

**Состав группы проекта:** Голихин Иван Сергеевич и Суханов Артемий Петрович

**Ментор** — Смирнова Елена Валентиновна, преподаватель кафедры ИУ6 МГТУ им. Н.Э. Баумана

**Трекер** — Цаплина Ольга Сергеевна, старший преподаватель ИБМ7, ИУ12 МГТУ им. Н.Э. Баумана

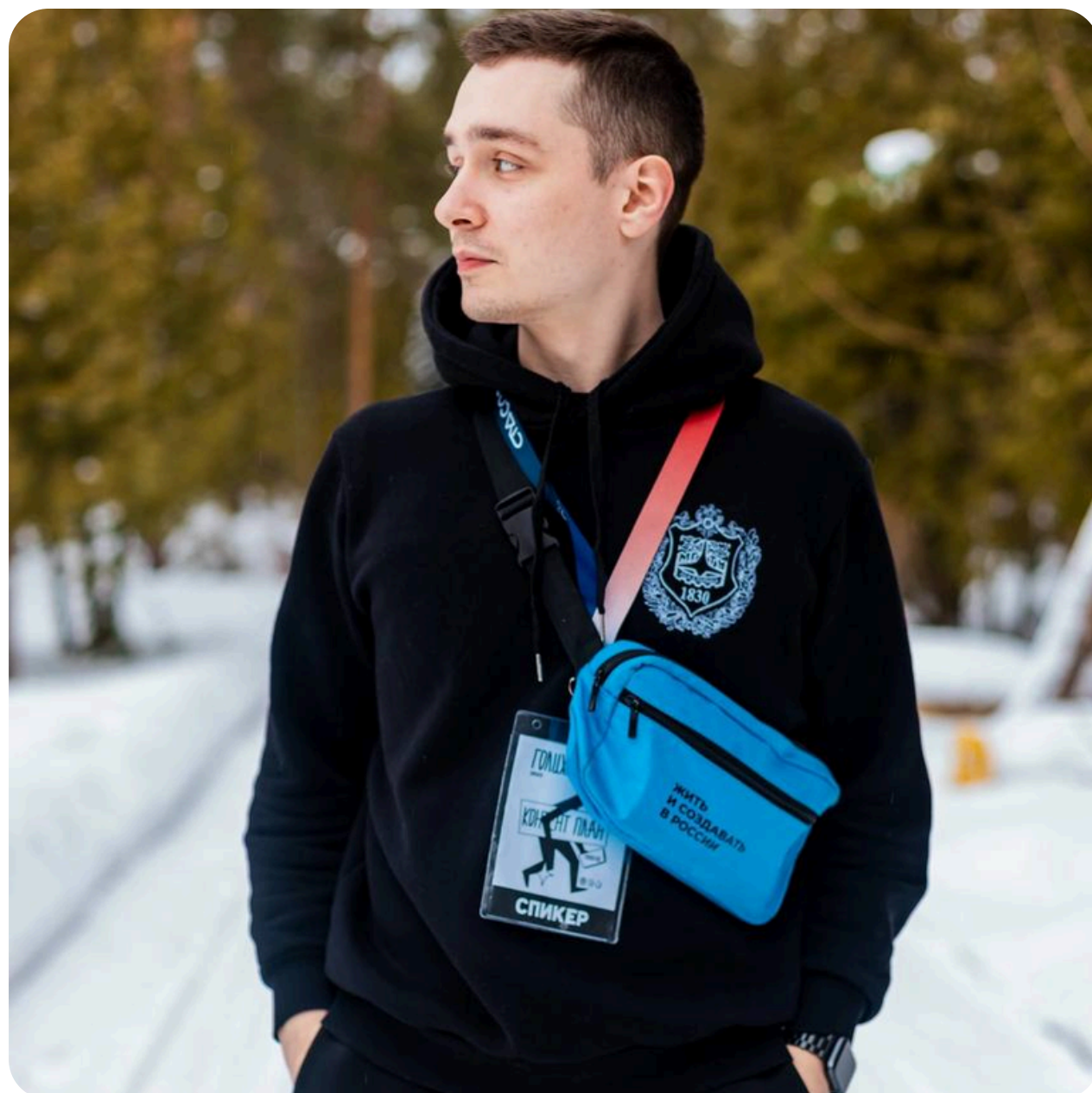


# TurboLët

**FAST 4 PEOPLE — БЫСТРО ДЛЯ ЛЮДЕЙ**

Проект TurboLët – разработка БПЛА-решений для транспортировки биологических образцов, в том числе в целях содействия правосудию, эффективной работе медицинских подразделений, лабораторий, а также частных коммерческих компаний в сфере логистики, медицины и высоких технологий.

Команда



## Голихин Иван, лидер проекта

Студент 5-го курса кафедры ЮР  
МГТУ им. Н.Э. Баумана.

PR-менеджер, робототехник-любитель,  
конструктор микропроцессоров, разработчик  
Python, Java, JS, UX/UI-дизайнер Figma



## Суханов Артемий, product manager

Студент 5-го курса кафедры ЮР  
МГТУ им. Н.Э. Баумана.

Product manager, инженер-  
проектировщик, разработчик Python, C++,  
C#, 3D-дизайнер Blender



Проблема

# Проблема

Важность поддержки инициатив в сфере БПЛА подтверждает высказывание Президента РФ Путина В.В., который в своем Послании Федеральному Собранию Государственной Думы России от 29.02.2024 указал на необходимость достижения технологического суверенитета в сквозных сферах, в том числе в области робототехники и беспилотных систем.

## СКОРОСТЬ и БЕЗОПАСНОСТЬ

Современные решения в области доставки биоматериалов не обеспечивают безопасной и быстрой доставки объектов конечному получателю.

Биоматериалы являются хрупкими и скоропортящимися продуктами, таким образом их транспортировка должна быть быстрая, вне зависимости от внешних факторов.

## ВРЕМЯ

Быстрая доставка биоматериалов позволит сократить время транспортировки необходимых объектов.

Так как множество биоматериалов, исходя из природы, быстро теряют свою пригодность для дальнейшего использования, время, затрачиваемое на их перевозку, должно быть максимально коротким.

## ДОСТУПНОСТЬ

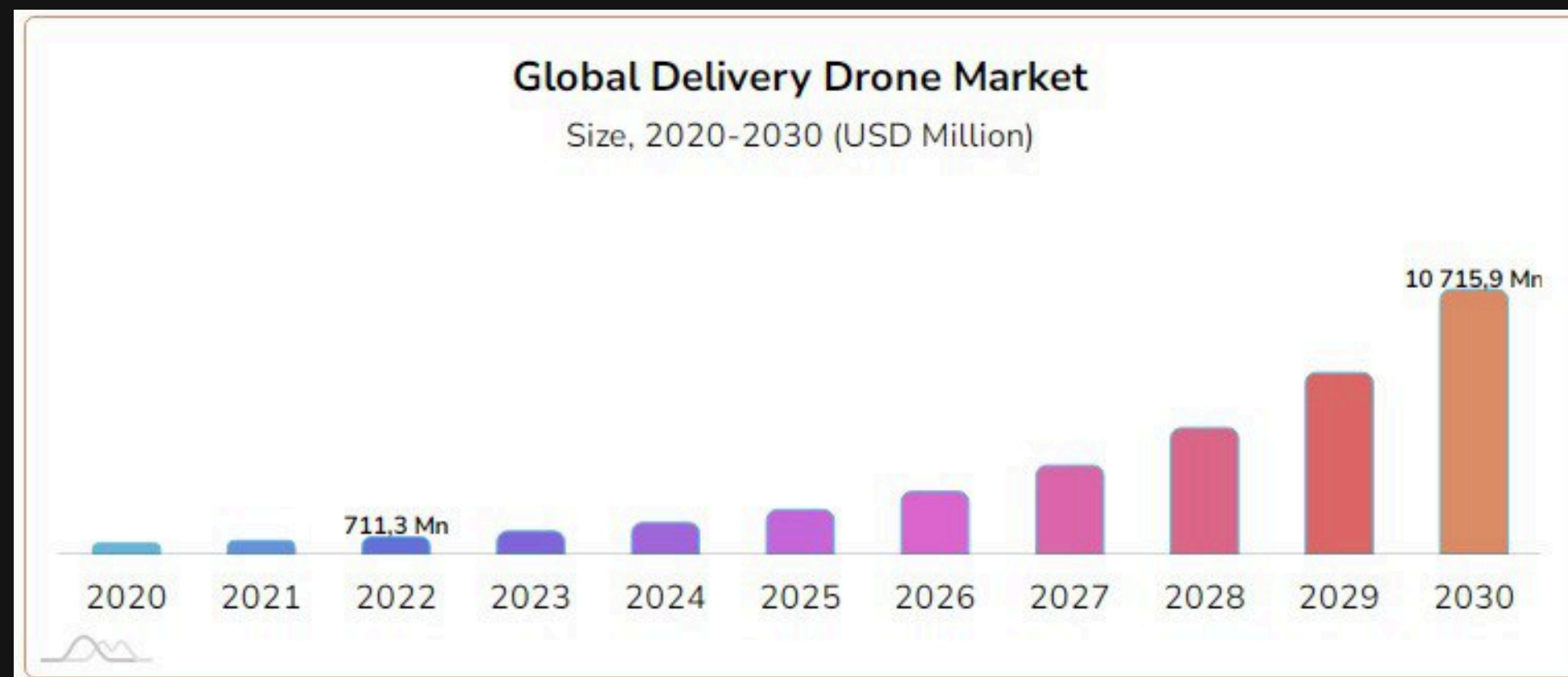
Труднодоступные места, эпидемиологическая ситуация, природные катаклизмы и многие другие причины могут стать препятствием для успешной наземной доставки биоматериала.

