

Робономика: платформа интеграции кибер-физических систем в экономику человека

Инженерам, создателям умных городов и Индустрии 4.0

Сергей Лоншаков
Александр Крупенькин
Александр Капитонов, PhD
email research@aira.life

Евгений Радченко
Алишер Хасанов
Александр Старостин
email engineering@aira.life

12 мая 2018 г.

Аннотация

Полномасштабное применение кибер-физических систем в производстве, логистике и жизни городов позволит справиться с возрастающей сложностью цепочек обеспечения потребителей качественной продукцией. В данном направлении одним из ключевых вопросов, по мнению авторов статьи, является организация совместной работы множества кибер-физических систем, работающих по всей планете.

Мы считаем, что наиболее жизнеспособным вариантом организации совместной работы автономных кибер-физических систем в масштабах планеты является объединение их в сеть на основе рыночных механизмов.

Адаптирующаяся к потребностям человека сеть распределения, контроля и оказания услуг кибер-физическими системами, построенная на основе рыночных механизмов и напрямую доступная конечному потребителю, может быть создана на основе p2p технологий, позволяющих объединить технические и экономические параметры транзакции к машине.

Построение подобной пиринговой сети может осуществляться на базе инфраструктуры Ethereum. Расширив возможности базового протокола коммуникации, мы можем обучить кибер-физические системы взаимодействию с рыночными механизмами и контрактными обязательствами.

Создание и развитие платформы, предоставляющей инструменты работы с сетью экономики роботов (более кратко - платформы Робономики), позволит проектировщикам новых городов и промышленных зон создавать доверие к услугам автономных роботов, предоставит прямой доступ пользователей к заказу продукции автономных фабрик и услуг сенсорных сетей городов; позволит задействовать децентрализованную систему, следящую за деятельностью кибер-физических систем по всей планете.

Содержание

1	Введение	4
2	Экономика роботов	5
3	Представление о кибер-физической системе	5
4	Обязательство машины	6
5	Платформа Робономики	6
5.1	Ethereum Blockchain транзакции и контракты обязательств	7
5.1.1	Фабрика контрактов	7
5.1.2	Легковесные контракты	7
5.1.3	Создание контракта на основе отложенных подписей	8
5.2	Провайдеры и маяки	8
5.3	Программа Робономики	9
5.4	Операционная система агента Робономики	10
5.4.1	Декларативная конфигурация и поколения ОС	10
5.4.2	Исполнение программ, описанных в умных контрактах	11
5.5	Наблюдающая сеть	13
5.5.1	Алгоритм работы контракта наблюдателя	13
6	Токен Робономики, XRT	14
6.1	Эмиссия за исполнение программы Робономики в Ethereum компьютере . . .	14
6.1.1	TGE и Голландский аукцион	15
6.1.2	Становление	15
6.1.3	Дальнейшая эмиссия	16
6.2	Комиссии за запуск программы Робономика в Ethereum инфраструктуре . .	16
6.3	Дополнительные задачи XRT токена	16
6.3.1	Наблюдающие сети существуют за счет комиссии пользователей . . .	16
6.3.2	XRT на счету маяков как механизм доверия к Робономике	16
6.3.3	XRT как количественное отражение капитала сети	17
6.3.4	Рынок кибернетики	17
6.3.5	Выводы о движении XRT в Робономике	17
6.3.6	Denomination	17
7	Описание работы Робономики	17
7.1	Предложение новой модели поведения инженером	17
7.2	Начало работы провайдера сети Робономики	18
7.3	Начало работы кибер-физической системы как сервиса	19
7.4	Появление пользователя в сети Робономики	19
7.5	Создание обязательства на исполнение сервиса кибер-физической системой .	20
7.6	Исполнение обязательства кибер-физической системой	21
8	Приложения платформы Робономики	22
8.1	Создание доверия к продукции умных городов и умных фабрик	22
8.2	Запуск умной фабрики пользователем	23
8.3	Управление экономикой роботов с помощью капитала	23
8.3.1	Референсная имплементация	24

8.4	Токенизированные ценности на основе труда работа	24
9	Масштабируемость	25
10	Сегмент Роботики в сети Ethereum	25
11	Заключение	25

1 Введение

Глобальное изменение большей части производственных процессов ожидается в ближайшем будущем [15] благодаря значительному потенциалу робототехнических систем: они способны адаптироваться [18] к широкому спектру задач, они эффективнее во многих видах операционной деятельности, и они снижают временные затраты на производство. Роботизация производства растет: мировое среднее число робототехнических единиц на 10 000 работников предприятий увеличилось на 11% с 2015 по 2016 год [17]. Например, только в Центральной и Восточной Европе в 2017 году было оборудовано на 28% [5] робототехнических единиц больше чем в 2016 году. Большинство предпринимателей видят в робототехнике возможность повышения эффективности своего производства и компенсации дефицита кадров [5].

Системы вычислительных устройств, которые с одной стороны кооперируются друг с другом через сетевые службы доступа и обработки данных, а с другой - активно взаимодействуют с окружающим физическим миром, представляются понятием кибер-физических систем (КФС) [7]. Основные технологические тенденции этого понятия, помимо автономных роботов, это Большие данные, Интернет Вещей, облачные вычисления и т.д. Кибер-физические системы являются ключевым элементом концепции Индустрии 4.0 [6] — четвертой промышленной революции, происходящей благодаря внедрению CPS. Масштаб изменений прогнозируют настолько значительным, что они затронут и изменят [14] не только способы производства, но и основополагающие принципы по которым человечество занималось промышленной индустрией.

Таким образом эти глобальные изменения ставят перед учеными и разработчиками множество новых вызовов. Один из важных вопросов [12], который стоит на пути внедрения автономных агентов (мы объединяем в это понятие робототехнические системы, умные вещи и чисто программных агентов) в производство и который рассматривают [20] исследователи в области робототехники — вопрос организации совместной и отлаженной работы множества агентов [13]. Причем, специфика [11] современной промышленной индустрии привела к тому, что централизованная структура организации столкнулась с большими издержками, связанными с рисками сбоя, ошибок или взлома. Исследователи считают, что распределенный подход к производству становится более практичным: благодаря ему снижаются транзакционные издержки (затраты на заключение контрактов), транспортные расходы, расходы на хранение запасов и издержки на устаревание оборудования [10]. Из-за этого в производстве наиболее перспективными выглядят децентрализованные системы [3].

Bitcoin создал сеть, в которой роботы могут иметь собственные аккаунты в экономической среде человеческого общества. Машины получили возможность хранить и обменивать ценности без прямой помощи человека. Другими словами, Bitcoin стал первыми деньгами для роботов [9].

Запуск сети Ethereum позволил нам расширить возможности экономически значимой коммуникации роботов с помощью умных контрактов [1]. Умные контракты теперь позволяют добавить технические детали в финансовую транзакцию к машине.

Вместо двух транзакций, экономически значимой транзакции от потребителя к оператору кибер-физической системы и технической транзакции от оператора к роботу, который должен предоставить сервис, мы можем использовать возможность, которую нам дает использование умных контрактов сети Ethereum, и отправить одну транзакцию от потребителя напрямую к кибер-физической системе.

Роботы стали способны исполнять программы, описанные в условиях контракта, и создавать токенизированные ценности на основе своего труда.

Благодаря развитию современной кибернетики в трудах Норберта Виннера [19]; форми-

рованию первых представлений об экономической кибернетике Виктором Михайловичем Глушковым [21]; и вкладу Рональда Коуза в понимание природы фирмы [2] сегодня мы можем говорить о возможности построения сети по управлению кибер-физическими системами в умных городах и Индустрии 4.0 с помощью рыночных механизмов.

2 Экономика роботов

Экономическая теория рассматривает вопросы ведения хозяйства в условиях неограниченного набора потребностей и ограниченных ресурсов. Стремление общества к автоматизации является элементом решения задачи рационального, или же более эффективного, расходования ресурсов для удовлетворения все большего количества потребностей. Мы просто физически не способны производить ручным трудом столько товаров и услуг, сколько потребляем уже сегодня. Помимо количественного показателя, немаловажным является и качественный вопрос: сложность как самих товаров, так и обеспечение контроля за их производством выходят за рамки человеческих возможностей. Производство микроэлектроники, тяжелая промышленность, фармацевтика – эти индустрии уже сегодня не способны существовать без участия машин.

В работе промышленных зон и жизни современных городов неизбежно появление полностью автоматизированных предприятий. Предприятий, контролируемых кибер-физическими системами и представляющих услуги, как автономные агенты. Неизбежен также и процесс формирования сетей из автономных КФС с целью повышения скорости и качества коммуникации в процессе производства и оказания услуг.

Мы защищаем рыночный механизм как способ построения сети кибер-физических систем. Рынок, с одной стороны, сделает сеть КФС адаптирующейся к изменяющимся потребностям человека, а с другой стороны – даст регуляцию размеров КФС с точки зрения их экономической эффективности как автономных агентов.

Объединение КФС с помощью рыночного механизма дает нам возможность реализовать планетарную систему массового производства товаров и услуг, напрямую интегрированную в экономику общества. Эта система и есть сеть экономики роботов.

3 Представление о кибер-физической системе

При проектировании роботизированных сервисов важно учитывать, что использование рыночных механизмов для объединения в сеть накладывает требование по представлению КФС как экономического агента. Подобно процессу появления фирм на рынке, описанного Рональдом Коузом в статье “Природа фирмы”, можно рассмотреть процесс эволюции кибер-физических систем в автономных экономических агентов. Издержки и возможности рыночного механизма реализуют среди КФС экономическую игру, которая, с одной стороны, создает фильтр, отсеивающий возможность коммуникации для экономически неэффективных роботизированных сервисов, а с другой – позволяет избежать увеличения КФС до размеров, вызывающих риски и излишние издержки существования сверх централизованных систем для окружающего общества.

