

Musereum

Artem Aler Dan

4 декабря 2017 г.

Оглавление

1	Введение	3
2	Обзор индустрии	4
2.1	Доходы от распространения музыкального произведения	5
2.2	Структура музыкальной индустрии	8
2.3	Основные проблемы музыкальной индустрии	11
2.3.1	Отсутствие доступной и достоверной информации о принадлежности прав	11
2.3.2	Непрозрачность транзакций в индустрии	12
2.3.3	Отсутствие эффективных средств контроля за использованием музыки .	12
2.3.4	Черные ящики	13
2.3.5	Пиратство	13
2.3.6	Сложности монетизации музыкального контента	13
2.4	Решение вышеописанных проблем проектом Musereum	14
3	Platform Overview	19
3.1	Tools	19
3.2	Roles	20
4	Technical Description of Platform	21
4.1	Overview	21
4.1.1	Naming convention	22
4.1.2	Mathematical functions and symbols	22
4.2	Platform architecture	24
4.2.1	Storage and record layer	25
4.2.2	Access layer	25
4.2.3	Interaction interface layer	26

4.3	Blockchain	26
4.3.1	Version selection	27
4.3.2	Block finalization	27
4.3.3	Notary Witness	27
4.3.4	Crosschain communication and Ethereum Classic	29
4.3.5	Block Generation Reward	30
4.3.6	Ethereum VM	31
4.3.7	Смарт-контракты Ethereum	31
4.3.8	Musereum rules for EVM interaction	31
4.4	Decentralized Storage	32
4.4.1	Overview	32
4.4.2	Naming System	33
4.5	Децентрализованные приложения	33
4.5.1	Governance dApp	33
4.5.2	Contract Registry	36
4.5.3	Musereum Name System (MNS)	36
4.5.4	Voting dApp	38
4.5.5	Musical Assets Registry	40
4.5.6	Decentralized Autonomous Labels	42
4.5.7	Soundchain	48

1 | Введение

«Musereum» – это мульти-блокчейн проект, созданный консорциумом блокчейн-профессионалов в сотрудничестве с экспертами в разработке программного обеспечения, юристами и профессиональными участниками музыкальной индустрии, способными применить сильные стороны указанных сфер и технологии для решения реальных бизнес-задач, а также преобразования мировой музыкальной индустрии в целом.

Проект представляет собой музыкальную платформу, сформированную на базе смарт-контрактов. Токены «Musereum» являются основополагающей единицей на платформе для построения экономических взаимоотношений между всеми участниками сети.

Мы понимаем, что изменить текущую ситуацию в индустрии, которая формировалась десятки лет, в один момент не получится, поэтому инновационные механизмы будут внедряться постепенно в коллаборации с заинтересованными сторонами процесса.

В рамках проекта «Musereum» участники смогут создавать токены, в которые будет “упакована” доля в совместном исключительном праве на музыкальное произведение. Эмиссия таких токенов будет со-

провождаться лицензированием того или иного объекта музыкального произведения и таймштампингом в публичной сети Ethereum или Ethereum Classic. Музыкальные токены будут свободно обмениваться на авторизованных биржах, позволяя музыкантам привлекать средства по модели краудинвестинга, профессиональным участникам рынка – использовать прозрачный инструмент для очистки прав на музыкальные произведения, слушателям – иметь легальную возможность напрямую поддерживать любимого автора, а инвесторам – получить новый инвестиционный инструмент.

Результатом реализации проекта станет коренное изменение отношений, складывающихся в музыкальной сфере, а именно:

- создание базы данных для управления правами интеллектуальной собственности;
- достижение справедливости при выплате роялти посредством децентрализации процессов;
- создание музыкальной платформы для начинающих талантов;
- упрощение процедуры лицензирования.

2 | Обзор индустрии

Уже к середине 2017 года общее потребление музыки выросло на 9,9% по сравнению с прошлым годом.

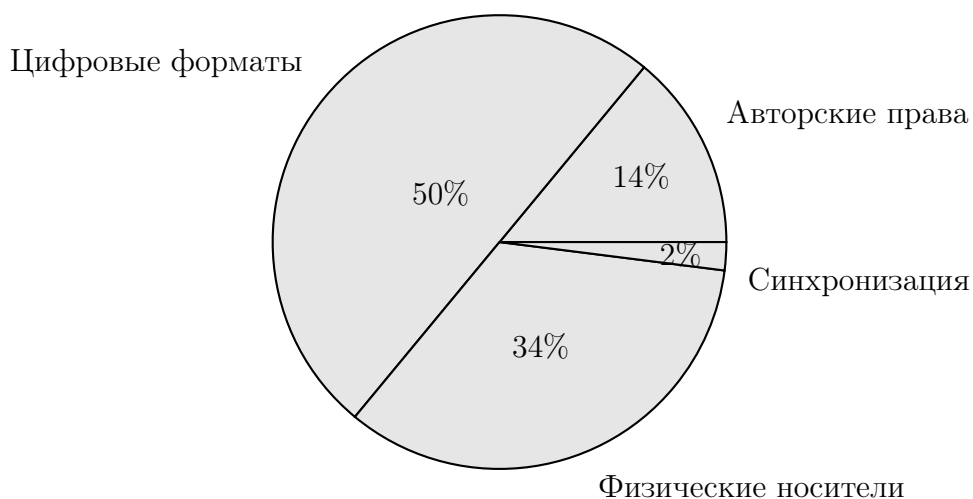
Аналитическая компания «Buzzanngelmusic» провела исследование музыкального рынка и пришла к следующим выводам: общий объем потребления музыкальных альбомов вырос на 9%, а музыкальных произведений - на 29,5%.