Воздушная доставка с помощью БПЛА позволит исключить данные факторы риска.

# РЫНОК

Среднегодовой темп роста составит 41,53% с 2023 по 2030 год.

Согласно отчету ведущей фирмы по исследованию рынка и консалтингу Kings Research, объем мирового рынка дронов-доставщиков оценивался в 711,3 млн долларов США в 2022 году и, по прогнозам, к 2030 году достигнет 10 715,9 млн долларов США, что более чем в 12 раз превышает показатели 2022 года.

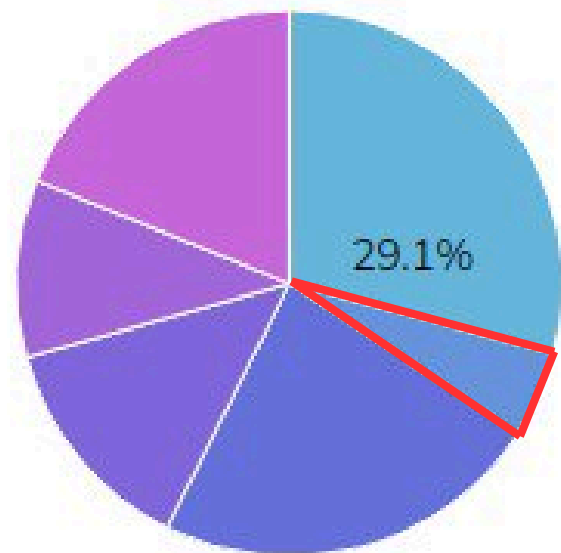


В исследовании учитывались продукты, выпускаемые ведущими мировыми компаниями:



# РЫНОК

Global Delivery Drone Market



В зависимости от области применения, рынок дронов для доставки сегментирован на розничную торговлю и электронную коммерцию, здравоохранение, логистику и транспорт, продукты питания и напитки, оборону и оборону, а также сельское хозяйство.

На приведенном графике можно увидеть, что доставка дронами в области Здравоохранения в мире, происходит меньше всего.

Таким образом, можно сделать вывод о том, что данная ниша практически не освоена в мире, особенно в России.

## Потенциальные заказчики

### Медицинские центры и клиники

Если у вас возникла необходимость сотрудничества с профильной лабораторией, которая не имеет своей курьерской службы, наша компания обеспечит оперативную транспортировку биологических образцов.

### Лаборатории

Поможем наладить эффективные логистические цепочки, снизить транспортные расходы. Не придется содержать и контролировать собственных курьеров, обеспечивать их контейнерами и расходниками.

### Владельцы питомников, хозяева домашних животных

Доставка биологического материала позволит сдать анализы в крупную ветеринарную лабораторию, многие из которых не выполняют транспортировку образцов.

### Физические лица в частных интересах

Отправить биоматериал в любую лабораторию, способную провести исследование можно с помощью

# Продукт

## TurboLët HA-0.1

БПЛА, который посредством взаимодействия с пользователем через приложение, доставляет биоматериалы из точки А в точку В

Предполагаемые сроки завершения работ над MVP (первым рабочим прототипом) — лето 2025 года.