Кибер-физическая система (КФС), подобно фирме в экономике, объединяет множество роботов в тесно связанную сеть сенсоров и актуаторов, способных к организованной совместной деятельности.

4 Обязательство машины

Мы живём в мире прав и обязательств. Если человечество по умолчанию не видит для себя пользы от приобретения роботами прав, то обязательства, накладываемые на машины, людей точно должно заинтересовать.

Защищая рыночный механизм как способ коммуникации в сети автономных КФС, мы должны определить, что будет являться предметом соглашения с машиной.

В процессе производства товаров и оказания услуг машина действительно не способна воспринимать предмет соглашения так же, как человек. Но если мы рассмотрим первые изменения в экономической теории после 17 века и вспомним теорию цены Маршалла, то заметим, что восприятие стоимости товара потребителем опирается на субъективное восприятие полезности этого товара в хозяйстве данного индивида, а оценка стоимости товара производителем - из издержек производства. В связи с этим мы можем говорить о том, что и в сегодняшнем, исключительно человеческом, обществе восприятие товара или услуги различно для потребителя и производителя.

Машины не льют воду. Если вы заключили договор с машиной, то она исполнит свою программу. Риски, связанные с мотивацией на выполнение задачи человеком, могут быть исключены при сделке с КФС. В таком случае мы можем использовать обязательство на исполнение некоторой модели поведения КФС с пользовательскими данными, как предмета сделки с машиной. В результате оценки потребительских качеств различных моделей поведения КФС, которые уже исполнены, мы увидим градацию цены спроса на различные модели поведения машин внутри одного и того же рынка товаров или услуг. Данной информации достаточно для того, чтобы инвестировать в развитие робототехники для исполнения все более совершенных моделей поведения КФС.

Предоставив машине возможность брать на себя обязательства и создавать обязательства по своему обслуживанию, мы можем получить полноценную интеграцию экономически автономного агента в среду потребителей. Предметом такого обязательства должно быть исполнение модели поведения, в результате чего потребитель понимает, что будут удовлетворены его потребности. Например, представьте вендинговую машину, которая собирает плату с работников офиса в интересах самообслуживания на цифровом рынке услуг и товаров города.

Модель поведения, или программа, КФС, в которой учтены технические и экономические параметры ее коммуникации, может быть представлена как унифицированный товар на рынке экономики роботов, выраженный контрактным обязательством.

5 Платформа Робоники

Нами рассматривается схема, которая позволяет организовать сеть экономики роботов с помощью уже имеющихся на сегодняшний день технологий. На основании результатов экспериментов в период с 2015 по 2018 год мы предлагаем использование Ethereum как инфраструктуры, которая предоставляет минимально необходимые возможности для объединения технических и экономических параметров обязательства машины по исполнению модели поведения, согласованной с пользователем. В рамках рассмотрения платформы Робоники мы:

- расширяем возможности Ethereum для появления рынка спроса и предложения моделей поведения КФС;
- описываем операционную систему агента Робоники как интерфейс Robot Operating System [16] совместимой КФС в сеть экономики роботов;

- приводим примеры применения платформы Роботоники для реализации SDK разработчика пользовательских dapp Умных городов и Индустрии 4.0.

5.1 Ethereum Blockchain транзакции и контракты обязательств

Транзакции в Ethereum Blockchain отправляются при появлении в Роботонике спроса и предложения, удовлетворяющих друг друга по условиям. В таких ситуациях Роботоника создает сообщение для участников платформы Ethereum. Сообщение содержит параметры, необходимые для создания контрактного обязательства робота фабрикой контрактов.

5.1.1 Фабрика контрактов

В сети Ethereum существует два способа создать умный контракт:

- загрузить код контракта специальной транзакцией;
- создать контракт из другого контракта.

В первом случае внешний аккаунт компилирует контракт из исходных текстов и отправляет его код в транзакции. Это требует наличия у пользователя компилятора и набора исходных текстов, что создает трудности для конечного пользователя и систем автоматизации.

Рассмотрим более интересный второй вариант, когда контракты в сети Ethereum сами создают контракты, проверяя заданные условия. Этот механизм мы именуем Фабрикой контрактов.

```

1  contract A {}
2
3  contract B {
4      function foo() {
5          var a = new A();
6          ...
7      }
8  }
```

В приведенном примере код контракта А полностью включается в код контракта В.

5.1.2 Легковесные контракты

Иногда фабрикой создается множество умных контрактов. В таком случае при развертывании нового контракта не только аллоцируется место под пользовательские данные, но и копируется код. Это сжигает газ и засоряет блокчейн.

Рассмотрим одну очень мощную технику. В наборе инструкций EVM существует опкод DELEGATECALL, который позволяет вызвать исполнение внешнего кода в контексте текущего контракта. Такой подход называется контракт с разделяемым кодом. Рассмотрим рисунок ниже.

Пусть полезный контракт содержит только пользовательские данные, а контракт с разделяемым кодом содержит методы для их обработки. Используя опкод DELEGATECALL мы можем вызывать методы для обработки пользовательских данных из одного общего контракта, значительно облегчая реплицируемый множество раз полезный контракт. Для этого достаточно добавить fallback-метод в полезный контракт.

```

1  function () {
2      require(shared.delegatecall(msg.data));
3  }
```

Здесь shared - адрес контракта с разделяемым кодом и все транзакции, которые не могут быть обработаны полезным контрактом будут проксироваться в общий контракт с сохранением контекста.

Техника репликации контрактов с разделяемым кодом оказывается тем полезнее, чем больше и чаще планируется создавать новый контракт. Например для контракта обязательства робота этот метод позволил сэкономить 30-40% газа с каждой транзакции.

5.1.3 Создание контракта на основе отложенных подписей

Ethereum поддерживает цифровые подписи на эллиптических кривых, функция ecrecover. Такая функция позволяет восстановить публичный ключ, а именно адрес Ethereum, из подписи произвольных данных.

```
1 contract Checker is owned {
2     function byOwner(bytes32 _message, uint8 _v, bytes32 _r, bytes32 _s) view
3         returns (bool) {
4         return ecrecover(_message, _v, _r, _s) == owner;
5     }
6 }
```

В вышеприведенном примере метод byOwner возвращает true тогда и только тогда, когда сообщение подписано приватным ключом владельца контракта. Для подписи сообщения популярные клиенты сети Ethereum предоставляют метод eth.sign(account, message).

Цифровые подписи можно использовать, например, для создания контракта стороны которого не являются отправителями транзакции, но указывают свои параметры.

```
1 var promisee = ecrecover(keccak256(MSGPREFIX, askHash), uint8(_sign[1]), _sign[2],
2   _sign[3]);
3 var promisor = ecrecover(keccak256(MSGPREFIX, bidHash), uint8(_sign[5]), _sign[6],
4   _sign[7]);
5 var inst = new RobotLiability(_model, _objective, promisee, promisor, _param[0],
6   _param[1], _param[2]);
```

Фабрикой обязательств производится восстановление адреса отправителя транзакции из цифровой подписи. Так в контракт обязательства робота попадают адреса, гарантированно желающие этот контракт заключить. Доверия между отправителем транзакции и сторонами может не быть. При этом отправитель транзакции никак не может осуществить мошенничество путем подстановки адресов участников сделки, так как адрес явно не фигурирует в параметрах транзакции.

Механизм цифровой подписи совместно с умными контрактами позволяет делегировать роль отправителя транзакции третьей недоверенной стороне.

5.2 Провайдеры и маяки

Поселение экономически значимых данных сети Робоники в Ethereum Blockchain требует выделения особой группы участников, которые возьмут на себя эту функцию, а именно провайдеров сети экономики роботов.

Маяк - автономный рабочий процесс, позволяющий распределить время работы провайдеров, обслуживающих один широкополосный канал.

Выделим две группы узлов сети Робоники:

1. обычные участники сети - узлы, заинтересованные в сервисе сети Робоники:
 - отправляют сообщения спроса и/или предложения на исполнение программ КФС;
 - отправляют результаты (протокол) работы КФС по исполнению программ;

- принимают сообщения от других участников;
 - ретранслируют сообщения другим участникам.
2. провайдеры - узлы, заинтересованные в поселении экономически значимой информации сети Робономики в блокчейн Ethereum:
- принимают сообщения от других участников;
 - ретранслируют сообщения другим участникам;
 - поселяют информацию о сделке при нахождении соответствия спроса и предложения в канале;
 - поселяют информацию о результатах работы при нахождении подтверждения со стороны сети наблюдения.

Выделим два типа транзакций контракта обязательства, поддерживаемых маяками:

1. открытие (создание) контракта на основании сопоставления спроса и предложения на рынке робономики;
2. закрытие (финализация) контракта на основании результата работы, подтвержденного сетью наблюдателей.

Экономическая игра провайдера маяка заключается в:

1. поиске согласованных спроса и предложения с целью поселения транзакции создания контрактного обязательства. За что маяк будет вознаграждаться суммарной предлагаемой комиссией от пользователей;
2. поиске подтвержденных результатов работ с целью поселения транзакции финализации контрактного обязательства. За что маяк будет вознаграждаться эмиссией токена за финализацию обязательства.

5.3 Программа Робономики

Программа Робономики - рабочий процесс, поселенный в Ethereum Blockchain, с помощью которого провайдеры могут исполнить модель поведения КФС на основании технико-экономических параметров транзакции.