Таблица 2.1: Предзаготовленные типовые ограничения лицензий

	2016 год	2017 год	% роста
Общее потребление музыкальных произведений	1,167,384,931	1,512,049,118	29.5%
Продажи песен	410,920,611	313,305,154	-23.8%
Потоковое вещание	113,469,648,006	179,811,594,535	58.5%
Видео вещание	95,696,158,924	101,531,507,971	6.1%
Общее потребление альбомов	266,565,904	292,986,056	9.9%
Продажи альбомов	86,029,972	74,093,472	-13.9%

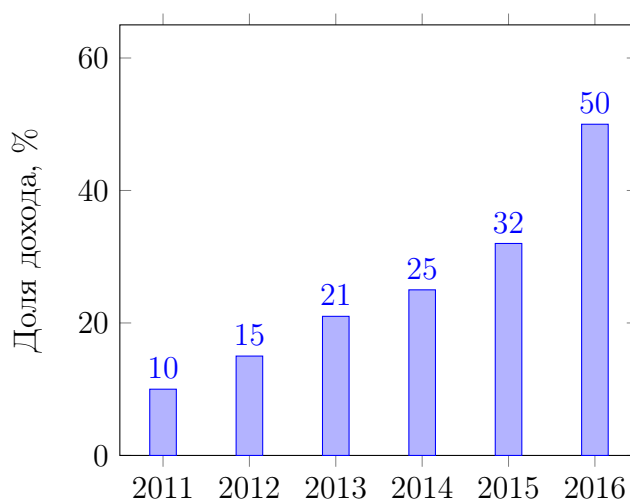
2.1 Доходы от распространения музыкального произведения

Рис. 2.1: Доходы от распространения музыкального произведения



Такая популярность обусловлена соответствующим ростом доходности. Стоит отметить, что наибольший доход приносят произведения, распространяемые на цифровых форматах, которые составляют ровно половину всей потребляемой музыки, в отличие от тех, например, которые распространяются посредством физических продаж - 34%, согласно данным

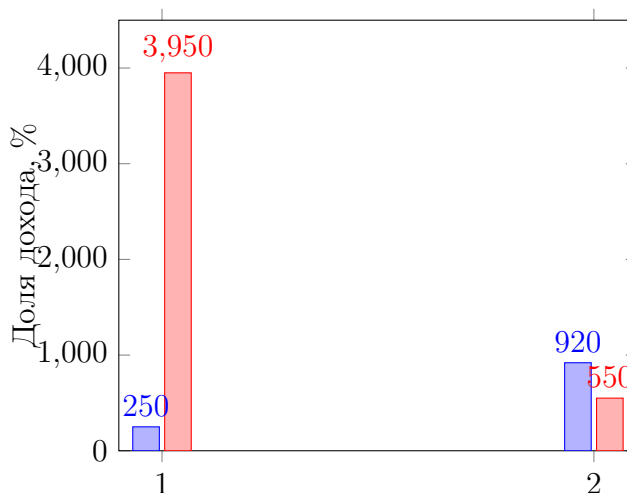
Рис. 2.2: Доходы от потокового вещания



В 2017 году аудиостриминг достиг рекордных отметок и вырос на 58,5%. На отдельных рынках рост доходности достигал более 300% (Южная Африка). По доходности аудиостриминг является лидером в своей сфере. К примеру, несмотря на то, что видео направление привлекает 900

млн пользователей, прибыль составила всего \$553 млн, в то время как, доход от потокового вещания равен 3,904 млн, а общее количество пользователей 212 млн.

Рис. 2.3: Аудио- и видео-стриминг: соотношение количества пользователей и прибыли



Несомненно, затраты на раскрутку одной композиции требует средств. В среднем инвестиции в новую запись составляют US\$498 000 - 2 000 000 (в частности, именно такие цифры подтверждают данные из США и Великобритании). В 2015 году было инвестировано US\$4.5 млрд в A&R и Marketing прибыли всей мировой музыкальной индустрии.

Таблица 2.2: Инвестиции в раскрутку новой записи

Инвестиции в раскрутку новой записи	
Привлечение клиентов	US \$50.000-350.000
Затраты на запись	US \$150.000-500.000
Видео	US \$25.000-300.000
Поддержка тура	US \$50.000-150.000
Маркетинг	US \$200.000-700.000
Итого	US \$475.000-2.000.000

На популярность музыкального произведения влияет и популярность артиста, его исполняющего, что, в наше время, напрямую связано с использованием социальных сетей. 6 из 10 наиболее понравившихся людей на Facebook, followed на Twitter, просматриваемых видео на Youtube – музыкальные исполнители.

