "Посмотрите ровно в камеру"
    }
  ]
}
----WebKitFormBoundary7MA4YWxkTrZu0gW

```

Пример ответа 200 ОК:

```

{
  "passed": true,
  "score": 0.8464555780221319,
  "details": [
    {
      "name": "ares",
      "http_status": 200,
      "weight": 1.0,
      "score": 0.9993751645088196,

```

```

    "passed": true,
    "APCER": 0.3040871116018226,
    "BPCER": 0.07302183611176333,
    "message": null
  },
  {
    "name": "triton",
    "http_status": 200,
    "weight": 1.0,
    "score": 0.693536,
    "passed": true,
    "APCER": 0.09362946932142573,
    "BPCER": 0.1292820272172689,
    "message": null
  }
],
"settings": {
  "attack_instrument": "all",
  "work_mode": "multiple_weighted",
  "aggregation_type": "arithmetic",
  "passing_policy": "solid_voting_on_threshold",
  "score_threshold": 0.5
}
}

```

Пример ответа 500 SERVER_ERROR:

```

{
  "message": "Can not get reply from one or more bio-processors!",
  "details": [
    {
      "passed": true,
      "score": 0.9993751645088196,
      "BPCER": 0.07302183611176333,
      "APCER": 0.3040871116018226,
      "weight": 1.0,
      "name": "ares",
      "http_status": 200
    },
    {
      "name": "triton",
      "http_status": 0
    }
  ]
}

```

7.1.2 Видео лица с записью голоса, text-instructions

```

POST /detect HTTP/1.1
Host: localhost:8888
Content-Length: ...
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW

----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="bio_sample"; filename="video.mp4"
Content-Type: video/mp4

(data)
----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="metadata"; filename="text-instructions.json"
Content-Type: application/json

{

```

```

    "mnemonic": "text-instructions",
    "actions": [
      {
        "type": "numbers-digits",
        "duration": 5000,
        "message": "Произнесите цифры:",
        "text": "ноль семь пять четыре три"
      }
    ]
  }
  ----WebKitFormBoundary7MA4YWxkTrZu0gW

```

Пример ответа 200 ОК:

```

{
  "passed": true,
  "score": 0.9261,
  "details": [
    {
      "name": "cepheus",
      "http_status": 200,
      "weight": 1.0,
      "score": 0.9261,
      "passed": true,
      "APCER": 0.5421686746987951,
      "BPCER": 0.21578099838969403,
      "message": null
    }
  ],
  "settings": {
    "attack_instrument": "all",
    "work_mode": "multiple_weighted",
    "aggregation_type": "arithmetic",
    "passing_policy": "solid_voting_on_threshold",
    "score_threshold": 0.5
  }
}

```

Пример ответа 400 BAD_REQUEST:

```

{
  "message": "Не удалось прочитать биометрический образец",
  "code": "LDE-002004",
  "details": [
    {
      "code": "LDE-002004",
      "message": "Не удалось прочитать биометрический образец",
      "weight": 1.0,
      "name": "cepheus",
      "http_status": 400
    }
  ]
}

```

7.1.3 Голос, text-instructions

```

POST /detect HTTP/1.1
Host: localhost:8888
Content-Length: ...
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW

----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="bio_sample"; filename="audio.wav"
Content-Type: audio/wav

```

```
(data)
----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="metadata"; filename="text-instructions.json"
Content-Type: application/json

{
  "mnemonic": "text-instructions",
  "actions": [
    {
      "type": "numbers-digits",
      "duration": 5000,
      "message": "Произнесите цифры:",
      "text": "четыре шесть восемь два пять"
    }
  ]
}
----WebKitFormBoundary7MA4YWxkTrZu0gW
```

Пример ответа 200 ОК:

```
{
  "passed": false,
  "score": 0.23839681,
  "details": [
    {
      "name": "iridium",
      "http_status": 200,
      "weight": 1.0,
      "score": 0.23839681,
      "passed": false,
      "APCER": 0.046228710462287104,
      "BPCER": 0.12451737451737452,
      "message": null
    }
  ],
  "settings": {
    "attack_instrument": "all",
    "work_mode": "multiple_weighted",
    "aggregation_type": "arithmetic",
    "passing_policy": "solid_voting_on_threshold",
    "score_threshold": 0.5
  }
}
```

7.1.4 По 816-му приказу

В СПО предусмотрен особый модификатор запроса POST /detect, – хедер «Work-Pipeline» со значением 816. Наличие этого модификатора в запросе заставляет СПО применить к обработке запроса на обнаружение витальности требования приказа Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации от 06.08.2021 № 816 "Об утверждении методик проверки соответствия предоставленных биометрических персональных данных физического лица его биометрическим персональным данным, содержащимся в информационных системах, обеспечивающих идентификацию и (или) аутентификацию с использованием биометрических персональных данных, а также об определении степени взаимного соответствия указанных биометрических персональных данных, достаточной для

проведения идентификации, предусмотренной Федеральным законом от 27 июля 2006 г. № 149-ФЗ "Об информации, информационных технологиях и о защите информации".

Фактически, добавление хедера «Work-Pipeline» со значением 816 означает, что мультимодальное биометрическое предъявление должно быть разделено на отдельные модальности. Затем, запросы на обнаружение витальности должны быть отправлены в соответствующие независимые БП-ры. Затем, результаты проверки БР-ми должны быть объединены методом единогласного голосования.

```

POST /detect HTTP/1.1
Host: localhost:8888
Work-Pipeline: 816
Content-Length: ...
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW

----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="bio_sample"; filename="video.mp4"
Content-Type: video/mp4

(data)
----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="metadata"; filename="text-instructions.json"
Content-Type: application/json

{
  "mnemonic": "text-instructions",
  "actions": [
    {
      "type": "numbers-digits",
      "duration": 5000,
      "message": "Произнесите цифры:",
      "text": "четыре шесть восемь два пять"
    }
  ]
}

----WebKitFormBoundary7MA4YWxkTrZu0gW

```

Пример ответа 200 ОК:

```

{
  "passed": true,
  "score": 0.9079945969733514,
  "details": [
    {
      "name": "cepheus",
      "http_status": 200,
      "weight": 1.0,
      "score": 0.9747,
      "passed": true,
      "APCER": 0.1355421686746988,
      "BPCER": 0.6457326892109501,
      "message": null
    },
    {
      "name": "iridium",
      "http_status": 200,
      "weight": 1.0,
      "score": 0.9505638,
      "passed": true,
      "APCER": 0.0036496350364963502,
      "BPCER": 0.6640926640926641,
    }
  ]
}

```

```
"message": null
},
{
  "name": "triton",
  "http_status": 200,
  "weight": 1.0,
  "score": 0.79872,
  "passed": true,
  "APCER": 0.03196095667223698,
  "BPCER": 0.28742374472078835,
  "message": null
}
],
"settings": {
  "attack_instrument": "all",
  "work_mode": "multiple_weighted",
  "aggregation_type": "arithmetic",
  "passing_policy": "solid_voting_on_threshold",
  "score_threshold": 0.5
}
}
```

Если СПО не обнаружит подключённых БП разных модальностей, то ответит ошибкой 400 BAD_REQUEST. Например, если нет подключённых БП для голоса:

```
{
  "code": "LDE-100006",
  "message": "No processors for ['passive_voice'] have been found!"
}
```