# Концептуальная модель первого прототипа класса HYPER.analysis – TurboLët HA-0.1

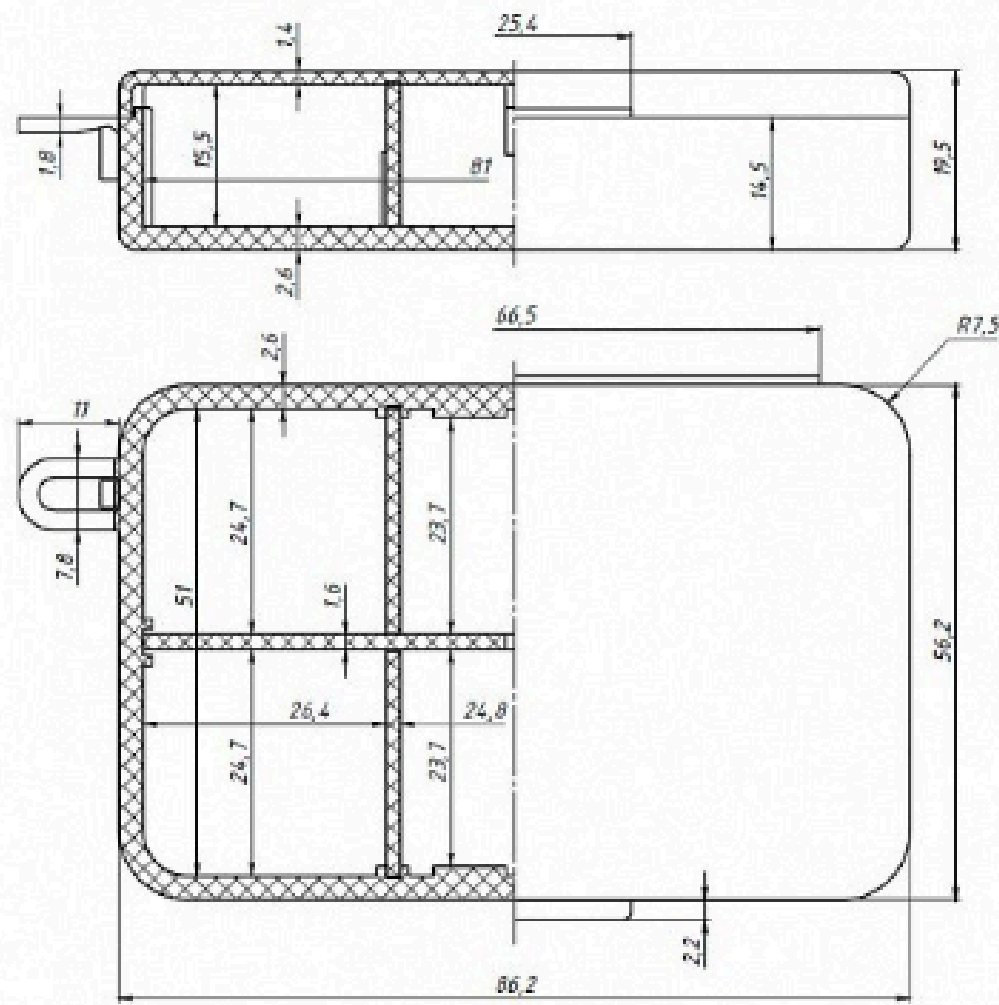




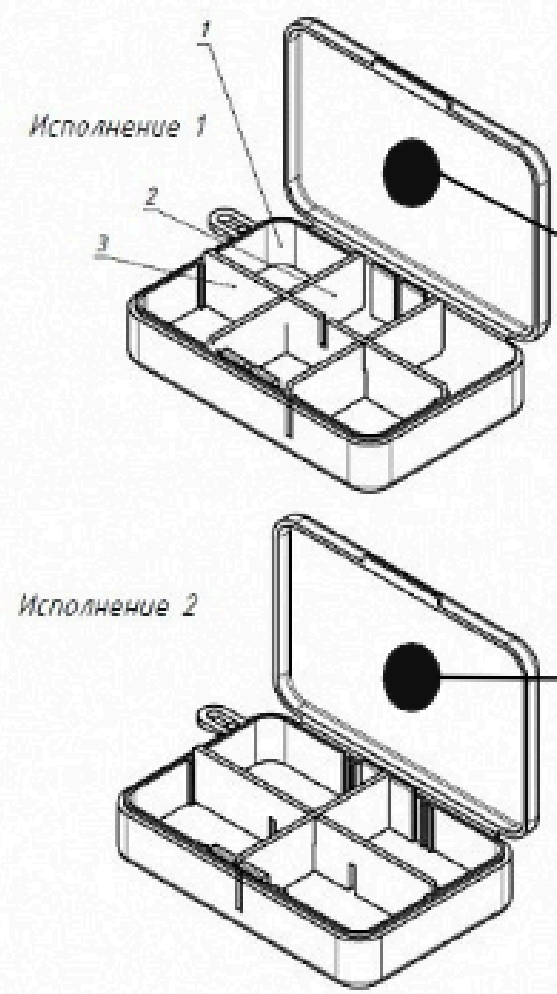


# Чертеж первого прототипа крепления TurboStack и контейнера TurboSTACK.analysis 0.1

ЧЕРТЕЖ И 3D-МОДЕЛЬ КОНТЕЙНЕРА КЛАССА TURBOSTACK.ANALYSIS  
И ЧЕРТЕЖ И 3D-МОДЕЛЬ КРЕПЛЕНИЯ КОНТЕЙНЕРА КЛАССА  
TURBOSTACK.ANALYSIS



ПОЗИЦИЯ	ДЕТАЛИ	К-ВО
1	В222 Контейнер	1
2	В222 Разделитель	2
3	В222 Разделитель 2	1



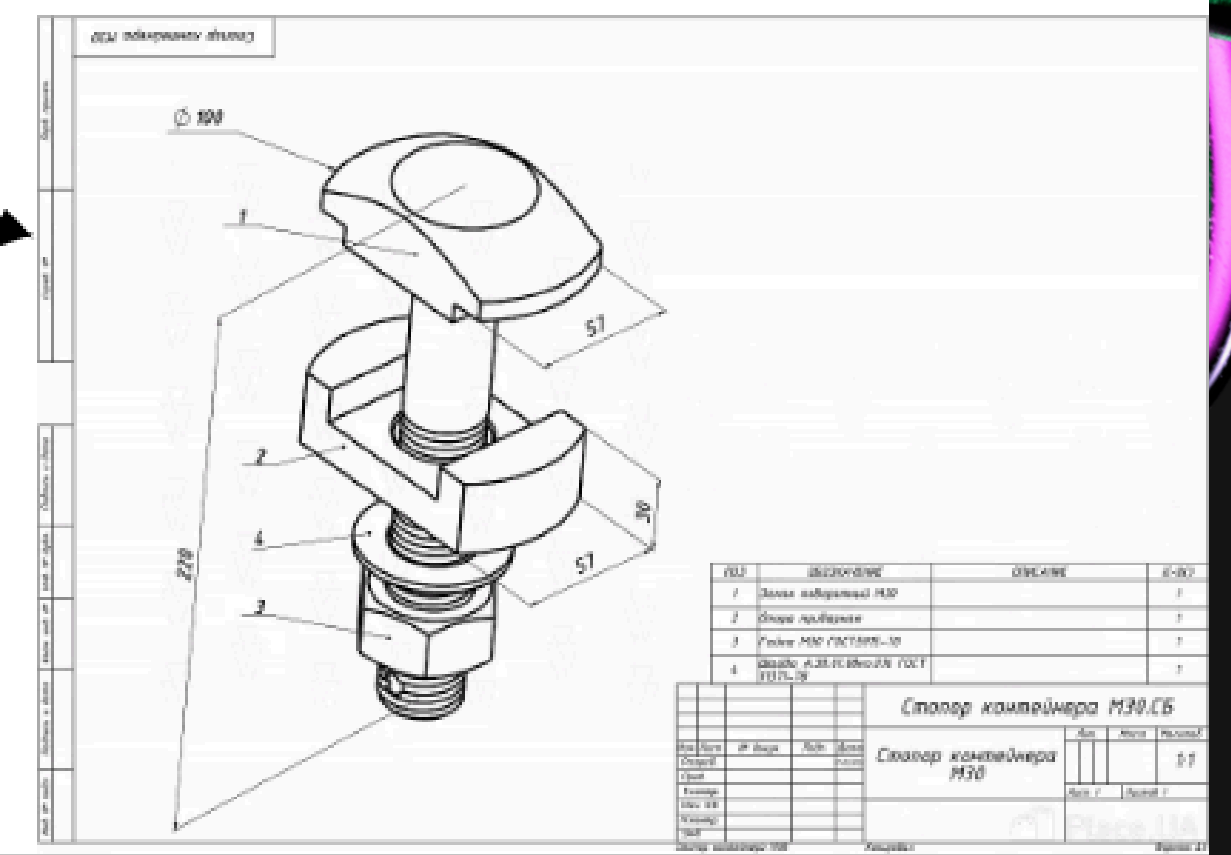
## Конструкция контейнера

Внутренний слой — медицинские полимеры  
Внешний слой — углепластики или алюминиевые сплавы  
Изоляционный слой — вакуумная изоляция

**Контроль параметров** — датчики для отслеживания температуры

**Температурный режим** (2 варианта реализации):

- 1) С помощью элемента Пельтье
- 2) Использование Гидрогелевых хладоэлементов



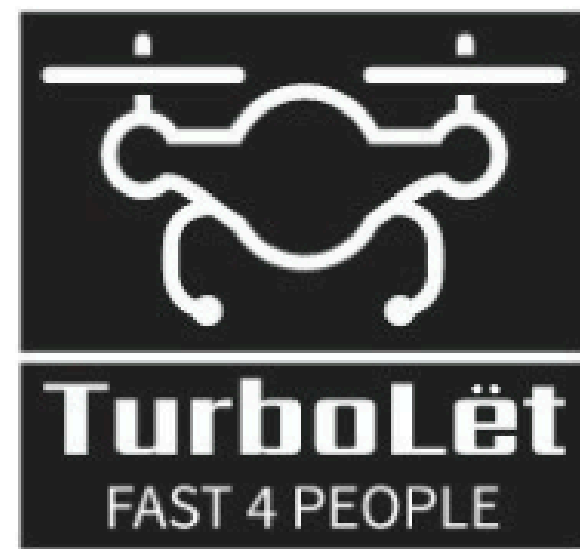
# Фирстиль

Фирменный шрифт проекта: ALS Sector, Capturanow, Montserrat

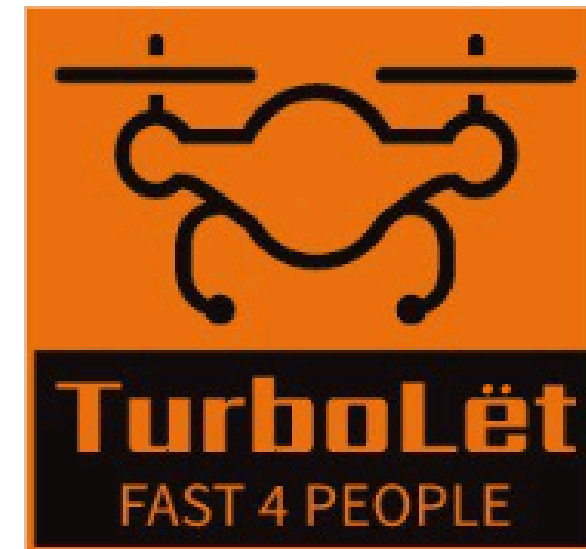
Фирменные цвета проекта: rgb(255,255,255), rgb(34,34,34),  
rgb(236,114,15), rgb(21,13,19)