Также необходимо учитывать международный характер потокового вещания. 230 цифровых музыкальных сервисов предлагают доступ к 40 млн треков на территории всего ЕС. При этом, около 50% всех популярных треков, загруженных и проигрываемых в Европе, не европейского производства.

Наибольший процент прибыли в 2016 года, а именно, 50%, мировому рынку музыкальной индустрии принесли именно музыкальные произведения в цифровом формате, общая сумма составила US\$15.7 млрд.

IFPI утверждает, что 50% доходов от записи поступают субъектам музыкальной индустрии именно благодаря наличию цифровых форматов.

К примеру, цифровые платформы выплачивают следующий процент:

Таблица 2.3: Выплаты платформ

Музыкальная платформа	Оплата за прослушивание	
	Отчисления для лейбла	% отчислений, попадающих артисту
Apple Music / iTunes	US \$0.0013	12%
Zune	US \$0.028	11%
Napster	US \$0.016	11-12%
Rhapsody	US \$0.013	11-12%
Spotify	US \$0.005	11-12%
Youtube	отчисление осуществляется в размере \$0,001 субъекту, разместившему произведение	

Но платформа и артист являются не единственными субъектами музыкальной индустрии.

Поскольку схема правоотношений оказывается крайне громоздкой, конечному пользователю, желающему получить лицензию на использование музыкального произведения, зачастую приходится обращаться к еще одному профессиональному игроку на рынке – к агентству по очистке прав, которое либо урегулирует отношения между всеми правообладателями, либо просто выкупает права на музыкальное произведение.

Таким образом, основными игроками музыкальной индустрии являются:

2.2 Структура музыкальной индустрии

Кажется очевидным, что центральной концепцией и итоговым продуктом, которым обусловлено само существование музыкальной индустрии, является музыка. Однако само это понятие является неоднородным, и именно благодаря этому вместо прямых отношений «Музыкант – слушатель», существовавших в каменном веке, мы имеем сложную, запутанную и не всегда эффективно работающую структуру.

Стоит сказать, что музыка совмещает в себе два объекта:

1. музыкальное произведение (инструкция по исполнению музыки),
2. фонограмма (исполненная и сохраненная на материальном носителе музыки).

В отношении каждого из этих объектов действует исключительное право, то есть, право определять порядок использования и получать вознаграждение за такое использование. Исключительным правом на музыкальное произведение обладают его авторы (композитор, автор текста), на фонограмму – исполнитель и изготовитель фонограммы. В самой простой ситуации, автор, исполнитель и изготовитель фонограммы оказываются одним человеком – однако, это редкость.

Таким образом, для использования музыки по общему правилу требуется получить разрешение авторов, исполнителей и изготовителей фонограммы, но музыканты редко самостоятельно управляют своими правами.

Исполнитель обычно обращается к звукозаписывающей компании – лейблу – и получает от него аванс в обмен на право лейбла удерживать часть роялти от дальнейшего использования музыки; зачастую при этом лейблу передаются и исключительные права на исполнение.

Лейблы можно разделить на два основных вида: мейджоры (the majors) и инди-лейблы (the independents). Первые являются крупными звукозаписывающими компаниями, вторые представляют микро-, малый и средний бизнес индустрии. Несмотря на это, именно на долю последних приходится 80% всех новых релизов, самый крупный из них занимает всего лишь 1,5% рынка. На мировом рынке индустрии инди-лейблы составляют 37,6% и оцениваются в \$5.6 млрд. В среднем они работают с 40 артистами. Обоим видам лейблов потоковое вещание приносит наибольшую долю прибыли.

Право изготовителя фонограммы также обычно принадлежит лейблу, хотя может удерживаться и исполнителем. Авторы, в свою очередь, передают свои права в управление музыкальному издательству – паблишеру – занимающемуся защитой авторских прав в обмен на часть дохода от использования произведения.

Распространением и доведением музыки до конечного пользователя занимаются дистрибьюторы (распространяющие музыку на физических носителях) и агрегаторы (распространяющие музыку в цифровом формате).

Поскольку даже крупные правообладатели не в состоянии отслеживать любое использование принадлежащей им музыки, в большинстве государств создаются организации по коллективному управлению правами (ОКУП). ОКУПы отслеживают случаи публичного использования произведений и взимают плату за такое использование, которая потом тем или иным способом распределяется между правообладателями; при этом обычно для каждого типа исключительных прав действует отдель-

Поскольку схема правоотношений оказывается крайне громоздкой, конечному пользователю, желающему получить лицензию на использование музыкального произведения, зачастую приходится обращаться к еще одному профессиональному игроку на рынке – к агентству по очистке прав, которое либо урегулирует отношения между всеми правообладателями, либо просто выкупает права на музыкальное произведение.

- Композиторы;
- Авторы текстов;

Однако при этом реальные отношения в отношении музыки оказываются еще сложнее. При создании современной музыки широко используется семплирование; при этом в информационно перенасыщенном пространстве невозможным становится создание абсолютно оригинального произведения, не содержащего заимствования. За каждым использованием в произведении семплом стоит такая же сложная структура прав, что делает получение лицензии на музыку

новый ОКУП. В некоторых странах такие организации существуют только для определенных типов прав. Кроме этого, в отдельных странах (Россия, скандинавские государства) ОКУПы управляют правами на бездоговорной основе – то есть имеют права взыскивать роялти в пользу правообладателей, не имеющих с ОКУПом никаких отношений.