Исполнение программы Робономики можно условно разделить на 2 фазы:

1. Инициализация обязательства
2. Финализация обязательства

На фазе (1) исполнения программы Робономики на основании двух отложенных подписей восстанавливаются данные, необходимые для инициализации контракта обязательства. Список полей участвующих в формировании отложенной подписи:

Поле	Тип	Описание
model	bytes32	Идентификатор модели поведения КФС (хеш значение)
objective	bytes32	Параметры модели поведения КФС (хеш значение) (опционально)
token	address	Операционный токен (Ethereum адрес)
cost	uint256	Стоимость исполнения модели поведения КФС (число)
count	uint256	Количество исполнений модели поведения КФС (число)
lighthouseFee	uint256	Комиссия сети маяков (число)
validator	address	Адрес сети наблюдения (Ethereum адрес)
validatorFee	uint256	Комиссия сети наблюдателей (число)
salt	bytes32	Соль (случайные данные)
signature	bytes	Цифровая подпись отправителя (данные)

Имея 2 отложенных подписи, согласованных по модели, токenu, цене и количеству исполнений, мы можем отправить транзакцию к умному контракту фабрики контрактов Робоники, который:

1. Восстановит из подписей данные;
2. Создаст новый умный контракт обязательства;
3. Переведет необходимое обеспечение в указанных токенах;

На фазе (2) провайдер финализирует контрактное обязательство, отправляя отложенную подпись КФС, содержащую хеш от лога операций, что приводит к:

1. записи хеша в поле, отвечающее за хранение результата исполнения обязательства;
2. перевода токенов, находящихся на контракте обязательства, на адреса участников;

5.4 Операционная система агента Робоники

Базовая и наиболее важная функция операционной системы Робоники заключается в обеспечении возможности запуска программ кибер-физической системой, параметры которых задает пользователь в умном контракте.

Назначением ОС Робоники является обеспечение кибер-физических систем ПО, необходимым для задействования в своей работе экономически значимых транзакций и сообщений, получаемых или отправляемых с использованием p2p сетей.

5.4.1 Декларативная конфигурация и поколения ОС

Когда программы были простые, а 640 килобайт хватало всем, распространение ПО осуществлялось обыкновенно копированием на ПК пользователя. Практика копирования влечет за собой довольно много очевидных проблем: обновления, зависимости, хаос файловой системы и др. Для решения этих задач были придуманы специальные программы - менеджеры пакетов. Они брали на себя отслеживание обновлений, зависимостей, учет изменений и многое другое. Такие пакетные менеджеры с императивным подходом сейчас используются повсеместно. Но существует и другой путь.

Пусть каждый пакет на целевой системе хранится изолированно и не может быть перезаписан. Целостность файлов подтверждается хеш-суммой. Тогда по аналогии со значением в функциональном языке пусть пакет собирается чистой функцией, функцией без побочных эффектов. Это даст нам ряд интересных последствий:

1. повторяемость сборок; на заданных аргументах чистая функция всегда вернет один и тот же результат, а значит пакет всегда будет сформирован один и тот же;

2. кеширование; необязательно собирать пакет каждый раз, чистота дает нам возможность сохранить пакет в бинарном кеше (и дать к нему доступ его другим участникам);
3. зависимости; изменение зависимости в аргументах пакета (например, обновление версии), естественно, приводит к изменению выхода и пересборке. А это, в свою очередь, вызывает рекурсивную пересборку, если этот пакет также являлся чьей-то зависимостью.

На большинстве современных ОС частичное обновление приводит систему к неконсистентному состоянию, и она может больше не загрузиться. Такие ситуации недопустимы на АСУ робототехнических систем и сервисах реального времени. Так как при чисто функциональном подходе новая конфигурация (она называется поколение) не перезаписывает старую, всегда возможно атомарно откатить изменения и вернуться к рабочему состоянию (поколению).

Возврат к первоначальному состоянию выполняется очень быстро, так как нет необходимости скачивать пакеты прошлого поколения, они уже есть в системе в неизменном виде.

Существует огромное множество робототехнических систем, а также способов их описания и моделирования. С точки зрения практики это приводит к появлению целого набора библиотек, программ и фреймворков, несовместимых как между собой, так и не применимых к робототехническим системам различных классов. Единственным адекватным решением в такой ситуации является выработка открытого стандарта коммуникации систем управления в робототехнике.

Такой стандарт был предложен в рамках открытого проекта Robotic Operating System. Это коммуникации двух форматов: событийно-ориентированный PubSub и клиент-серверный Services. Также формализуется формат данных в рамках целого набора программного обеспечения и библиотек, что формирует открытое сообщество и инфраструктуру вокруг единого способа коммуникации систем управления.

Поддержка единого стандарта внутренней коммуникации систем управления в робототехнике является важной задачей дистрибутива Робономики. В рамках этой задачи осуществляется подготовка, портирование и включение элементов ROS в референсную реализацию ОС Робономики - AIRA.

5.4.2 Исполнение программ, описанных в умных контрактах

Контракт обязательства робота создается с целью исполнения кибер-физической системой формализованного и параметризованного пользователем поведения.

Общая модель поведения описывается в поле контракта `model`.

```
1 bytes public model;
```

Это IPFS-хеш файла, размещенного в децентрализованном хранилище данных. Робот, как и любой другой участник сети, может загрузить этот файл локально и интерпретировать его содержимое. Пример содержимого файла модели:

```
1 {
2   parity.enable = true;
3   parity.chain = "kovan";
4
5   railway-game.enable = true;
6 }
```

Это инструкции на функциональном языке Nix [4], которые включаются в конфигурацию ОС Робономики для активации ПО заданной модели поведения. Здесь, например,

активируется сервис Ethereum-ноды в сети KOVAN и сервис испытательного стенда “Game of trains”.

Модель поведения описывает структуру программного обеспечения и статические параметры, наличие которых необходимо системе для выполнения желаемого действия.

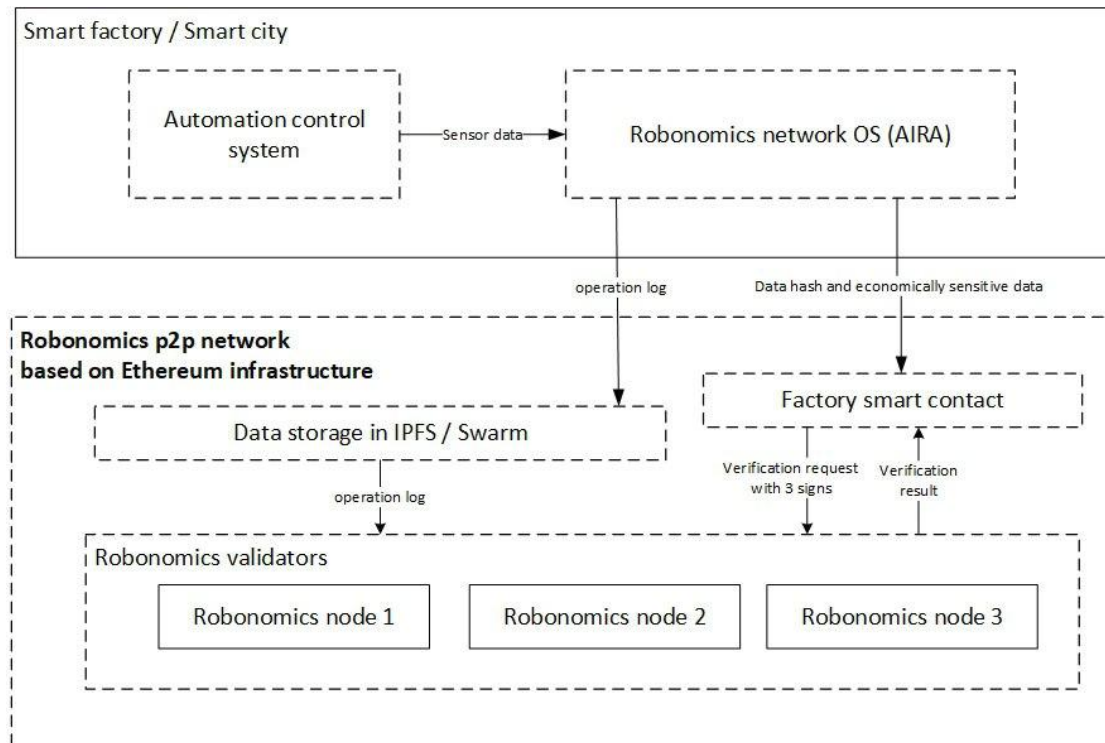
Пользовательские параметры передаются в контракте обязательства отдельно в поле `objective`. Это IPFS-хеш файла, размещенного в децентрализованном хранилище. Формат файла соответствует спецификации ROSBAG v2 (<http://wiki.ros.org/Bags/Format/2.0>) и определяет динамические параметры поведения кибер-физической системы. BAG-файл воспроизводится в момент исполнения обязательства роботом и поставляет в работающую систему сообщения с данными-параметрами, которые могут играть как роль триггеров на исполнение, так и уточнять характеристики желаемой услуги или продукта. Например, сообщение для фабрики стенда “Industry 4.0 in use” содержит сообщение:

Топик	Тип	Значение
model	bytes32	Идентификатор модели поведения КФС (хеш значение)
-	-	data: “blue”

Событие, связанное с получением сообщения, служит триггером на исполнение умной фабрикой обязательства по производству продукта, а содержимое сообщения уточняет его характеристики.