Логотип проекта  
(ч/б N°1)



Логотип проекта  
(ч/б N°2)



Логотип проекта  
(цветной, пример)



Бренд-лого (ч/б)

# Расчеты \$ (предполагаемые)

## Затраты

4 555 000 рублей, из них:

2 600 000 на разработку ПО

120 000 на тестовое оборудование в ходе разработки

1 600 000 инвестиций на налаживание производства

235 000 на обучение и подготовку операторов дронов

**Прибыль спрогнозировать тяжело, однако  
предполагаемые доходы в первый месяц — 350 тыс. руб.**

# Текущие результаты

На момент декабря 2024 года

## СФОРМИРОВАНА БИЗНЕС-МОДЕЛЬ ПРОЕКТА

---

В дополнение к бизнес-модели создана предварительная смета услуг

## СОЗДАНА 3D-МОДЕЛЬ И КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ ПЕРВОГО ПРОТОТИПА КЛАССА HYPER.ANALYSIS – TURBOLET HA-0.1

---

Также создан MVP-код программной прошивки блоков управления TurboLët HA-0.1

## СОЗДАН ЧЕРТЕЖ МОДУЛЯ ХРАНЕНИЯ TURBOSTACK.ANALYSIS 0.1

---

Дополнительно разработан непосредственно крепеж МХ к дрону

# Таймлайн

**11.09.2024**

Старт  
разработки

**30.09.2024**

Созданы чертеж и 3D-  
модель 1-го прототипа

**с 03.10.2024**

**по н. вр.**

Начало поиска  
привлекаемых  
специалистов



**11.12.2024**

- Укомплектованный штат специалистов
- Заинтересованные покупатели
- Смета детализированная первого прототипа
- Начало разработки ПО
- Начало разработки приложения

КРАШ-ТЕСТ

Создана  
концептуальная  
модель 1-го прототипа

**20.09.2024**

Начало поиска  
потенциальных  
покупателей

**с 01.10.2024**

**по н. вр.**

Создан  
фирстиль  
проекта

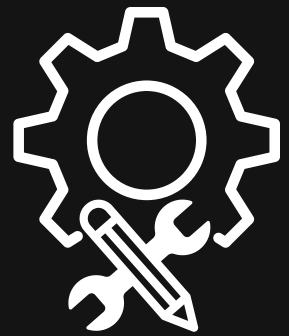
**07.10.2024**

CUSTDEV

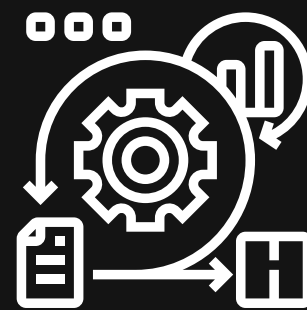
- Спецификация MVP;
- детализированные расчеты грузоподъемности/скорости/стоимости двух модификаций дронов/контейнеров; BPMN всех бизнес-процессов; эксп. образец NoSQL БД ПО

**зима 2025 года**

# Текущие процессы (на момент 11.12.2024)



Начата разработка прошивки (сл. 18-20)



Начата разработка приложения на базе ос-кода UberEats (сл. 21-22)



Детализированная смета MVP отдана в работу отв. за экономику проекта — *Вагуриной Ю.А.*  
(РУДН, каф. "Финансы и кредит")



Сформирован пул потенциальных покупателей (см. Приложение)



Сформирован штат идейных фрилансеров, происходит общение с SA, уточняются требования и формируется ТЗ  
(+ перевод тех. документации на eng, в связи с тем, что 1 исполнитель из Сербии)

*Отв-й SA: Соколов А.А. (ведущий специалист Raiffaisen Bank)*

*Перевод: Тюняева А.А. (МГТУ им. Н.Э. Баумана, каф. Л4 "Романо-германские языки")*

# Фрагмент кода: активация стартового звука запуска (кусочек драйвера сервоприводов в моторах дрона)

```
bool motorsTest(void)
{
    int i;

    for (i = 0; i < sizeof(MOTORS) / sizeof(*MOTORS); i++) {
        if (motorMap[i]->drvType == BRUSHED) {
#ifdef ACTIVATE_STARTUP_SOUND
            motorsBeep(MOTORS[i], true, testsound[i], (uint16_t)(MOTORS_TIM_BEEP_CLK_FREQ / A4) / 20);
            vTaskDelay(M2T(MOTORS_TEST_ON_TIME_MS));
            motorsBeep(MOTORS[i], false, 0, 0);
            vTaskDelay(M2T(MOTORS_TEST_DELAY_TIME_MS));
#else
            motorsSetRatio(MOTORS[i], MOTORS_TEST_RATIO);
            vTaskDelay(M2T(MOTORS_TEST_ON_TIME_MS));
            motorsSetRatio(MOTORS[i], 0);
            vTaskDelay(M2T(MOTORS_TEST_DELAY_TIME_MS));
#endif
        }
    }

    return isInit;
}
```



# Фрагмент кода прошивки I2C-шины: дефиниции переменных внутреннего стека данных, библиотеки

```
#include "freertos/FreeRTOS.h"  
#include "freertos/semphr.h"  
#include "freertos/queue.h"  
#include "driver/i2c.h"
```

```
#include "stm32_legacy.h"
```

```
▼ typedef struct _I2cMessage {  
    uint32_t    messageLength;    ///< How many bytes of data to send or received.  
    uint8_t     slaveAddress;     ///< The slave address of the device on the I2C bus.  
    uint8_t     nbrOfRetries;     ///< The slave address of the device on the I2C bus.  
    I2cDirection direction;      ///< Direction of message  
    I2cStatus   status;          ///< i2c status  
    xQueueHandle clientQueue;    ///< Queue to send received messages to.  
    bool        isInternal16bit;  ///< Is internal address 16 bit. If false 8 bit.  
    uint16_t    internalAddress;  ///< Internal address of device.  
    uint8_t     *buffer;         ///< Pointer to the buffer from where data will be read for transmission, or into which received data will be placed.  
} I2cMessage;  
  
typedef struct {  
    i2c_port_t    i2cPort;  
    uint32_t      i2cClockSpeed;  
    uint32_t      gpioSCLPin;  
    uint32_t      gpioSDAPin;  
    gpio_pullup_t gpioPullup;  
} I2cDef;
```