Таким образом, примерная структура создания ценности в музыкальной индустрии выглядит следующим образом:

- Музыканты-исполнители;
- Лейблы;
- Пабlishеры;
- Агентства по очистке прав;
- Дистрибьюторы и агрегаторы;
- Организации по коллективному управлению правами.

Каждое использование музыки создает поток роялти, распределяющиеся между всеми указанными игроками:

фрактально сложным предприятием.

Эта сложность иллюстрируется заявлением RIAA, один из членов которой для выпуска альбома был вынужден получить 1481 лицензию от 51 автора с 89 долями, одна из которых представляла собой всего 1,5 процента от общего объема прав, но была разделена между двумя пабlishерами.

Учитывая международный характер цифрового распространения музыки, о кото-

ром упоминалось выше, процесс еще больше усложняется, так как возникает необходимость учета особенностей правового регулирования отдельных юрисдикций. Так, в

США, правообладатели не имеют права на получение вознаграждения при вещании их музыки по радио.

2.3 Основные проблемы музыкальной индустрии

Чрезмерная усложненность структуры отношений в музыкальной индустрии ведет к возникновению общепризнанных проблем, которые в той или иной степени затрагивают всех участников отношений.

2.3.1 Отсутствие доступной и достоверной информации о принадлежности прав

Как указывает RIAA, «сложно идентифицировать и отслеживать принадлежность права на музыкальное произведение в связи с изменениями при переходе произведений и каталогов из рук в руки». Помимо необходимости отслеживания истории смены правообладателей, ситуация усложняется многослойностью музыки как объекта интеллектуальных прав – существованием отдельных прав на музыкальное произведение и аудиозапись, а также традиционной для нынешнего состояния индустрии множественностью правообладателей для каждого из таких слоев. Кроме этого, проблема заключается в отсутствии универсально принятых в индустрии идентификаторов данных.

Эта ситуация ведет к отсутствию определенности для потенциальных лицензиатов – им неясно, с кем именно надо договариваться, чтобы получить возможность использовать музыку на законных основаниях. Музыкальный рынок частично решает эту проблема появлением нового типа игроков – агентств по очистке прав – однако это решение является субоптимальным, поскольку увеличивает количество работы, вынуждая специалистов затрачивать суще-

ственные ресурсы на работу со сложно отслеживаемой историей транзакций, вместо того, чтобы отслеживать и записывать эту историю в реальном времени. Результатом является существенное повышение стоимости музыкального произведения для конечного пользователя, который вынужден из своего кармана оплачивать борьбу с такой неопределенностью.

Более того, некоторые композиторы и музыканты в связи с особо сложно отслеживаемой историей могут вообще оказаться лишены права на доведение своих произведений до конечного пользователя, как в силу отказа со стороны агентства по очистке прав, так и в результате дедлока между множественными правообладателями.

В настоящее время нет единой базы данных для хранения информации об авторских правах на музыкальное произведение. В книге “Биткойн для рок-звезд” Д.А. Уоллак объясняет это так: “Корень проблемы – отсутствие одного набора данных со сведениями об авторах и правах. Сегодня эта информация разпылена между многими организациями, каждая из которых считает ее своей собственностью. Это вполне понятно, если учесть, что они инвестируют в свои наборы данных серьезные деньги”.

В 2011 году была предпринята попытка создать такую базу данных. При поддержке музыкальных издателей был запущен проект по составлению “Глобальной БД репертуара” (GRD). Предполагалось, что она станет центром регистрации всех звукозаписей и связанных с ними метаданных. Увы, в июле 2014 года проект был закрыт из-за

претензий со стороны обществ по сбору авторских отчислений.

2.3.2 Непрозрачность транзакций в индустрии

С первой проблемой неразрывно связана вторая. Даже если на определенный момент времени принадлежность прав на произведение или аудиозапись оказывается достоверно зафиксированным, отсутствуют универсальные решения, позволяющие убедиться в актуальности такой информации. Более того, в случае возникновения спора, связанного с потенциальным инфринджментом в прошлом, зачастую невозможно доказать, кто именно был надлежащим лицензиаром на конкретную дату.

Отсутствие прозрачности бьет не только по лицензиатам, увеличивая для них неопределенность в отношениях с контрагентом. Невозможность отследить транзакции приводит и к невозможности конечных производителей музыкального контента эффективно защищать свои интересы при определении причитающегося им вознаграждения. Так, например, в США паблишеры не имеют права на аудит передаваемой лицензиатом информации, и вынуждены принимать расчет полученных роялти на веру.