В момент начала воспроизведения параметров пользователя также запускается ведение протокола операций кибер-физической системы. Результаты работы кибер-физической системы в виде лога операций в формате ROSBAGv2 загружаются в децентрализованное хранилище. А IPFS-хеш (адрес) файла отправляется в умный контракт в качестве подтверждения выполненной кибер-физической системой работы. Задачей наблюдения за исполнением роботами их обязательств занимается независимая сеть узлов, выполняющая интерпретацию протоколов работы в соответствии со связанным контрактом обязательства.

5.5 Наблюдающая сеть



Робоника способна наблюдать за исполнением обязательств роботов с помощью использования моделей верификации результата работы машины. Для этого необходимо, чтобы в контракте обязательства была указана желаемая в использовании наблюдающая сеть и модель верификации.

Модель поведения в соответствии с framework'ом, применяемым инженером и роботом, является основным элементом свершения работы.

5.5.1 Алгоритм работы контракта наблюдателя

Инициализация контракта происходит так же как и контракта Маяка, на контракт участниками вносится обеспечение, пропорционально которому выдаются квоты на верификацию обязательств.

1. В канале результатов появляется сообщение с результатом для обязательства, требующего подтверждение валидатором.
2. Участник сети наблюдения с маркером подписывает сообщение с результатом и своим решением и отправляет его в канал результатов.
3. Маяк ловит сообщение с подтверждением валидатора, но поселить его не может N блоков.
4. Если остальные участники сети наблюдения не согласны с текущим владельцем маркера, они могут опубликовать сообщение в канал результатов со своим решением.

5. Если по решению не согласному с текущим владельцем маркера набирается BFT, то эта транзакция может быть сразу поселена в контракт обязательства, а текущий владелец маркера лишается квоты.
6. За верификацию контракта обязательства участнику сети наблюдения переводится комиссия в XRT.

6 Токен Робоники, XRT

Задача токена Робоники - обеспечить работу децентрализованной сети по обслуживанию Умных городов и Индустрии 4.0 в инфраструктуре Ethereum. Для достижения данной цели в экономике токена необходимо отразить стимулы выполнения полезной функции сети независимыми провайдерами. Данные стимулы должны быть распределены между эмитсией и комиссией таким образом, чтобы обеспечивать пропускную способность Робоники в Ethereum в зависимости от цены токена XRT, а также мотивировать провайдеров запускать программу Робоники в EVM с данными, предлагаемыми пользователями.

6.1 Эмиссия за исполнение программы Робоники в Ethereum компьютере

Существует интересное свойство эмиссии в Bitcoin и Ethereum: вне зависимости от того, есть ли транзакции или нет, блоки всё равно создаются. Данное относительное “безразличие” позволяет гарантировать пропускную способность сети в соответствии с протоколом коммуникации. Эмиссия является ключевым стимулом для майнеров поддерживать заданную пропускную способность, не дожидаясь накопления выгодной суммарной комиссии в блоке.

Представление Робоники как программы для EVM позволяет связать эмиссию XRT токенов с величиной утилизируемого газа как метрикой, являющейся общей для оценки вычисления и сохранения данных любой программы в Ethereum компьютере. Это обеспечит зависимость пропускной способности Робоники в общей инфраструктуре Ethereum от цены токена XRT.

Введем базовую техническую единицу Робоники под названием Wiener (1 XRT = $10^9 Wn$) и определим для нее следующее назначение:

Emission of 1 wn = 1 gas utilized by Robonomics

Такая модель означает:

1. 1 XRT не равен 1 ETH, так как 1 gas может стоить различное количество wei.
2. Стоимость 1 wn должна покрывать расходы провайдера на утилизацию 1 единицы gas в Ethereum.
3. Эмиссию Wn должна иметь возможность выполнять только программа Робоники.
4. Повышение цены XRT будет способствовать увеличению пропускной способности Робоники.
5. Общая капитализация XRT может быть больше капитализации ETH, так как несёт в себе дополнительную функцию, кроме утилизации газа в EVM.

Для того, чтобы достичь показателя *Emission of 1 Wn = 1 gas utilized by Robonomics*, мы предлагаем:

1. Во время TGE с помощью Голландского аукциона найти начальный множитель эмиссии W_n для покрытия минимально необходимых расходов провайдеров на обеспечения присутствия Робономики в инфраструктуре Ethereum на начальный момент эмиссии.
2. Внедрить этап становления сети, в течение которого множитель эмиссии будет алгоритмически снижаться в зависимости от общего объема утилизированного газа в исполнение программы Робономики в EVM.
3. В последующем обеспечить постоянную эмиссию W_n за каждую утилизированную единицу газа в Ethereum Робономикой.

6.1.1 TGE и Голландский аукцион

В начальный момент запуска Робономики мы предлагаем выпустить 10,000,000 XRT, из которых от 5% до 77% будет доступно для голландского аукциона. Мы ставим цель в N ETH для сбора средств на счет специально создаваемого Фонда экономики роботов, который будет развивать базовую концепцию представления КФС как экономически автономных агентов. Чем больше моделей экономически автономных агентов, доступных для использования в любой точке планеты, тем выше вероятность задействования Робономики в целом.

В результате аукциона все участники получают токены по минимально возможной цене, а Робономика получит начальное соотношение XRT к ETH. Совместив эти данные со статистикой текущей цены газа, мы сможем определить начальный множитель эмиссии на период становления Робономики в инфраструктуре Ethereum, чтобы покрывать расходы провайдеров по конкурентной цене при условии естественно меньшей капитализации Робономики относительно Ethereum на начальном этапе. По нашим расчетам на весну 2018 года минимальной конкурентной ценой в сети Ethereum является цена в 2 gwei за 1 gas.

6.1.2 Становление

Период становления Робономики в инфраструктуре Ethereum мы разделяем на 5 эпох, каждая из которых оценена статистически в суммарном объеме утилизированного газа в Ethereum. По своему завершению эпоха алгоритмически снижает множитель эмиссии таким образом, чтобы по завершению периода становления выполнялось условие:

$$Emission\ of\ 1\ w_n = 1\ gas\ utilized\ by\ Robonomics$$

То есть множитель должен снизиться до единицы, исчезнуть в прошлом.

Ниже приводится таблица параметров пяти эпох становления Робономики:

Epoch	Utilized gas	E(i), W_n	E(i), XRT
1	3,47E+12	2,08E+13	2,08E+06
2	3,47E+12	1,39E+13	1,39E+06
3	3,47E+12	9,26E+12	9,26E+05
4	3,47E+12	6,17E+12	6,17E+05
5	3,47E+12	4,12E+12	4,12E+05
Total	1,74E+13	5,43E+13	5,43E+06

Нами заложен объем газа в блоке как постоянная величина, так как предсказать его рост на сегодня фактически невозможно. Соответственно, мы можем только судить о том, что прохождение каждой эпохи означает утилизацию процентного количества газа по данным о размере блока в Ethereum на период весны 2018 года. То есть при завершении пятой эпохи мы можем говорить о том, что Робономика справилась с показателем в один годовой объема газа, который был доступен в сети Ethereum весной 2018 году. В любом случае данной статистики будет достаточно для объективной оценки роста Робономики относительно момента проведения TGE в 2018 году.

6.1.3 Дальнейшая эмиссия

После прохождения стадии становления Робономика выйдет на постоянную величину эмиссии, количественно равную желаемому показателю $Emission\ of\ 1\ wn = 1\ gas\ utilized\ by\ Robonomics$.

Данная линейная зависимость будет способствовать обеспечению минимальной пропускной способности, которую способны дать провайдеры после выравнивания капитализации Робономики без наличия комиссии.

6.2 Комиссии за запуск программы Робономика в Ethereum инфраструктуре

Первое отличие исполнения программы Робономики в EVM с настоящими участниками от симуляции заключается в неопределенном промежутке времени с момента создания контрактного обязательства для КФС до момента отправки результата работы и финализации выполнения программы на основании этой транзакции. Отсюда следует, что КФС должны быть заинтересованы в оплате комиссии сети, покрывающей расходы провайдера на работу в информационных каналах для поиска согласованных спроса с предложением КФС, а также издержек на поселение транзакции по созданию контрактного обязательства в Ethereum Blockchain.

6.3 Дополнительные задачи XRT токена

6.3.1 Наблюдающие сети существуют за счет комиссии пользователей

Мотивировать провайдеров финализировать выполнение программы Робономики с учетом наблюдающей сети или без нее нет необходимости. Любой провайдер с удовольствием финализирует выполнение программы Робономики ради эмиссии.

Таким образом, комиссия пользователя услуги КФС может быть использована исключительно для оплаты проверки результатов исполнения модели поведения КФС.

6.3.2 XRT на счету маяков как механизм доверия к Робономике

В нулевой момент существования Робономики провайдеры будут заинтересованы в создании “индивидуальных” маяков с минимальной ставкой XRT, в формате обеспечения, на контракт маяка. С момента появления спроса и предложения в Робономике интерес к работе на конкретном маяке будет тем выше, чем больше суммарная комиссия, которую может получить провайдер с информационного канала, обслуживаемого маяком. Со временем провайдеры начнут консолидироваться вокруг тех маяков, где доля обеспечения провайдера, размещенная на контракте маяка, будет давать достаточное время работы, чтобы получить дополнительную выгоду.

В конечном результате, чем выше будет суммарная комиссия с определенных информационных каналов, тем большую долю XRT провайдеры готовы будут заморозить. Чем больше объем замороженных на маяке средств, тем выше доверие остальных участников к работе данного маяка и к Робономике в целом.

6.3.3 XRT как количественное отражение капитала сети

Капитал Робономики есть не что иное, как отражение ценности сети, использующей Ethereum инфраструктуру для построения коммуникации с экономически автономными КФС.