# Фрагмент кода конфига (слева направо): именование задач; приоритетность выполнения; определение выделенных битов памяти

```
#define SYSTEM_TASK_PRI 1
#define PM_TASK_PRI 1
#define LEDSEQCMD_TASK_PRI 1
// communication TX tasks
#define UDP_TX_TASK_PRI 2
#define CRTP_TX_TASK_PRI 2
// communication RX tasks
#define UDP_RX_TASK_PRI 2
#define EXTRX_TASK_PRI 2
#define UART2_TASK_PRI 2
#define SYSLINK_TASK_PRI 2
#define USBLINK_TASK_PRI 2
#define WIFILINK_TASK_PRI 2
#define CRTP_RX_TASK_PRI 2
#define CMD_HIGH_LEVEL_TASK_PRI 3
#define INFO_TASK_PRI 2
#define LOG_TASK_PRI 2
#define MEM_TASK_PRI 2
#define PARAM_TASK_PRI 2
// sensors and stabilize related tasks
#define PROXIMITY_TASK_PRI 5
#define FLOW_TASK_PRI 5
#define ZRANGER2_TASK_PRI 5
#define ZRANGER_TASK_PRI 5
#define SENSORS_TASK_PRI 6
#define STABILIZER_TASK_PRI 7
#define KALMAN_TASK_PRI 4

#define CMD_HIGH_LEVEL_TASK_NAME "CMDHL"
#define CRTP_RX_TASK_NAME "CRTP-RX"
#define CRTP_TX_TASK_NAME "CRTP-TX"
#define EXTRX_TASK_NAME "EXTRX"
#define FLOW_TASK_NAME "FLOW"
#define KALMAN_TASK_NAME "KALMAN"
#define LEDSEQCMD_TASK_NAME "LEDSEQCMD"
#define LOG_TASK_NAME "LOG"
#define MEM_TASK_NAME "MEM"
#define PARAM_TASK_NAME "PARAM"
#define PM_TASK_NAME "PWRMGNT"
#define PROXIMITY_TASK_NAME "PROXIMITY"
#define SENSORS_TASK_NAME "SENSORS"
#define STABILIZER_TASK_NAME "STABILIZER"
#define SYSLINK_TASK_NAME "SYSLINK"
#define SYSTEM_TASK_NAME "SYSTEM"
#define UART2_TASK_NAME "UART2"
#define UDP_RX_TASK_NAME "UDP_RX"
#define UDP_TX_TASK_NAME "UDP_TX"
#define USBLINK_TASK_NAME "USBLINK"
#define WIFILINK_TASK_NAME "WIFILINK"
#define ZRANGER2_TASK_NAME "ZRANGER2"
#define ZRANGER_TASK_NAME "ZRANGER"

#define configBASE_STACK_SIZE CONFIG_BASE_STACK_SIZE
#define CMD_HIGH_LEVEL_TASK_STACKSIZE (2 * configBASE_STACK_SIZE)
#define CRTP_RX_TASK_STACKSIZE (3 * configBASE_STACK_SIZE)
#define CRTP_TX_TASK_STACKSIZE (3 * configBASE_STACK_SIZE)
#define EXTRX_TASK_STACKSIZE (1 * configBASE_STACK_SIZE)
#define FLOW_TASK_STACKSIZE (3 * configBASE_STACK_SIZE)
#define KALMAN_TASK_STACKSIZE (3 * configBASE_STACK_SIZE)
#define LEDSEQCMD_TASK_STACKSIZE (2 * configBASE_STACK_SIZE)
#define LOG_TASK_STACKSIZE (3 * configBASE_STACK_SIZE)
#define MEM_TASK_STACKSIZE (2 * configBASE_STACK_SIZE)
#define PARAM_TASK_STACKSIZE (2 * configBASE_STACK_SIZE)
#define PM_TASK_STACKSIZE (4 * configBASE_STACK_SIZE)
#define SENSORS_TASK_STACKSIZE (5 * configBASE_STACK_SIZE)
#define STABILIZER_TASK_STACKSIZE (5 * configBASE_STACK_SIZE)
#define SYSLINK_TASK_STACKSIZE (1 * configBASE_STACK_SIZE)
#define SYSTEM_TASK_STACKSIZE (6 * configBASE_STACK_SIZE)
#define UART2_TASK_STACKSIZE (1 * configBASE_STACK_SIZE)
#define UDP_RX_TASK_STACKSIZE (4 * configBASE_STACK_SIZE)
#define UDP_TX_TASK_STACKSIZE (4 * configBASE_STACK_SIZE)
#define USBLINK_TASK_STACKSIZE (1 * configBASE_STACK_SIZE)
#define WIFILINK_TASK_STACKSIZE (4 * configBASE_STACK_SIZE)
#define ZRANGER2_TASK_STACKSIZE (4 * configBASE_STACK_SIZE)
#define ZRANGER_TASK_STACKSIZE (2 * configBASE_STACK_SIZE)
```

# Фрагмент кода приложения\*:

## модуль регистрации типа mail\PWD (Java) + библиотеки

```
public void onClick(View v) {

    emailid = email.getText().getText().toString().trim();
    pwd = pass.getText().getText().toString().trim();

    if(isValid()){

        final ProgressDialog mDialog = new ProgressDialog(Cheflogin.this);
        mDialog.setCanceledOnTouchOutside(false);
        mDialog.setCancelable(false);
        mDialog.setMessage("Sign In Please Wait.....");
        mDialog.show();

        Fauth.signInWithEmailAndPassword(emailid,pwd).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {

                if(task.isSuccessful()){
                    mDialog.dismiss();

                    if(Fauth.getCurrentUser().isEmailVerified()){
                        mDialog.dismiss();
                        Toast.makeText(Cheflogin.this, "Congratulation! You Have Successfully Login", Toast.LENGTH_SHORT).show();
                        Intent Z = new Intent(Cheflogin.this,ChefFoodPanel_BottomNavigation.class);
                        startActivity(Z);
                        finish();
                    }else{
                        ReusableCodeForAll.ShowAlert(Cheflogin.this,"Verification Failed","You Have Not Verified Your Email");
                    }
                }else{
                    mDialog.dismiss();
                    ReusableCodeForAll.ShowAlert(Cheflogin.this,"Error",task.getException().getMessage());
                }
            }
        });
    }
}
```

```
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.textfield.TextInputLayout;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
```

# Фрагмент кода приложения\*: модуль восстановление PWD (слева); модуль верификации моб. т. с таймером (справа)

```
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class ForgotPassword extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_forgot_password);
    }
}
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_verify_phone);

    phoneno = getIntent().getStringExtra("phonenummer").trim();

    entercode = (EditText) findViewById(R.id.codee);
    txt = (TextView) findViewById(R.id.text);
    Resend = (Button) findViewById(R.id.Resendotpp);
    verify = (Button) findViewById(R.id.Verifyy);
    FirebaseAuth = FirebaseAuth.getInstance();

    Resend.setVisibility(View.INVISIBLE);
    txt.setVisibility(View.INVISIBLE);

    sendverificationcode(phoneno);

    verify.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            String code = entercode.getText().toString().trim();
            Resend.setVisibility(View.INVISIBLE);

            if (code.isEmpty() && code.length() < 6) {
                entercode.setError("Enter code");
                entercode.requestFocus();
                return;
            }
            verifyCode(code);
        }
    });
    new CountdownTimer(60000, 1000) {

        @Override
        public void onTick(long millisUntilFinished) {

            txt.setVisibility(View.VISIBLE);
            txt.setText("Resend Code Within "+millisUntilFinished/1000+"Seconds");
        }
    };
}
```

# Планы развития

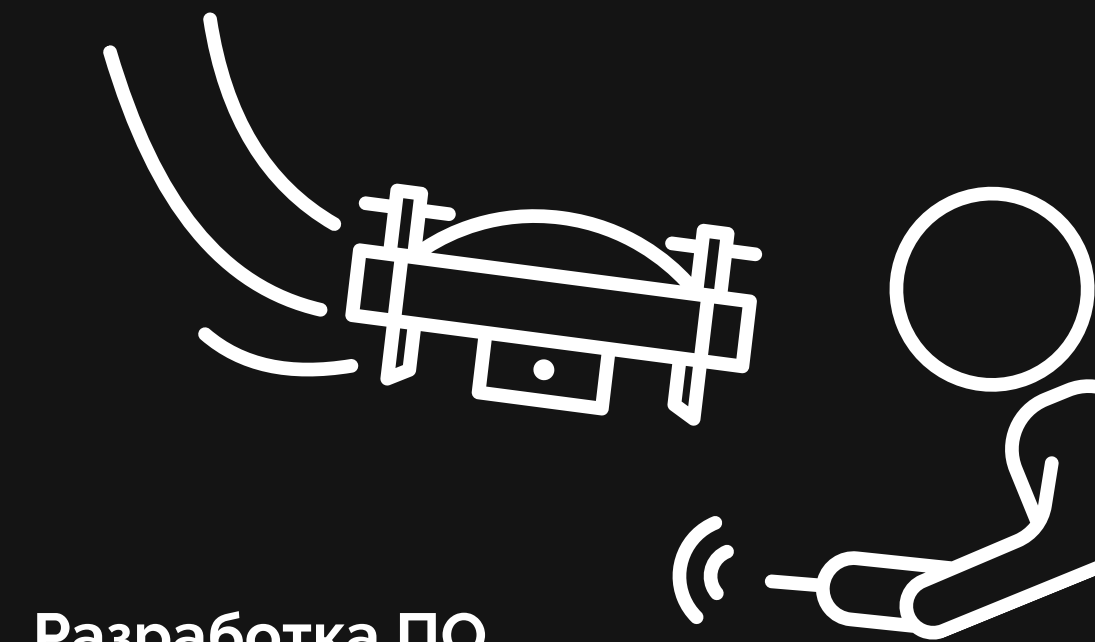
На данный момент управление дроном осуществляется с помощью оператора. В планы развития входит:

Определение комфортной зоны, которая подходит для полетов дрона в городской застройке г.Москвы и Московской области, как это разработала компания Amazon и согласование полетного режима с правительством Москвы и МО

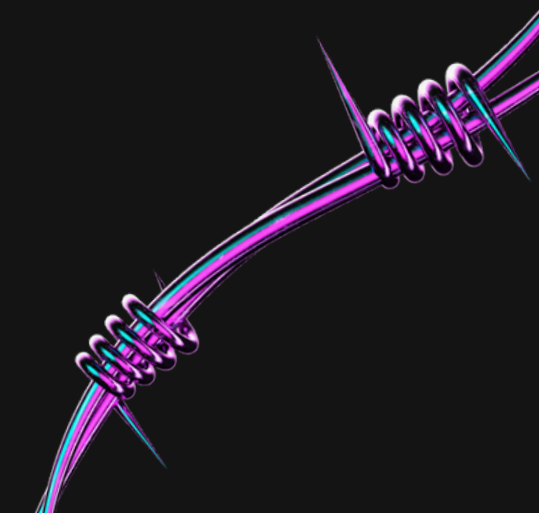
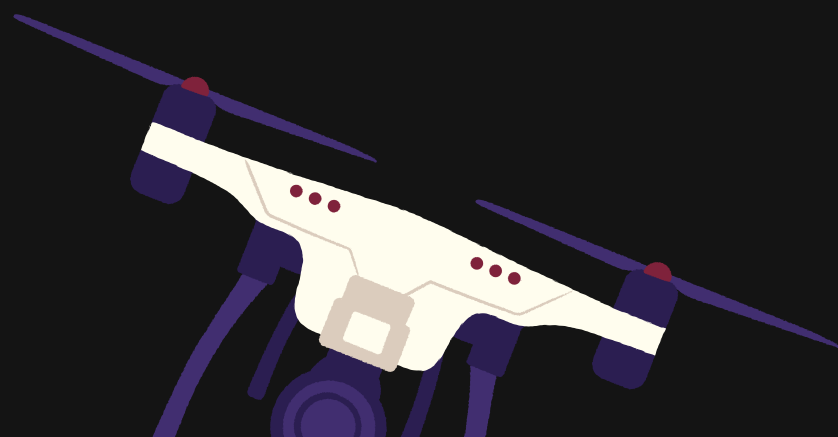
Создание двух дополнительных модификаций дронов с креплениями – класса Bio.FLY и класса Justice.AIR

Разработка автоматизированного перемещения дронов на основе искусственного интеллекта, без участия оператора.

Создание двух дополнительных модификаций контейнеров – TurboSTACK.bio и TurboSTACK.justice



Разработка ПО и создание станции базирования дронов TLCharge. Внедрение станций базирования в работу.



# Контакты команды проекта

**Голихин И.С.**

Моб. телефон: +7 925 276 61 97

Телеграм: @bludpunk

**Суханов А.П.**

Моб. телефон: +7 901 420 93 38

Телеграм: @jh3nd3Lll





# СПАСИБО ЗА ВНИМАНИЕ

В приложении можно найти  
фирстиль, спецификации классов  
дронов и контейнеров к ним, миссию  
и цель проекта, список  
потенциальных заказчиков в России