Даже при условии честного предоставления информации конечным лицензиатом, цепочка передачи ценности оказывается слишком сложной для эффективного контроля со стороны конечного звена – автора или музыканта. Непрозрачные транзакции стимулируют посредников оставить себе как можно больше – так, например, Перри Резник, аудитор в музыкальной индустрии, сообща-

ет о схеме, при которой лейбл в обмен на лицензию получает долю в бизнесе лицензиата, таким образом превращая движение денежных средств в

2.3.3 Отсутствие эффективных средств контроля за использованием музыки

Еще одной проблемой являются сложности по определению путей и способов использования музыкальных произведений и аудиозаписей. Помимо прямого коммерческого интереса в получении роялти, авторам и музыкантам принадлежат права на контроль за путем и способом использования музыки, которые порой несут не меньшую ценность. Так, автор симфонического произведения, наполненного эмоциональным подтекстом, может быть принципиально против использования его произведения в рекламе кошачьего туалета; иногда и просто сосуществование с другими произведениями в рамках одного медиума может быть нежелательным.

Вместе с тем, с увеличением количества правообладателей, определение порядка использования становится все сложнее. Помимо чисто организационных сложностей, ситуация усугубляется тем, что за рамками общего стремления к максимизации роялти цели авторов и паблишеров, музыкантов и лейблов могут существенно отличаться. Зачастую лейблу полностью передается право на определение порядка использования произведения; даже если этого и не происходит, голос первоначального автора заглушается коммерческими мотивами посредников, принимающих решения вместе с ним

или за него.

Отдельную проблему составляет режим принудительного лицензирования музыки. Так, в США неинтерактивные стриминговые музыкальные сервисы (к которым, в том числе, относится Pandora), могут использовать музыку без согласия правообладателей. Интерактивные сервисы должны договариваться с правообладателями о получении лицензии на использование аудиозаписи, однако в отношении авторов музыкального произведения требуется только направить уведомление о намерении использовать произведение (NOI). Эта ситуация в принципе лишает авторов контроля за стримингом своих музыкальных произведений.

2.3.4 Черные ящики

Все три базовых проблемы, указанные выше, в совокупности приводят к очень серьезному перекосу рынка и появлению так называемых «черных ящиков», под которыми подразумеваются огромные суммы нераспределенных роялти. Возникновение «черных ящиков» обусловлено следующим:

причитающиеся правообладателю роялти не смогут быть выплачены, пока этот правообладатель не будет установлен. В связи с отсутствием базы данных о принадлежности прав на музыкальные произведения, сумма роялти «повисает в воздухе».

Более того, даже если правообладатель известен, каждое звено длинной цепочки заинтересовано в том, чтобы максимально задержать выплату. К примеру, в США известны случаи массового злоупотребления при направлении NOI. Автор может так и не узнать об использовании его произведения.

Слабое развитие международных коопераций паблিশеров и организаций по коллективному управлению музыкальными правами приводит к тому, что при транснациональном использовании музыкальных произведений и звукозаписей, собранные роялти могут так и остаться в стране использования музыки, не дойдя до правообладателя, domicilia другого государства.

2.3.5 Пиратство

Сложность и неповоротливость описанной схемы приводит к тому, что, порой, пользователю дешевле нарушить исключительное право, нежели разбираться в сложностях установления договоренностей со всеми правообладателями. Стоит отметить, что возможно и «этическое пиратство» как протеста против искусственно усложненной и эксплуатирующей музыкантов системы. По данным американского исследования профессора Питера Ди Колы, ежегодно BitTorrent используют более 1 млрд чел. Данная цифра позволяет лишь приблизительно оценить ущерб, причиняемый музыкальным пиратством. Объединенный центр Европейского союза указывает, что в 2011 году в мировом масштабе ущерб от использования пиратского контента составил 8%, что равно примерно \$5,2 миллиардов. Однако данная организация считает, что стремиться к искоренению интернет-пиратства не нужно.

2.3.6 Сложности монетизации музыкального контента

Непрозрачность выплат, сложность структуры и огромное количество посредников

приводит к тому, что период от использования музыки до получения роялти может превышать год. При этом, как указывалось

выше, риск невыплаты роялти также имеет место быть.

2.4 Решение вышеописанных проблем проектом Musereum

Проект Musereum направлен на устранение обозначенных проблем. Мы предлагаем внедрение технического решения, позволяющего в реальном времени отслеживать и контролировать принадлежность прав на музыкальное произведение, а также автоматизировать выплату роялти с помощью токенизации музыкальных произведений и лицензировании этих токенов смарт-контрактами.

Таким образом, проект «Musereum» является мощным правовым механизмом управления различными типами прав в музыкальной индустрии.

Действующие музыкальные платформы, использующие Blockchain-технологии, также не имеют преимуществ, содержащихся в проекте «Musereum».