Рост капитала такой сети, с одной стороны, будет связан с ростом количества людей в обществе, принимающих идею представления КФС как автономных экономических агентов, а с другой - взглядом на Ethereum как на инфраструктуру, подходящую для экономически значимой коммуникации по академическим и промышленным стандартам автоматизации.

6.3.4 Рынок кибернетики

В долгосрочной перспективе, являясь токеном необходимым КФС для заключения контрактных обязательств и верификации их результата, XRT превратится во внутреннюю валюту рынка контрактных обязательств роботов для роботов. Рынок, в котором каждое исполнение программы будет зациклено на исполнение программы.

6.3.5 Выводы о движении XRT в Робономике

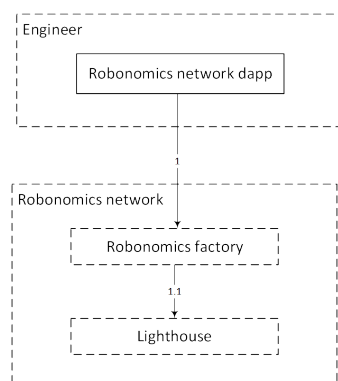
Итак, мы получаем, что эмиссия XRT призвана поддерживать постоянную готовность Робономики выполнить коммуникацию с КФС как с автономным экономическим агентом в инфраструктуре Ethereum. Комиссии создают стимул для использования пользовательских данных в исполнении программы Робономики провайдерами.

6.3.6 Denomination

Значение	1	10^3	10^6	10^9
Наименование в Robonomics platform	wiener	coase	glushkov	robonomics token

7 Описание работы Робономики

7.1 Предложение новой модели поведения инженером



Lighthouse - автономный рабочий процесс, позволяющий распределить время работы провайдеров, обслуживающих конкретный широкополосный канал.

Robonomics factory - автономный рабочий процесс, позволяющий создать новый маяк или обязательство, используя соответствующие библиотеки, сохраненные в Ethereum Blockchain.

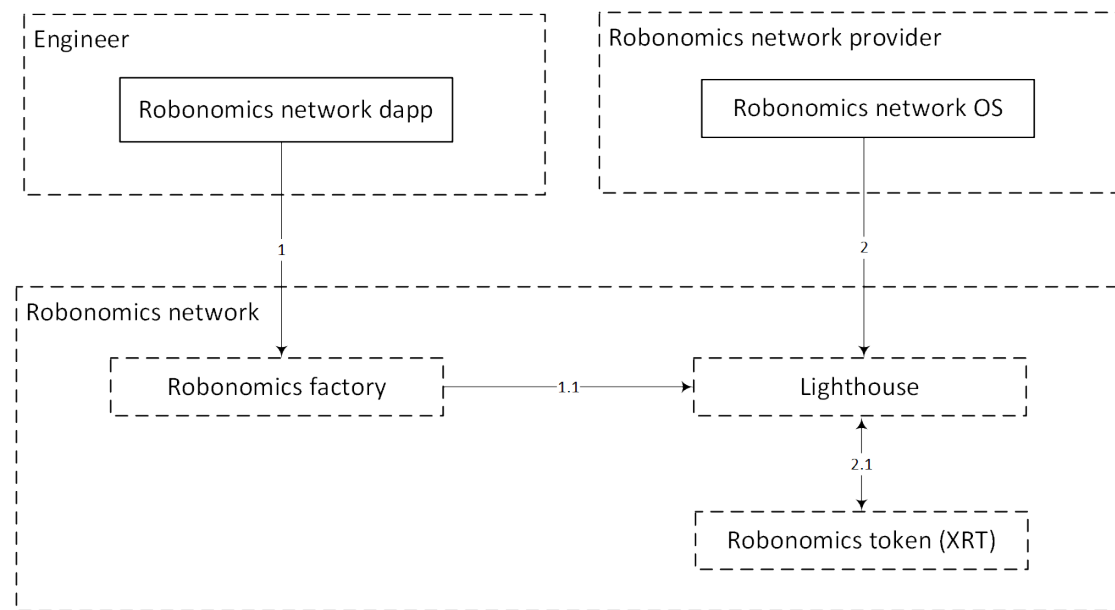
Организации, создающие кибер-физические системы, могут быть заинтересованы в представлении их роботов как сервисов, способных учитывать собственную экономику: от получения технических и финансовых параметров перед оказанием услуги до самостоятельного обеспечения своей работы необходимыми ресурсами и обслуживанием за счет средств, находящихся под управлением КФС.

Для подобных задач инженер может создать ROS-совместимую модель поведения, которую он хочет предложить использовать владельцам кибер-физических систем, чтобы их роботы начали работать как экономически автономная система.

Инженер с помощью dapp подключается к сети Робоники и отправляет в Ethereum транзакцию (1), содержащую хэш модели поведения ROS-совместимой кибер-физической системы.

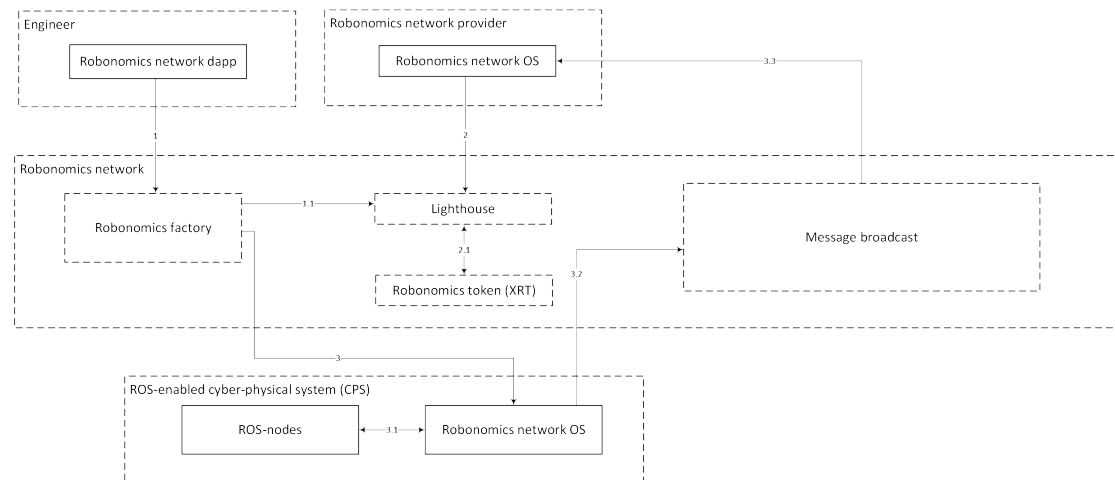
На основе транзакции от инженера (1) происходит внутренний вызов (1.1) создания нового автономного рабочего процесса Маяка в Ethereum Blockchain.

7.2 Начало работы провайдера сети Робоники



Провайдер сети Робоники отправляет транзакцию (2) к созданному (1) маяку, передав некоторое количество токенов XRT (2.1) как обеспечение, используемое в распределении времени работы на маяке среди провайдеров.

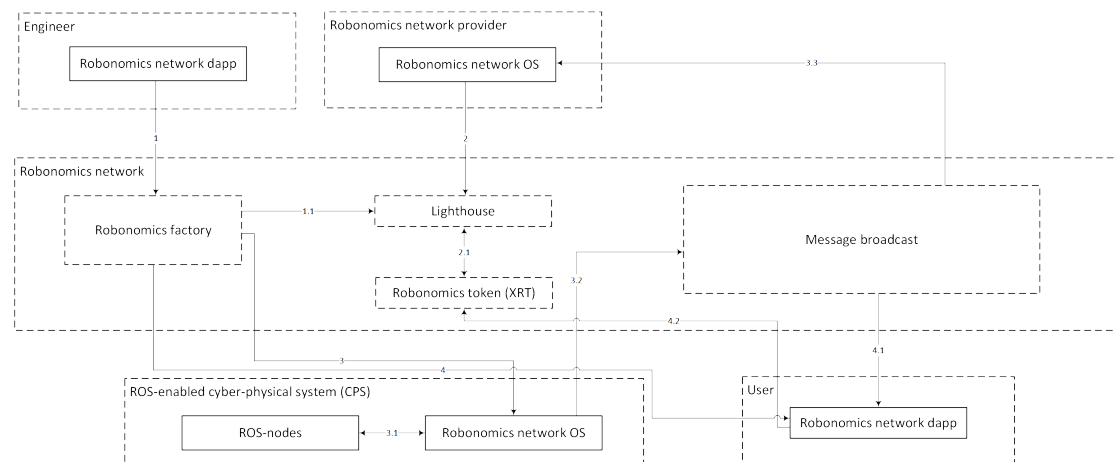
7.3 Начало работы кибер-физической системы как сервиса



Robonomics network OS получает список созданных маяков от фабрики Робоники (3). Используя информацию из описания маяков и опрашивая свои узлы (3.1), КФС начинает публиковать свои предложения об оказании услуг в каналы широковещательной рассылки Робоники (3.2).

