



universa



РОССЕТИ



УДМУРТСКАЯ РЕСПУБЛИКА

ИНФОРМАЦИОННАЯ ПЛАТФОРМА

Автоматизация процессов согласования проектов присоединения к электросетям с использованием технологии распределенного реестра (блокчейн)

Удмуртская Республика



Проблемы присоединения к электросетям:



ЗАТРАТЫ ВРЕМЕНИ на выявление всех участников, проводящих работы в выбранном районе



ДЛИТЕЛЬНОЕ СОГЛАСОВАНИЕ проекта **БОЛЕЕ 200 ДНЕЙ**



ОТСУТВИЕ одномоментной **ВИЗУАЛИЗАЦИИ** всех процессов



УДОРОЖАНИЕ СТОИМОСТИ проекта присоединения

Проблемы существующего подхода:



НЕСООТВЕТСТВИЕ СРОКАМ

технологического присоединения **70 дней**
в рейтинге Всемирного банка Doing Business-2018



ЗАТРАТЫ на согласования проектов составляют
более 100 000 000 рублей в год



БОЛЕЕ 8 000 000 ЧАСОВ затрачиваются
на процесс согласования



НАРУШЕНИЕ ПОСТАНОВЛЕНИЯ ПРАВИТЕЛЬСТВА
и существующей дорожной карты

Как разрабатывается проект сейчас:

1. Заказ на бумаге новой топографической карты района
 2. Отображение проекта на карте с учетом ограничений
 3. Выявление сторонних компаний, которые ведут работы на данной территории
 4. Только последовательное согласование с заинтересованными компаниями и госорганами
-

Что изменит внедрение блокчейн-технологий Universa:

1. **Единая интерактивная** карта для работы всех компаний
2. **Разные слои** информации (кадастры/сети/геология)
3. Раздельный **доступ** к этой информации
4. **Простое согласование** проекта в информационной системе
5. **Скрытие информации** с грифом секретности с сохранением возможности проверки возможности прокладки проекта через эти территории.

```
elif_operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
elif_operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end -add back the deselected mirror mod
mirror_ob.select=1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected: " + str(modifier_ob)) #modifier ob is the act
mirror_ob.select=0
```

Блокчейн-технологии Universa позволяют:

1. **Актуализировать карту** всех проводимых проектов
 2. **Упростить схемы** взаимодействия между участниками инфраструктурных сетей
 3. **Отобразить все варианты** реализации проекта на нужной территории
 4. **Автоматизировать согласования** проекта участниками
 5. **Ускорить исполнение** проекта и уменьшить издержки на его разработку
-

```
class MirrorX(bpy.types.Operator):
    """This adds an X mirror to the selected object"""
    bl_idname = "object.mirror_mirror_x"
    bl_label = "Mirror X"

    @classmethod
    def poll(cls, context):
        return context.active_object

    def execute(self, context):
        mirror_mod.use_x = True
        mirror_mod.use_y = False
        mirror_mod.use_z = False
        elif_operation == "MIRROR_Y":
            mirror_mod.use_x = False
            mirror_mod.use_y = True
            mirror_mod.use_z = False
        elif_operation == "MIRROR_Z":
            mirror_mod.use_x = False
            mirror_mod.use_y = False
            mirror_mod.use_z = True

        #selection at the end, add back the deselected mirror modifier
        mirror_ob.select = 1
        modifier_ob.select = 1
        bpy.context.scene.objects.active = modifier_ob
        print("Selected" + str(modifier_ob)) # modifier_ob is the active ob
        mirror_ob.select = 0
        #one = bpy.context.selected_objects[0]
        #bpy.data.objects[one.name].select = 1
    except ValueError:
        print("please select exactly two objects to create the modifier")

class MirrorY(bpy.types.Operator):
    """This adds a Y mirror to the selected object"""
    bl_idname = "object.mirror_mirror_y"
    bl_label = "Mirror Y"

    @classmethod
    def poll(cls, context):
        return context.active_object

    def execute(self, context):
        mirror_mod.use_x = False
        mirror_mod.use_y = True
        mirror_mod.use_z = False
        elif_operation == "MIRROR_X":
            mirror_mod.use_x = True
            mirror_mod.use_y = False
            mirror_mod.use_z = False
        elif_operation == "MIRROR_Z":
            mirror_mod.use_x = False
            mirror_mod.use_y = False
            mirror_mod.use_z = True

        #selection at the end, add back the deselected mirror modifier
        mirror_ob.select = 1
        modifier_ob.select = 1
        bpy.context.scene.objects.active = modifier_ob
        print("Selected" + str(modifier_ob)) # modifier_ob is the active ob
        mirror_ob.select = 0
        #one = bpy.context.selected_objects[0]
        #bpy.data.objects[one.name].select = 1
    except ValueError:
        print("please select exactly two objects to create the modifier")

class MirrorZ(bpy.types.Operator):
    """This adds a Z mirror to the selected object"""
    bl_idname = "object.mirror_mirror_z"
    bl_label = "Mirror Z"

    @classmethod
    def poll(cls, context):
        return context.active_object

    def execute(self, context):
        mirror_mod.use_x = False
        mirror_mod.use_y = False
        mirror_mod.use_z = True
        elif_operation == "MIRROR_X":
            mirror_mod.use_x = True
            mirror_mod.use_y = False
            mirror_mod.use_z = False
        elif_operation == "MIRROR_Y":
            mirror_mod.use_x = False
            mirror_mod.use_y = True
            mirror_mod.use_z = False

        #selection at the end, add back the deselected mirror modifier
        mirror_ob.select = 1
        modifier_ob.select = 1
        bpy.context.scene.objects.active = modifier_ob
        print("Selected" + str(modifier_ob)) # modifier_ob is the active ob
        mirror_ob.select = 0
        #one = bpy.context.selected_objects[0]
        #bpy.data.objects[one.name].select = 1
    except ValueError:
        print("please select exactly two objects to create the modifier")
```