Рис. 2.4: Структура создания ценности

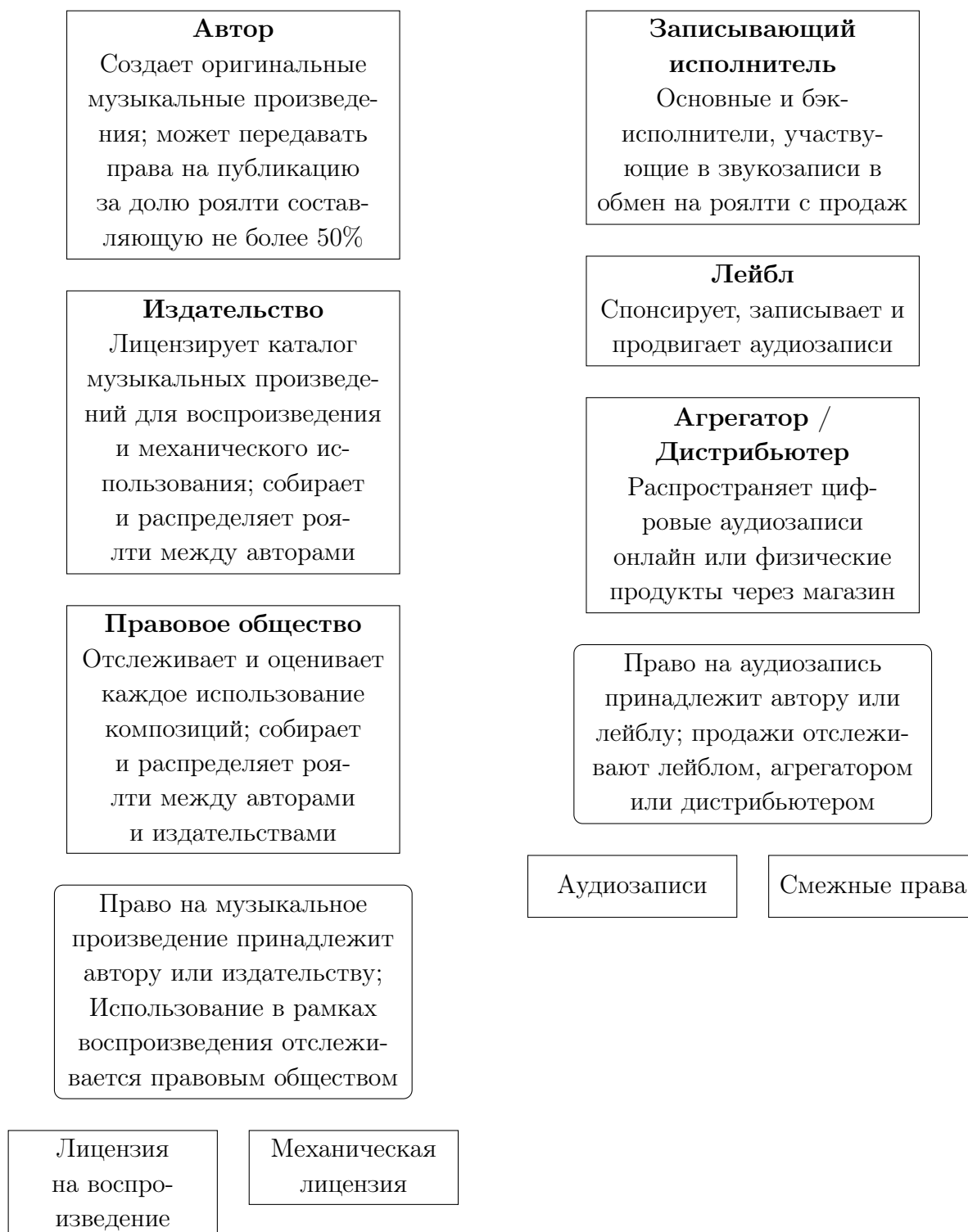


Рис. 2.5: Аудио- и видео-стриминг: соотношение количества пользователей и прибыли

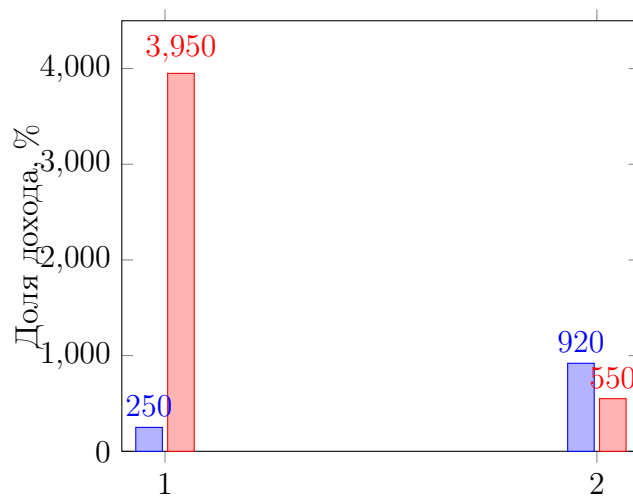


Рис. 2.6: Аудио- и видео-стриминг: соотношение количества пользователей и прибыли