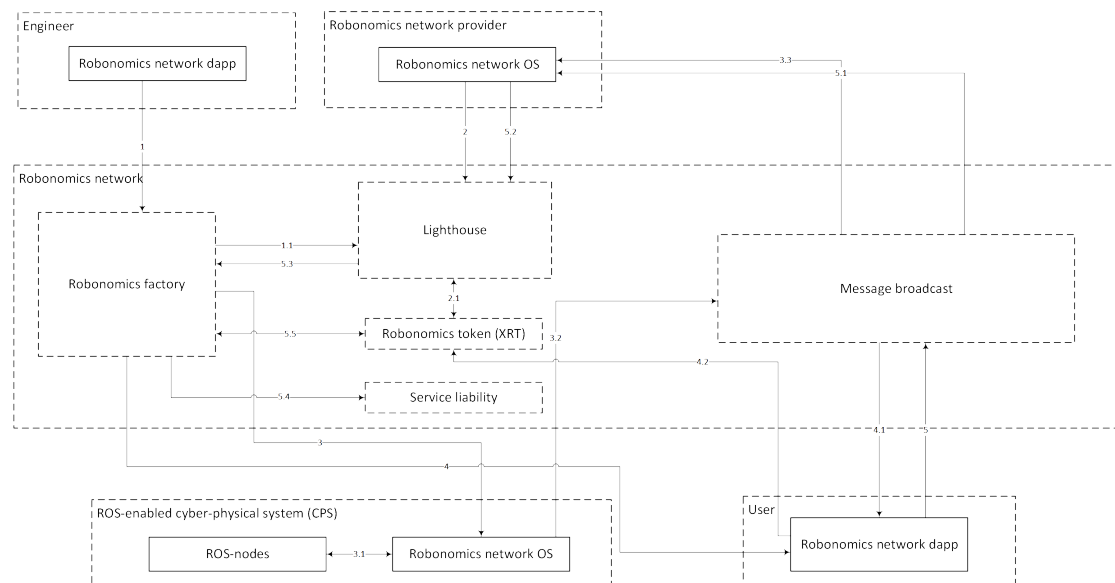
Провайдер сети Робоники получает сообщение через канал широковещательной рассылки о появлении нового предложения от КФС (3.3).

7.4 Появление пользователя в сети Робоники



Пользователь получает список широковещательных каналов Робоники (4) с помощью dapp и начинает слушать сообщения в данных каналах (4.1). Пользователь, убедившись, что в сети Робоники есть интересующая его услуга от КФС (3.2), отправляет транзакцию в Ethereum Blockchain для разрешения снятия XRT токенов автономным рабочим процессом фабрики Робоники (4.2).

7.5 Создание обязательства на исполнение сервиса кибер-физической системой



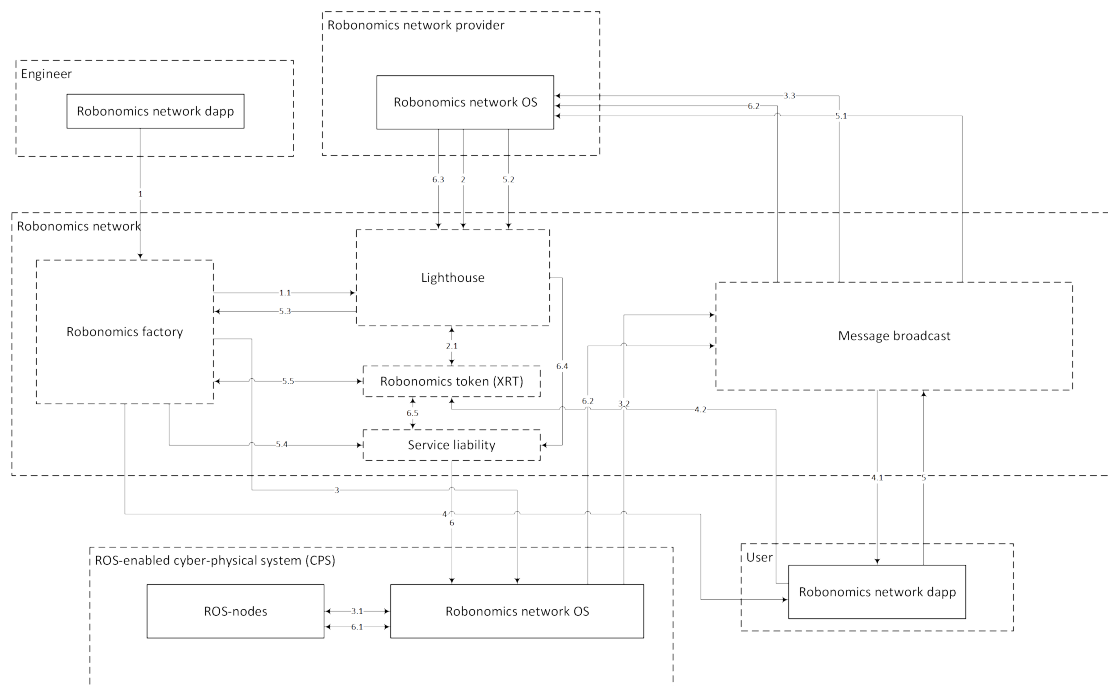
Пользователь, имея достаточное количество XRT токенов для отправки запроса на интересующую его услугу, может отправить в широковещательную рассылку сообщение о своем запросе (5) на ранее опубликованную КФС услугу (4.2).

Провайдер сети Робоники, получив сообщение пользователя (5.1) и имея локально сохраненное сообщение от КФС (3.3), которые согласованы по значению хэша от модели поведения ROS-совместимой КФС и стоимости оказываемой услуги, обращается к автономному рабочему процессу Маяка (5.2), передавая отложенные подписи обеих сторон.

Маяк обращается к фабрике Робоники (5.3) для создания обязательства на оказание услуги (5.4), фабрика же, создав контракт обязательства и зная его адрес в Ethereum Blockchain, обращается к контракту XRT (5.5) для:

1. Перевода стоимости оказываемой услуги со счёта пользователя (4.2) на счет контракта;
2. Эмиссии XRT токенов на счет контракта в объеме общего количества, требуемого для утилизации газа, умноженного на стартовый фактор, в Ethereum Blockchain.

7.6 Исполнение обязательства кибер-физической системой



С помощью Robonomics network OS кибер-физическая система получает уведомление о созданном для нее контракте обязательства, в котором можно узнать хэш модели поведения и хэш параметров (переменных) модели поведения (6).

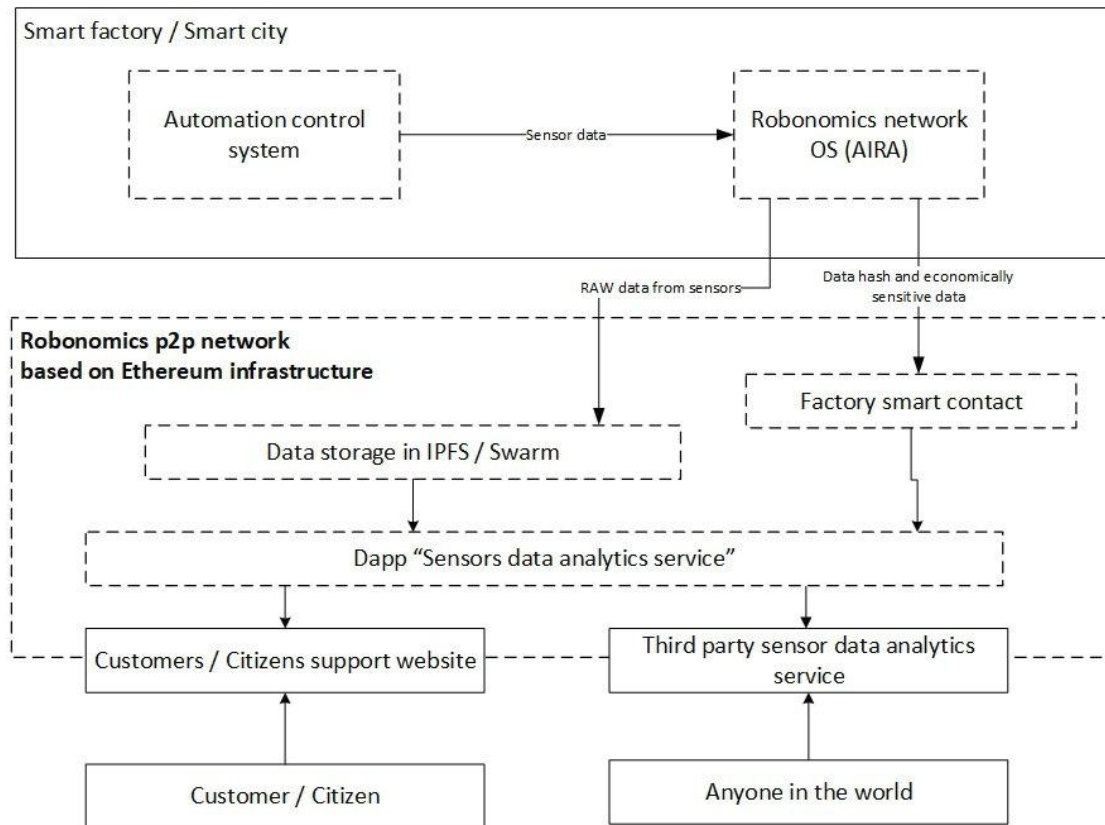
Загрузив на исполнение модель поведения и параметры (6.1), КФС с помощью ROS-bag приступает к исполнению полученного обязательства.

Во время исполнения КФС формирует лог операций и после завершения работы отправляет его хэш в информационный канал Робоники, обслуживающий данную модель поведения (6.2).

Провайдер сети Робоники получает хэш лога (6.2) и отправляет транзакцию о выполнении обязательства со стороны КФС (6.3) в Ethereum Blockchain к Маяку (6.3). Маяк создает внутреннюю транзакцию к соответствующему обязательству на выполнение услуги и записывает туда хэш лога операций КФС (6.4). После этого с контракта обязательства высвобождается вознаграждение за оказанную услугу на адрес КФС, а также эмиссия XRT на адрес того провайдера, который отправил транзакцию в Маяк.