# ПОТЕНЦИАЛЬНЫЕ ЗАКАЗЧИКИ

**BIOCARD**



**DHL**



**БИОБОКС | 24**

**DELIVME**  
МЫ ПЕРЕМЕЩАЕМ РАССТОЯНИЯ

**POSTBURO**  
БЫСТРО. УДОБНО.  
С УМОМ

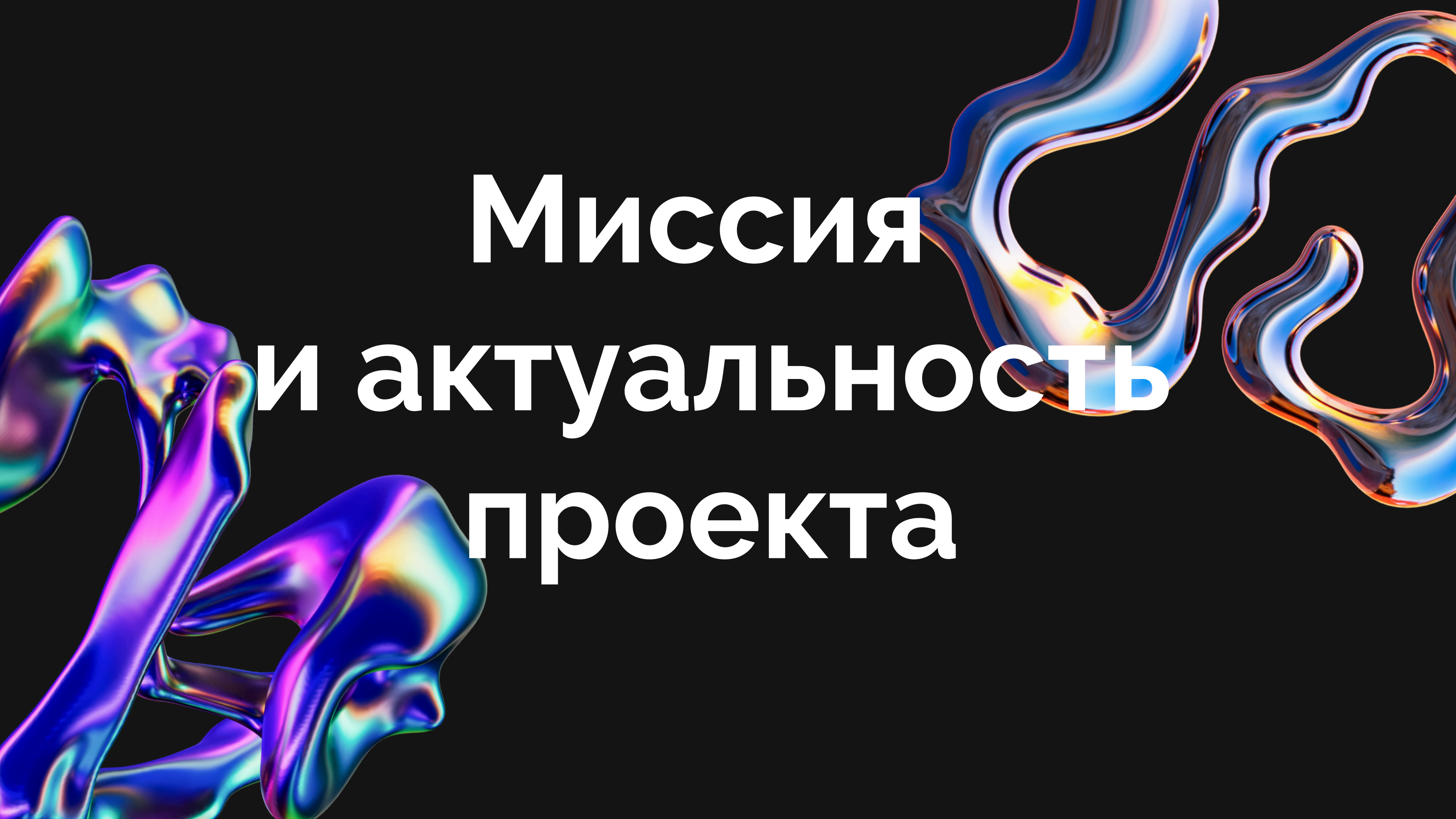
**Гарант-Логистика**

**ГЕНОМЕД**

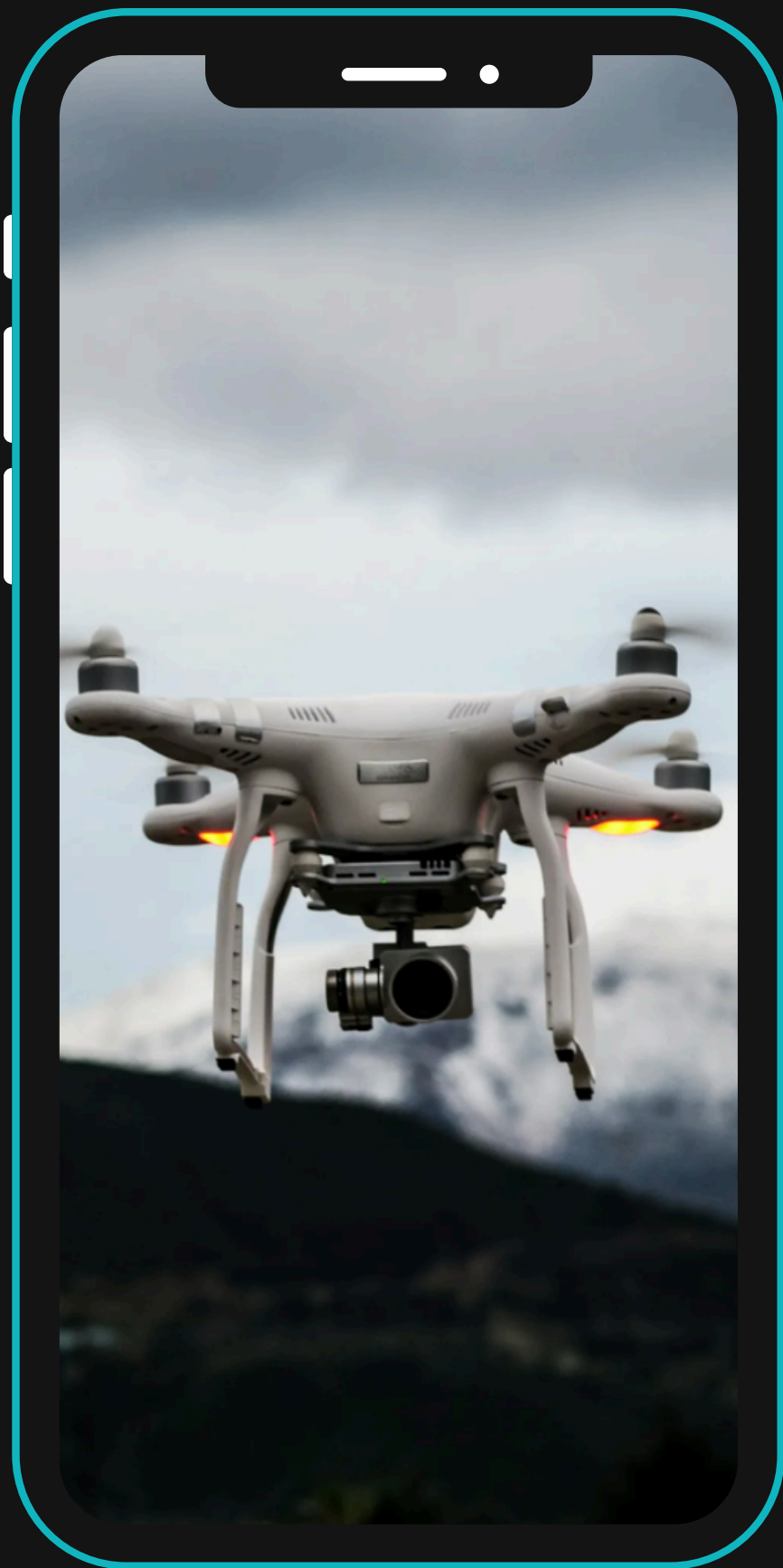


**GLOBE MEDICAL**  
Biopharma logistics company





**Миссия  
и актуальность  
проекта**



# Миссия и цель проекта



Миссия нашего проекта — доставлять биологические материалы туда, где они необходимы, с максимальной точностью и заботой и трансформировать традиционную систему логистики в медицине, правосудии и высоких технологиях с помощью инновационных беспилотных решений для транспортировки биологических образцов. Мы стремимся обеспечить скорость, точность и безопасность в каждом этапе доставки, минимизируя задержки и риски, чтобы спасти жизни, ускорять судебные процессы и повышать эффективность работы лабораторий, экспертных и медицинских учреждений.

Наша цель — создать в Москве и Московской области к 1 января 2026 года экосистему быстрой доставки биоматериалов с помощью БПЛА, включающую станцию базирования не менее 10 дронов-доставщиков, а также веб-сайт и MVP мобильного приложения сервиса.

Девиз нашего проекта: FAST 4 PEOPLE ("Быстро для людей")

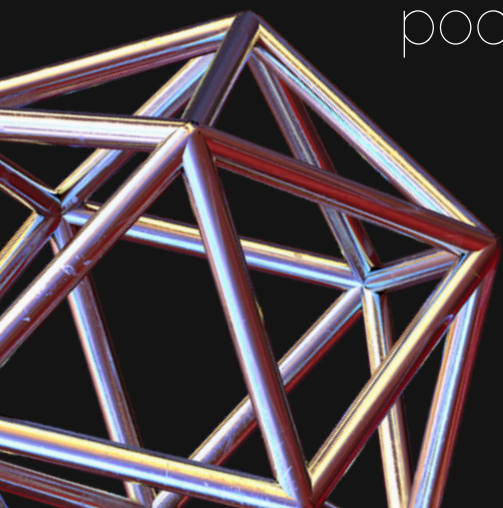
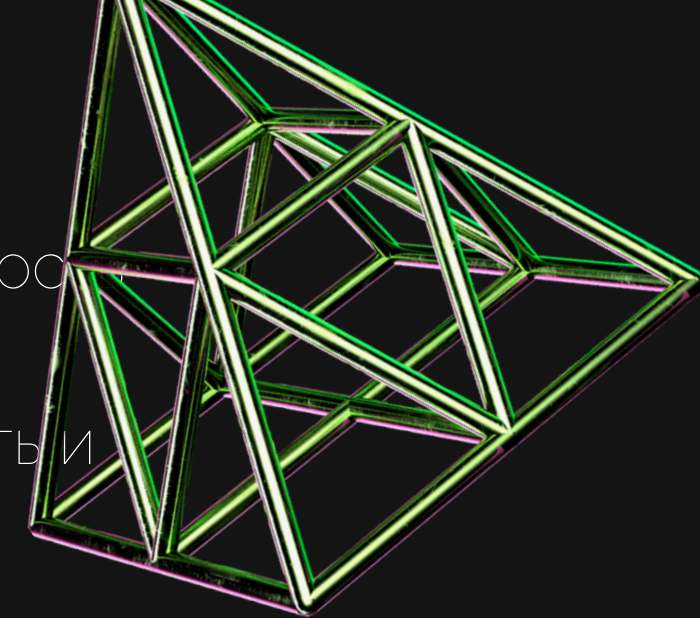
В настоящее время, с развитием технологий, требуется экстенсивно увеличивать возможности исследований биоматериалов человека. К сожалению, традиционная логистика не может обеспечивать достаточную скорость транспортировки, а человеческий фактор – надежность и безопасность отправляемых материалов.

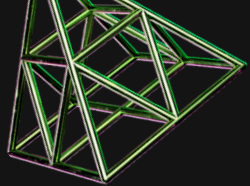
Скорость, надежность и безопасность являются самыми важными факторами при доставке биоматериалов. Это обусловлено тем, что срок их жизни крайне мал: от нескольких минут до нескольких суток, даже при создании нужной среды во время транспортировки. Поэтому проект TurboLët предлагает решение данных проблем путем доставки биоматериалов с помощью дронов-доставщиков.

Проанализировав рынок дронов-доставщиков в мире, можно сделать вывод о том, что данная сфера является развивающейся и перспективной.