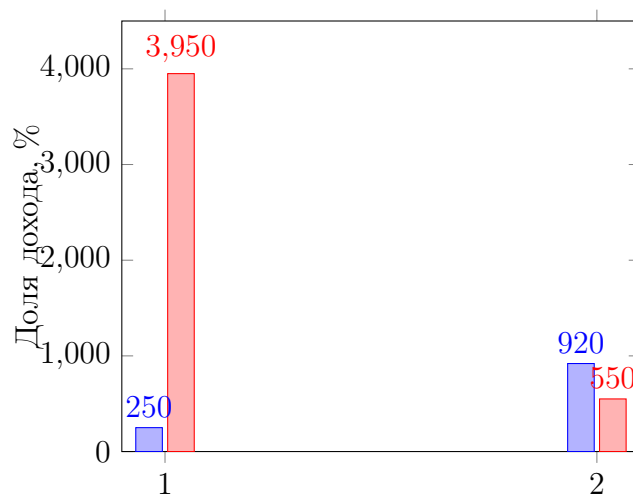


Таблица 2.4: Инвестиции в раскрутку новой записи

Проблема	Решение
Отсутствие доступной и достоверной информации о принадлежности прав	Создание проверенной базы данных, исчерпывающим образом описывающей распределение прав путем записи о соответствующих полномочиях в блокчейне.
Непрозрачность транзакций в индустрии	Любая запись в реестре позволяет отследить как выданные лицензии, так и произведенные выплаты, которые автоматически рассчитывают размер и порядок выплаты причитающихся роялти.
Отсутствие эффективных средств контроля за исполнением музыки	Проект вводит автоматизированный и унифицированный механизм голосования всех правообладателей по вопросам выдачи лицензий, передачи долей в праве и прочим вопросам через создание «Децентрализованного Автономного Лейбла»
“Черные ящики”	За счет прозрачности транзакций, непосредственному распределению роялти конечным правообладателям и доступности информации о принадлежности прав отпадут условия для формирования черных ящиков.
Пиратство	Использование смарт-контрактов для лицензирования музыки само по себе не может решить проблему пиратства, однако снизит транзакционные издержки легального доступа к музыке и исключит этические барьеры для законного получения лицензий. Кроме этого, проект Soundchain DAPP, который будет развиваться на базе проекта Musereum, позволит конечным слушателям получать бесплатный доступ к музыке, что само по себе существенно снизит экономическую мотивацию для нарушения исключительных прав.
Сложности монетизации музыкального контента	<p>Автоматизация выдачи лицензий и снижение издержек на очистку прав приведет к росту доходов правообладателей от коммерческого использования их музыки, облегчит процедуру лицензирования и снимет существенную часть административных барьеров.</p> <p>Техническая возможность учета принадлежности и передачи долей в исключительном праве в реальном времени открывает для правообладателя возможность проводить краудинвестиционные кампании, вместо того, чтобы на невыгодных условиях в обмен на аванс отдавать существенную часть права лейблу или паблишеру. Это позволит фанатам напрямую материально поддерживать любимых авторов и исполнителей через процедуру ICO музыкальных треков. Потенциальные лицензиаты получают дополнительный инструмент для оценки возможной коммерческой привлекательности</p>

Таблица 2.5: Инвестиции в раскрутку новой записи

Возможности	Soundchain	Ujo Music	Musicoин	Token FM	Jaak	PeerTracks	Opus
Свой блокчейн	Да	Да	Да	Нет	Нет	Да	Нет
Токенизация прав на треки	Да	Нет	Нет	Нет	Нет	Нет	Нет
Бланкетные лицензии на смарт-контрактах	Да	Нет	Нет	Нет	Нет	Нет	Нет
Коллективное управление	Да	Нет	Нет	Нет	Нет	Нет	Нет
Инвестирование в музыку	Да	Нет	Нет	Нет	Нет	Нет	Нет
Децентрализованное хранение файлов	Да	Да	Да	–	–	Да	Да
Децентрализованное хранение метаданных	Да	Да	Да	–	–	Да	Да
Шифрование контента	Да	Нет	Нет	–	–	–	Да
Вторичный рынок прав на музыку	Да	Нет	Нет	Нет	Нет	Нет	Нет
Фандрайзинг для музыкантов	Да	Нет	Нет	Нет	Нет	Нет	Нет

3 | Platform Overview

Musereum Platform is a system of digital products to solve core industry problem which described above.

Core platform products is:

- **Decentralized Ledger:** Our decentralized ledger based on Ethereum source code, but with different consensus algorithm to avoid incurring transaction fees compare to Ethereum main-net.
- **Decentralized Virtual Machine** with cheap computation power. Move forward from PoW (see [Blockchain](#) to detail) allow to decrease cost of computation power and utilize decentralized network to solve more complex tasks such as *searching, indexing, loop* over data.
- **Decentralized Content Network:** The current industry solutions delivers their musical content from centralized content networks. In order to deliver on the promise as laid out in the business objectives, one deliverable is that of a decentralized system wherein many ecosystem participants could run nodes of the system and use consensus algorithms to synchronize with one another.
- **Cryptocurrency:** The introduction of a cryptocurrency would enable a common exchange of value across all ecosystem participants globally. It would also serve to equalize the playing field and set a benchmark for services provided.
- **Transparent Rights Registry:**
- **Decentralized Exchange** without moment of authority to select what we trade and when. Any registered asset could to listing associated token as a tradable active.
- **Decentralized Marketplace** for the monetization of creativity through the sale of licenses.

3.1 Tools

All of tools could be grouped by three categories:

1. **Business and economy.** Core features provider here is *Musereum Wallet*. Musereum Wallet is not just a wallet to hold and spend crypto-currency, it is default door to whole economy system of

Musereum.

2. **Entertainment and listening.** Implements as a services around Musereum such as Soundchain to listen music for free.
3. **Infrastructure.** Service and maintain level tools to monitoring, developing

and auditing network. We are providing bunch of tools and services: blockchain

explorer, deploy utilities, public http api and also.