8 Приложения платформы Робоники

8.1 Создание доверия к продукции умных городов и умных фабрик



Вопрос доверия между потребителем и умной фабрикой заключается в том, как потребитель может быть уверен в качестве продукта, произведенного умной фабрикой. Продукты в магазине могут быть очень похожи, но иметь очень разную историю происхождения. Мы предлагаем протоколировать технологический процесс, размещая его в децентрализованной сети. Хеш от лога операций КФС сохраняется в блокчейн, наделяя историю продукта свойством иммутабельности. Потребитель уверен в продукте, так как его история не может быть изменена в будущем.

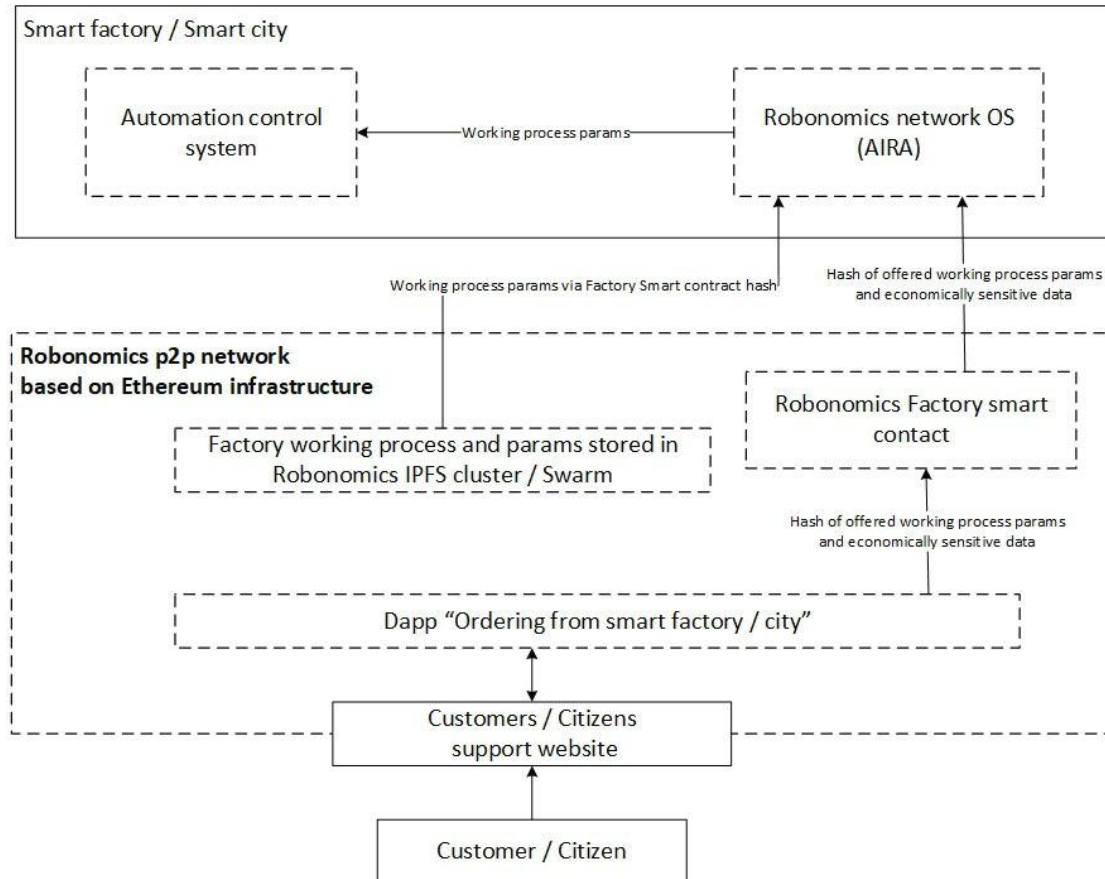
https://github.com/airalab/robonomics_comm/blob/master/robonomics_liability/src/robonomics_liability/executor.py#L50

Здесь происходит сбор и аккумуляция данных производственного процесса в момент исполнения обязательства умной фабрикой.

```
1 {  
2     msg.result = self.ipfs.add(result_file)['Hash']  
3     self.complete.publish(msg)  
4 }
```

Лог операций сохраняется в IPFS сеть, а его хеш публикуется в контракт обязательства и не может быть изменен в будущем.

8.2 Запуск умной фабрики пользователем



Проблема доступа к фабрике как к сервису для потребителя заключается в том, чтобы устранить посредника, который как-либо может повлиять на их взаимодействие. Умный контракт является подходящим объектом.

https://github.com/airalab/robonomics_comm/blob/master/robonomics_liability/src/robonomics_liability/executor.py#L34

Здесь умная фабрика слушает события от фабрики умных контрактов, принимая обязательства, в которых она указана как исполнитель.

```
1 rospy.logdebug('Getting objective %s...', msg.objective)
2 self.ipfs.get(msg.objective)
3
4 player = Player(msg.objective)
5 player.start()
```

Когда обязательство взято, умная фабрика начинает его исполнение, скачивая параметры обязательства из IPFS и воспроизводя их последовательность локально.

8.3 Управление экономикой роботов с помощью капитала

Представим множество экономически самостоятельных киберфизических систем или просто роботов как вектор (R), а капитал (инвестиции) — как вектор (K). Здесь вектор R =

$R_i, i = \overline{1, n}$, где R_i — количество роботов на рынке i , n — количество рассматриваемых рынков. А вектор $K = K_j, j = \overline{1, n}$, где K_j — инвестиции в j -й рынок. Сформулируем описание модели пропорционального капиталу распределения роботов на рынке следующим образом:

$$R'[k] = K[k] * \frac{\sum R_i[k]}{\sum K_i[k]}$$

$$e[k] = R'[k] - R[k - 1];$$

Здесь:

- $R'[k]$ — желаемое распределение роботов на рынках [штук];
- $R[k - 1]$ — количество роботов, уже присутствующих на данном рынке капитала;
- e — вектор ошибок распределения (отклонение от желаемого значения распределения) [штук];
- $i \in \overline{1, n}$ - индекс рынка;
- k - шаг дискретного времени;
- $\sum R_i[k]$ - количество роботов на всех рассматриваемых рынках на шаге k ;
- $\sum K_i[k]$ - объём инвестиций в рассматриваемые рынки на шаге k .

Исходя из выше описанного, можно сформулировать закон управления экономикой роботов следующим образом:

Каждый робот должен стремиться к участию на том рынке, где ошибка распределения выше [22].

При включении очередного робота в экономику рынок для размещения своего предложения этим роботом выбирается из равенства:

$$i = \text{max}(e[k]), \text{ что приводит к } e_i[k] \rightarrow 0.$$

Здесь $\text{max}(e)$ - индекс максимального элемента вектора отклонения.

8.3.1 Референсная имплементация

Реализация алгоритма управления экономикой роботов содержится в пакете `robonomics_control` https://github.com/airalab/robonomics_comm/tree/master/robonomics_control.

Здесь алгоритм линейного распределения основан на информации из умного контракта о распределении капитала и информации с рынков о лотах от кибер-физических систем.

https://github.com/airalab/robonomics_comm/blob/master/robonomics_control/src/robonomics_control/distribution.py

8.4 Токенизированные ценности на основе труда робота

Рассмотрим сценарий выпуска карбоновых единиц сенсором в умном городе. Сенсор наделяется правом эмиссии токена карбоновых единиц, если загрязнение воздуха ниже некоторого значения.

Пусть сенсор в результате обязательства по измерению степени загрязнения воздуха совместно с результатами присылает количество освобожденных карбоновых единиц. Расширим стандартный контракт обязательства методом, принимающим совместно с результатом работы количество единиц для эмиссии.


```
1 function setResult(bytes32 _result, uint256 _emission);
```

Таким образом, значения для выпуска карбоновых единиц сохраняются в контракте обязательства. И при условии, что сеть наблюдателей подтвердит корректность измерений сенсора, по контракту обязательства будет эмиссировано соответствующее число токенов.

```
1 function confirm() {  
2     ...  
3     carbonToken.emission(emission);  
4 }
```

Здесь важно, что контракт обязательства добавляется в группу, которой доступна эмиссия токена, совершенная фабрикой на этапе создания. Тогда фабрика контрактов обязательств обязана обладать информацией о соответствии заданной модели поведения такому токеноу, который может быть эмиссирован при успешном исполнении обязательства.

9 Масштабируемость

Робономика проектируется для представления таких крупных кибер физических систем, как, например, целая фабрика или город. Действующая на сегодня пропускная способность сети Ethereum является достаточной, чтобы обеспечить исполнения более 1,000 контрактных обязательств в сутки. К примеру, этого достаточно для: Организации ежедневного прямого заказа покупателями автомобилей на сайтах нескольких автоконцернов таких, как BMW, Porsche, Avtovaz. Регистрации регулярных маршрутов для беспилотной логистики текущих промышленных зон мира. Ежедневной публикации отчетов о состоянии окружающей среды от сенсорных сетей всех городов мира с населением свыше 1,000,000 человек.

10 Сегмент Робономики в сети Ethereum

С внедрением возможности создать собственный сегмент в сети Ethereum Робономика может выделиться в сегмент с повышенной пропускной способностью, а также позволить многократно повысить количество возможных исполнений роботами контрактных обязательств для того, чтобы покрыть запрос всей Индустрии 4.0 со стоимостью сделок свыше 1,000 \$.

11 Заключение

Платформа Робономики ставит своей целью решение социально-экономических задач тотальной роботизации массового производства, жизни городов и логистики. Основным применением платформы можно считать задачи, связанные с созданием доверия к услугам и продукции умных городов и фабрик, предоставления прямого доступа пользователей к автономным кибер-физическим системам и управление мультиагентными системами с помощью капитала.