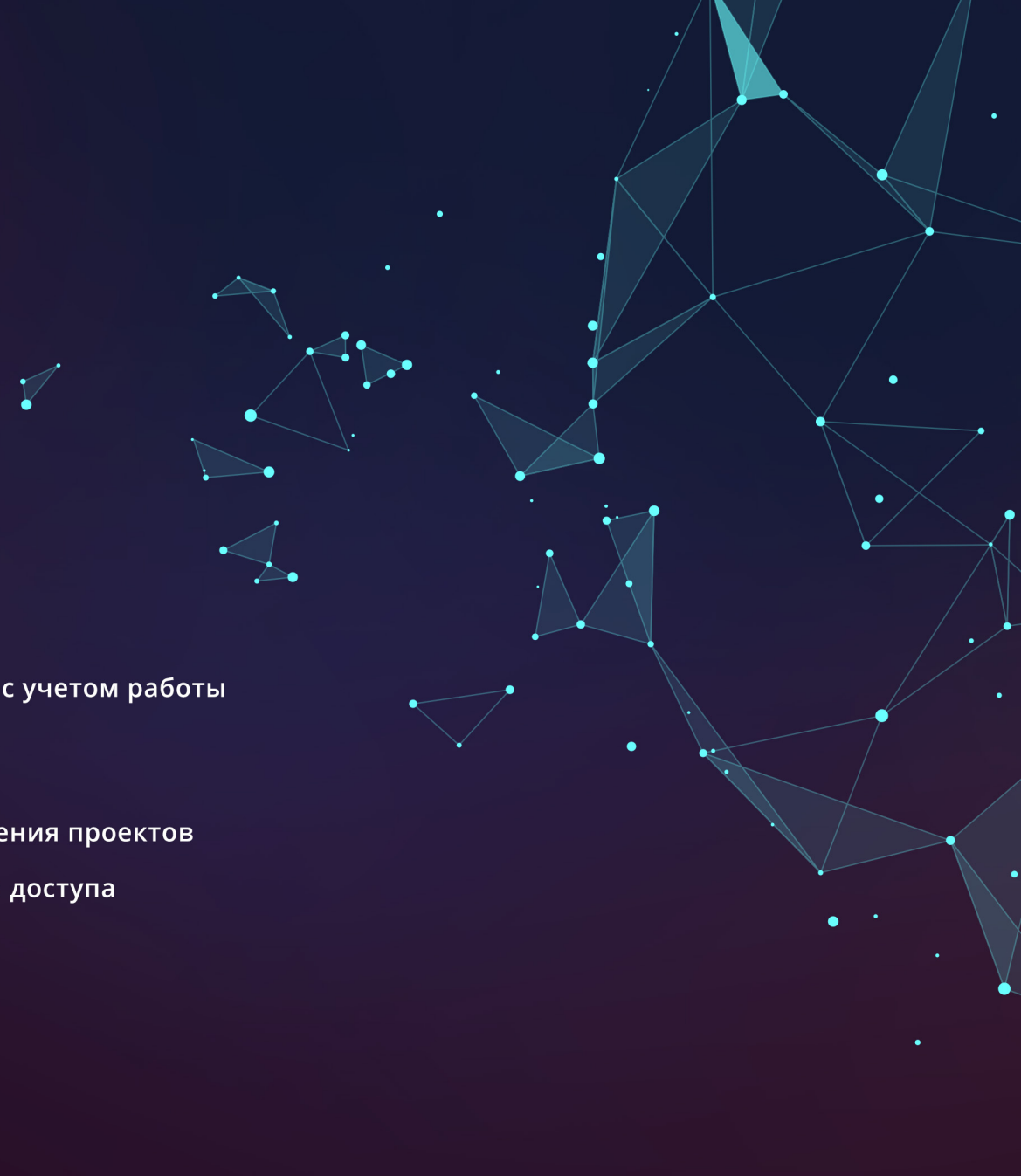

Пилотный проект системы — Удмуртия, Ижевск, территория ИЖМАШ

Подготовка к запуску пилотного проекта:

- Оцифровка карты района с отметками всех систем
- Добавление рабочих проектов
- Размещение серверов у контролирующих органов, которые будут видеть полную картину по проектам
- Подготовка работы системы в компаниях с возможностью корректировки проектов компаний на территориях



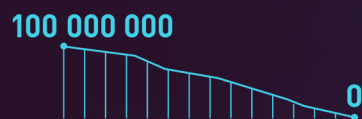
Внедрение системы

- > Создание программного комплекса
 - > Программное решение для оцифровки карты и сетей с учетом работы с их отображением на изображении
 - > Редактор сетей
 - > Внедрение нового алгоритма регистрации и утверждения проектов
 - > Создание единой базы проектов с разными уровнями доступа
- 
- An abstract network diagram consisting of numerous nodes (small circles) connected by thin lines, forming a complex web of relationships. The nodes and lines are rendered in shades of light blue and white against a dark background. The diagram is positioned on the right side of the slide, extending from the top right towards the bottom right.

Автоматизация проектного бюро

По мере накопления данных по проектам и условиям их разрешения появится возможность автоматизировать задачи проектного бюро. Информационная система сможет без участия оператора рассчитывать стоимость и сроки присоединения всех возможных проектов на указанной территории.

Что изменит внедрение блокчейн - технологии Universa:



СНИЖЕНИЕ ИЗДЕРЖЕК на согласование проекта присоединения



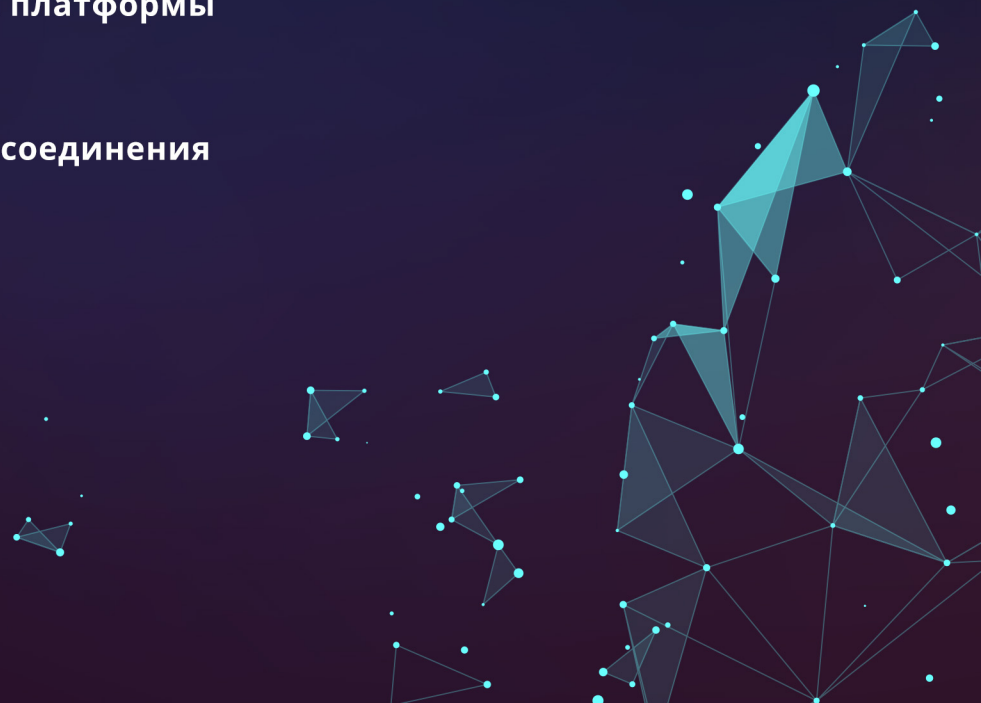
ЭКОНОМИЯ 8 000 000 ЧАСОВ на согласование проектов

Ускорение среднего времени согласования проекта

200 дней



С внедрением информационной платформы на технологии блокчейн Universa, **мы совершим настоящий прорыв в области технологического присоединения к электроснабжению**



Сроки реализации и стоимость

Пилотный проект —

6 месяцев и 25 млн рублей

Масштабирование на весь регион —

8 - 12 месяцев и 90 млн рублей



Что такое universa blockchain

Universa —

блокчейн - платформа для построения приложений, которые решают задачи корпоративного бизнеса и госуправления.

Universa —

представляет технологию децентрализованного управления криптозащищенными документами и развитые сервисы для работы с ними.

Universa —

позволяет поставщикам услуг и рыночным игрокам предлагать услуги в блокчейн и выполнять платежные операции в различных токенах или валютах*.

*Универсальные контракты **Universa** применяются для программирования внутрикорпоративных и публичных приложений