В России же рынок дронов и БПЛА только начинает интенсивно развиваться. На данный момент в России не существует аналогов зарубежным сервисам беспилотной доставки биоматериалов с помощью дронов, поэтому проект TurboLet призван создать конкурентоспособный российский сервис в рассматриваемой транспортно-логистической нише.

АКТУАЛЬНОСТЬ





## 1. Расчет подъемной силы

Для оценки подъемной силы, создаваемой пропеллерами, можно использовать упрощенные аэродинамические формулы. Общая формула подъемной силы для пропеллеров выглядит так:

$F = k \cdot \rho \cdot n^2 \cdot D^4$ , где:  $F$  — подъемная сила одного пропеллера,  $k$  — коэффициент, зависящий от формы и угла атаки лопастей (обычно экспериментально определяется),  $\rho$  — плотность воздуха (около  $1.225 \text{ кг/м}^3$  на уровне моря),  $n$  — скорость вращения пропеллера в оборотах в секунду (RPM/60),  $D$  — диаметр пропеллера.

*Если мы не знаем точные значения  $k$  и  $D$ , можно воспользоваться средними значениями для дронов подобного типа.*

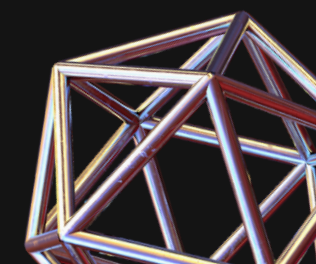
Переводим RPM в обороты в секунду:  $n = 6500/60 \approx 108$  оборотов/с

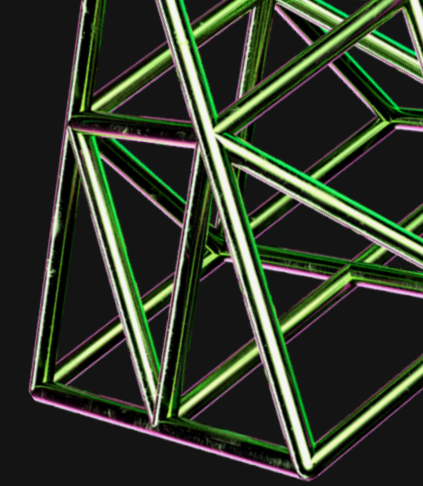
Диаметр пропеллеров: Допустим, диаметр пропеллеров составляет около 0.3 м (30 см), что типично для небольших дронов с массой до 2 кг.

Оценка подъемной силы: приблизительно, один такой пропеллер может создавать около 4–5 Н подъемной силы на максимальной скорости. Тогда:  $F_{total} = 4 \times F_{propeller} \approx 4 \times 5 = 20 \text{ Н}$

Так как масса дрона составляет 1.6 кг, это соответствует весу:  $W = m \cdot g = 1.6 \times 9.81 \approx 15.7 \text{ Н}$

Таким образом, такая конфигурация пропеллеров теоретически может удерживать вес дрона в 1.6 кг и поднимать дополнительный вес (около 4 Н, что соответствует примерно 400 граммам) в идеальных условиях.





## 2. Максимальная грузоподъемность

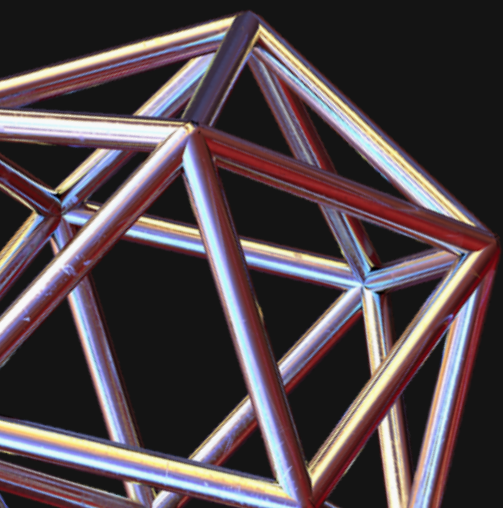
Итак, в теории, дрон с массой 1.6 кг и указанными параметрами пропеллеров сможет поднимать дополнительно около 300–400 грамм полезной нагрузки, но на практике эти значения могут быть меньше из-за энергопотерь, трения и влияния ветра.

## 3. Максимальная скорость

Максимальная горизонтальная скорость дрона зависит от множества факторов, таких как угол наклона при движении, сопротивление воздуха, а также мощности двигателей. Средняя скорость для дронов такого размера обычно составляет около 30–40 км/ч. Точные расчеты потребовали бы дополнительных данных по тяге в горизонтальном направлении и характеристикам аккумулятора.

РАСЧЕТЫ

Параметр	Значение
Вес дрона	1.5 кг
Средняя скорость полета	35 км/ч (9.72 м/с)
Максимальная грузоподъемность	1 кг
Скорость вращения пропеллеров	6500 об/мин
Грузоподъемность	около 0,7 кг



# Решение

---

- Проект TurboLet решает проблему с помощью дронов-доставщиков с использованием программного обеспечения собственной разработки, которое даст возможность пользователям сервиса заказать доставку биоматериала из любой точки города в любое время суток.
- Также мы предлагаем собственную разработку контейнеров для fragile-транспортировки и поддержания криоклимата для скороразрушающихся биологических материалов, учитывающих периоды полураспада и физические свойства наиболее часто встречающихся биоматериалов.
- На основе изученных исследований, анализе рынка и статистики, мы убеждены, что использование дронов, а также БПЛА-логистика станет ключом к повышению оперативности работы экспертных, правоохранительных и медицинских организаций, а также улучшению пользовательского опыта и комфорта граждан России. Проект совместит в себе дешевизну содержания, безопасность транспортировки и скорость доставки груза.

# Спецификация разрабатываемых продуктов

## ДРОН СО СПЕЦИАЛИЗИРОВАННЫМИ КРЕПЛЕНИЯМИ TURBOSTACK

### КЛАСС BIO.FLY

HEALTHCARE

Дрон с наивысшей мобильностью, оснащен контейнером типа TurboSTACK.bio для повышенной безопасности транспортировки материала.

### КЛАСС JUSTICE.AIR

JUSTICE

Дрон с высокой мобильностью, оснащен контейнером типа TurboSTACK.justice. ПО дрона разработано с использованием протокола шифрования, благодаря которому гео-метка БПЛА отображается оператору/пользователю, но не может быть перехвачена сторонним несертифицированным ПО.

### КЛАСС HYPER.ANALYSIS

EXPERT  
RESEARCH

Дрон средней мобильности с контейнером типа TurboSTACK.analysis. Оснащен контейнерным модулем хранения повышенной вместимости для транспортировки большего числа образцов за 1 цикл доставки.

# Спецификация разрабатываемых продуктов

## КОНТЕЙНЕР ПОД КРЕПЛЕНИЯ TURBOSTACK

### TURBOSTACK. BIO

HEALTHCARE

Контейнерный модуль хранения повышенной прочности, оснащенный внутренними крепежами-фиксаторами биоматериала.

### TURBOSTACK. JUSTICE

JUSTICE

Контейнерный модуль хранения, блокирующий криминальный доступ к гео-метке дрона несертифицированным сторонним ПО.

### TURBOSTACK .ANALYSIS

EXPERT  
RESEARCH

Контейнерный модуль хранения повышенной вместимости и грузоподъемности.



# Заимствования кода

Component	License	Origin	Commit ID
core/crazyflie	GPL3.0	Crazyflie*	tag_2024_01 b448553
lib/dsp_lib	open source	esp32-lin**	6fa49f4c

\* API-поддержка и код для тестов будущего продукта

\*\* Базовые мат. функции, база функций контроллера и матрицы

# Бизнес-модель (BRMN)



При разработке стартовой BRMN-модели рассматривался  
стартовый бизнес-процесс приемки и доставки заказа  
пользователя

**USER STORY:** Я, как пользователь, хочу заказать БПЛА-доставку, чтобы быстро доставить биоматериал

**Актор:** авторизованный пользователь

**Предусловия:** у пользователя есть профильное ПО

**Критерий приемки:** Заказ отправляется в конечную точку только после подтверждения пользователя

# BRMN