Платформа Робономики позволит расширить возможности инфраструктуры сети Ethereum с целью ее использования в таких областях, как Индустрия 4.0, IoT, умные города.

Список литературы

- [1] Buterin и Vitalik. «Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform». В: *Ethereum* (2014).
- [2] R. H. Coase. «The Nature of the Firm». В: *Economica* (1937). ISSN: 14680335. DOI: [10.1111/j.1468-0335.1937.tb00002.x](https://doi.org/10.1111/j.1468-0335.1937.tb00002.x).
- [3] Maria Carmela De Gennaro и Ali Jadbabaie. «Decentralized control of connectivity for multi-agent systems». В: *Decision and Control, 2006 45th IEEE Conference on*. 2006, с. 3628–3633.
- [4] Eelco Dolstra, Andres Löb и Nicolas Pierron. «NixOS: A purely functional Linux distribution». В: *Journal of Functional Programming* (2010). ISSN: 09567968. DOI: [10.1017/S0956796810000195](https://doi.org/10.1017/S0956796810000195).
- [5] *Enter the robots: automation fills gaps in east Europe's factories*. 2018. URL: <https://www.reuters.com/article/us-easteurope-automation-insight/enter-the-robots-automation-fills-gaps-in-east-europes-factories-idUSKBN1FR1ZK>.
- [6] N Jazdi. «Cyber physical systems in the context of Industry 4.0». В: *2014 IEEE Automation, Quality and Testing, Robotics* (2014). ISSN: 1844-7872. DOI: [10.1109/AQTR.2014.6857843](https://doi.org/10.1109/AQTR.2014.6857843).
- [7] Hyoung Seok Kang и др. «Smart manufacturing: Past research, present findings, and future directions». В: *International Journal of Precision Engineering and Manufacturing - Green Technology* (2016). ISSN: 21980810. DOI: [10.1007/s40684-016-0015-5](https://doi.org/10.1007/s40684-016-0015-5).
- [8] Aleksandr Kapitonov и др. «Blockchain-based protocol of autonomous business activity for multi-agent systems consisting of UAVs». В: *Research, Education and Development of Unmanned Aerial Systems (RED-UAS), 2017 Workshop on*. 2017, с. 84–89.
- [9] Leo Kelion. *Could driverless cars own themselves?* 2015. URL: <http://www.bbc.com/news/technology-30998361>.
- [10] Siavash H. Khajavi, Jouni Partanen и Jan Holmström. «Additive manufacturing in the spare parts supply chain». В: *Computers in Industry* (2014). ISSN: 01663615. DOI: [10.1016/j.compind.2013.07.008](https://doi.org/10.1016/j.compind.2013.07.008).
- [11] Heiner Lasi и др. «Industry 4.0». В: *Business and Information Systems Engineering* (2014). ISSN: 18670202. DOI: [10.1007/s12599-014-0334-4](https://doi.org/10.1007/s12599-014-0334-4).
- [12] Paulo Leitão и Stamatis Karnouskos. *Industrial Agents: Emerging Applications of Software Agents in Industry*. 2015. ISBN: 9780128004111. DOI: [10.1016/C2013-0-15269-5](https://doi.org/10.1016/C2013-0-15269-5).
- [13] Yabo Liu и др. «Multi-robot coordination in complex environment with task and communication constraints». В: *International Journal of Advanced Robotic Systems* (2013). ISSN: 17298806. DOI: [10.5772/54379](https://doi.org/10.5772/54379).
- [14] László Monostori. «Cyber-physical production systems: Roots, expectations and R&D challenges». В: *Procedia CIRP*. 2014. ISBN: 2212-8271. DOI: [10.1016/j.procir.2014.03.115](https://doi.org/10.1016/j.procir.2014.03.115).
- [15] Mikkel Rath Pedersen и др. «Robot skills for manufacturing: From concept to industrial deployment». В: *Robotics and Computer-Integrated Manufacturing* (2016). ISSN: 07365845. DOI: [10.1016/j.rcim.2015.04.002](https://doi.org/10.1016/j.rcim.2015.04.002).
- [16] Morgan Quigley и др. «ROS: an open-source Robot Operating System». В: *Icra* (2009). DOI: <http://www.willowgarage.com/papers/ros-open-source-robot-operating-system>.
- [17] *Robot density rises globally - International Federation of Robotics*. 2018. URL: <https://ifr.org/ifr-press-releases/news/robot-density-rises-globally>.

- [18] T. Stock и G. Seliger. «Opportunities of Sustainable Manufacturing in Industry 4.0». В: *Procedia CIRP*. 2016. ISBN: 22128271 (ISSN). DOI: [10.1016/j.procir.2016.01.129](https://doi.org/10.1016/j.procir.2016.01.129).
- [19] Norbert Wiener. «Cybernetics : Control and Communication in the Animal and the Machine –2nd. ed». В: *212 p. Cambridge, Mass.: The MIT press, 1961. includes index* (1961).
- [20] Lijun Zhu и Zhiyong Chen. «Robust homogenization and consensus of nonlinear multi-agent systems». В: *Systems and Control Letters* (2014). ISSN: 01676911. DOI: [10.1016/j.sysconle.2013.12.007](https://doi.org/10.1016/j.sysconle.2013.12.007).
- [21] Глушков Виктор. «Макроэкономические модели и принципы построения ОГАС.» В: (1975).
- [22] Лоншаков Сергей. *Как мы учим умные фабрики быть отзывчивыми с помощью протокола робономики*. 2017. URL: <http://ensrationis.com/smart-factory-and-capital/>.

Приложение 1: Вопросы на стыке робототехники и экономической теории

Тотальная роботизация. Приход роботов в жизнь каждого человека неизбежен. Машины способны выполнять недоступные человеку задачи, они эффективнее во многих видах операционной деятельности и экономят человеку время в повседневных делах уже сегодня.

Развитие робототехники достигло такого этапа, на котором появились задачи коммуникации между физически или логически выделенными автономными агентами - роботами, имеющими право решать, какие действия являются целесообразными в рамках их существования [8]. Технологии, находящие применение в мире машин, расширяют доступный роботу набор принимаемых решений, что с каждым разом повышает уровень его автономности.

Наиболее ярко задачи коммуникации автономных роботов раскрываются в таких идеях, как Индустрия 4.0 и Интернет вещей. Ниже приведены наиболее значимые на взгляд авторов данной работы вопросы участия автономных роботов в жизни человека:

- Как обеспечить доверие к продукции роботизированных сервисов?
- Как обеспечить передачу права собственности потребителю, если в процессе производства и логистики не участвует человек?
- Как полностью автоматизированному предприятию за окном понять изменяющиеся потребности человека?
- Как обеспечить прямое взаимодействие между роботами двух различных корпораций?
- Как взимать налоги с деятельности роботов и что считать отдельной роботизированной единицей?

Варианты ответов на данные вопросы влекут за собой различные риски повышения нестабильности и возможного коллапса мировой системы - это вызов на пути построения общества, в котором нет места рабскому труду человека.

Приложение 2: Дистрибутив AIRA

Опорной (референсной) реализацией ОС Робономики является проект AIRA (<https://github.com/airalab/aira>). Дистрибутив AIRA основан на дистрибутиве NixOS GNU/Linux и развивается открытым сообществом, в которое входят разработчики Airalab.

NixOS Основу дистрибутива составляет ответвление от базы пакетов nixpkgs (<https://github.com/airalab/airalab-channels>), в которое командой Airalab добавлены важные пакеты инфраструктуры Ethereum (такие, как parity, web3.py); поддержка Robot Operating System, а также фундаментальные пакеты коммуникационного стека сети Робономики: robonomics_market, robonomics_liability.

ROS Минимальная поддержка ROS обеспечена в рамках метапакета ros_comm с задействованием необходимого базового стека ПО для сериализации и обмена сообщениями между участниками сети ROS, вызова сервисов и интеграции в единое пространство имен rosgraph.

На базе дистрибутива формируются установочные образы стендов Фабрики экспериментов (<https://aira.life/cases/>): Game of trains, Industry 4.0 in use.

Приложение 3: Внешнее API AIRA

/market/incoming/ask	Входящие сообщения со спросом из информационного канала.
/market/incoming/bid	Входящие сообщения с предложением из информационного канала.
/market/sending/ask	Исходящие сообщения спроса в информационный канал.
/market/sending/bid	Исходящие сообщения предложения в информационный канал.
/market/signing/ask	Исходящие сообщения спроса на подпись.
/market/signing/bid	Исходящие сообщения предложения на подпись.

Приложение 4: Программа ДАО

Представление экономики роботов, как программы для децентрализованного компьютера Ethereum Чтобы получить децентрализованную службу, необходим децентрализованный компьютер и программа, адаптированная под его специфику. Рабочие процессы, позволяющие задействовать инфраструктуру сети Ethereum для Умных городов и Индустрии 4.0, могут быть представлены в виде программы для компьютера Ethereum.

Представим, что у нас есть программа с названием “Сеть экономики роботов”, мы загружаем её в Ethereum Blockchain на хранение. Программа позволяет участникам инфраструктуры Ethereum включить “Сеть экономики роботов” на своих виртуальных машинах и тем самым организовать новую службу в децентрализованной среде, выступив её провайдером.

Представление экономики роботов, как ДАО Автономные рабочие процессы и их исполнители образуют ДАО. Исполнитель не может повлиять на работу ДАО, он может лишь следовать инструкциям. Экономические стимулы должны поддерживать интерес исполнителя для полного выполнения рабочего процесса. Рабочий процесс не может быть изменен без последствий в Ethereum Blockchain.