3.2 Roles

In a nature of platform users could identify yourself with desired roles as well as delegate roles to others.

Primary platform select next core roles:

- **User** is the person who create an account on the platform and got permissions to hold and transfer ETM and interact with a musical catalog.
- **Person** is the user who provide proof-of-personality to receive permissions to invest and trade with a musical assets.
- **Artist** is a person with at least one registered musical asset on a platform. Should be at least person to register asset.
- **Notary** is the person chosen for the production of blocks
- **Affiliated specialist** is the person chosen to do some important work for the platform.
- **Keeper** is the user provided storage to store musical content.

4 | Technical Description of Platform

4.1 Overview

Musereum protocol tasks can be divided into the following components:

- Providing unified system state without central computing for reliable operation;
- Decentralized big data storage that is censor-proof due to inability to block a unified (centralized) data source;
- Providing proof of system integrity for change history auditing;
- Interface for making changes in the future state of the system.

Application of the "Blockchain concept" using Ethereum Virtual Machine (EVM) provides monitoring and changing single state of Musereum system without the need to use centralized computing (see [Blockchain](#) for detail). IPFS technology is applied to store big data volumes (audio tracks, metadata, text and graphical description) (*see [Decentralized Storage](#) for more detail*).

4.1.1 Naming convention

Here and elsewhere all entities with defined content or interface will get a short mathematical name with a common rule: *capital latin letter except for N, B, Z*, e. g:

$$Y \text{ is Notary} \quad (4.1)$$

All globally defined contracts will be assigned to a lowercase Greek symbol, e. g. Notary Registry Contract – ν and a special record associated with certain function of the contract:

$$\nu \text{ is Notary Registry Contract} \quad (4.2)$$

$$\nu_{vote} \text{ is vote contract method} \quad (4.3)$$

Sets of all known entities are defined as accepted entity name in a blackboard record except for $\mathbb{N}, \mathbb{B}, \mathbb{Z}$, e. g.: list of notaries:

$$\mathbb{Y} \text{ is set of all known Notaries} \quad (4.4)$$

\mathbb{N} – set of all integers within the range $(0; 2^{256} - 1)$. \mathbb{N}_1 – subset \mathbb{N} without 0.

\mathbb{Z} – set of all integers within the range $(-2^{255}; +2^{255} - 1)$.

\mathbb{B} – is a set of all bytes $(0; 255)$.

\mathbb{X}^i – is a set of dimension i where every set member belongs to \mathbb{X} , e.g.:

$$\mathbb{B}^{32} \text{ is a set of } \mathbb{B} \text{ with size } 32 \quad (4.5)$$

Obtaining the set member can be described in two ways: with a subindex or in square brackets:

$$\begin{aligned} \mathbb{Y}_2 &= \mathbb{Y}[2] \\ \mathbb{Y}_n &= \mathbb{Y}[n] \end{aligned} \quad (4.6)$$

External data is represented in the formulae as members of set Γ with readable names (or abbreviations if mentioned), e. g.: Global time in UNIX format:

$$\Gamma_{time} \text{ is world time variable} \quad (4.7)$$

All **temporal** variables in formulae are written as latin letters, e. g.:

Let a is a second notary in list of notaries

$$a = \mathbb{Y}[2] \quad (4.8)$$

4.1.2 Mathematical functions and symbols

All formulae apply $|\mathbb{Y}|$ as an operator of number of \mathbb{Y} -set elements.

The following conventions and functions are applied to reduce and/or make a mathematical notation readable:

Search index in set – \mathcal{I} :

Let a is a set of values:

$$a = 100, 200, 300$$

$$f(v, x) = \{i \mid i \in \mathbb{N}, i > |v|, v_i = x\} \quad (4.9)$$

$$\mathcal{I}(v, x) = \begin{cases} \min(f(v, x)), & \text{if } |f(v, x)| > 0 \\ -1, & \text{otherwise} \end{cases} \quad (4.10)$$

$$(4.11)$$

$$r1 = \mathcal{I}(a, 200) = 1$$

$$r2 = \mathcal{I}(a, 400) = -1 \quad (4.12)$$

Determining named tuple – \mathcal{O} :

Let a is a set of names:

$$a = \{name, age, height\}$$

and let b is a set of indexes:

$$b = \{0, 1, 2\} = \{x \mid x \in \mathbb{N}, x < |a|\} \quad (4.13)$$

finally r is a target named tuple, where

$$r = \mathcal{O}(a) = \mathcal{O}(name, age, height) \quad (4.14)$$

$$r = \{x_i \mid i \in a, x_i = 0\} \quad (4.15)$$

Determining event tuple – \mathcal{E} :

Special defined tuple containing target data and general data for every event: `event`, `txHash`, `txIndex`, `blockHash`, `blockHeight`, `address`.

Let a is a set of default event fields:

$$a = \{event, txHash, txIndex, blockHash, blockHeight, address\}$$

Let b is a set of extra event fields:

$$b = \{asset, index\}$$

finally e is a target event tuple, where

$$r = \mathcal{E}(b) \quad (4.16)$$

$$r = \mathcal{O}(\{asset, index\} \cup a) \quad (4.17)$$

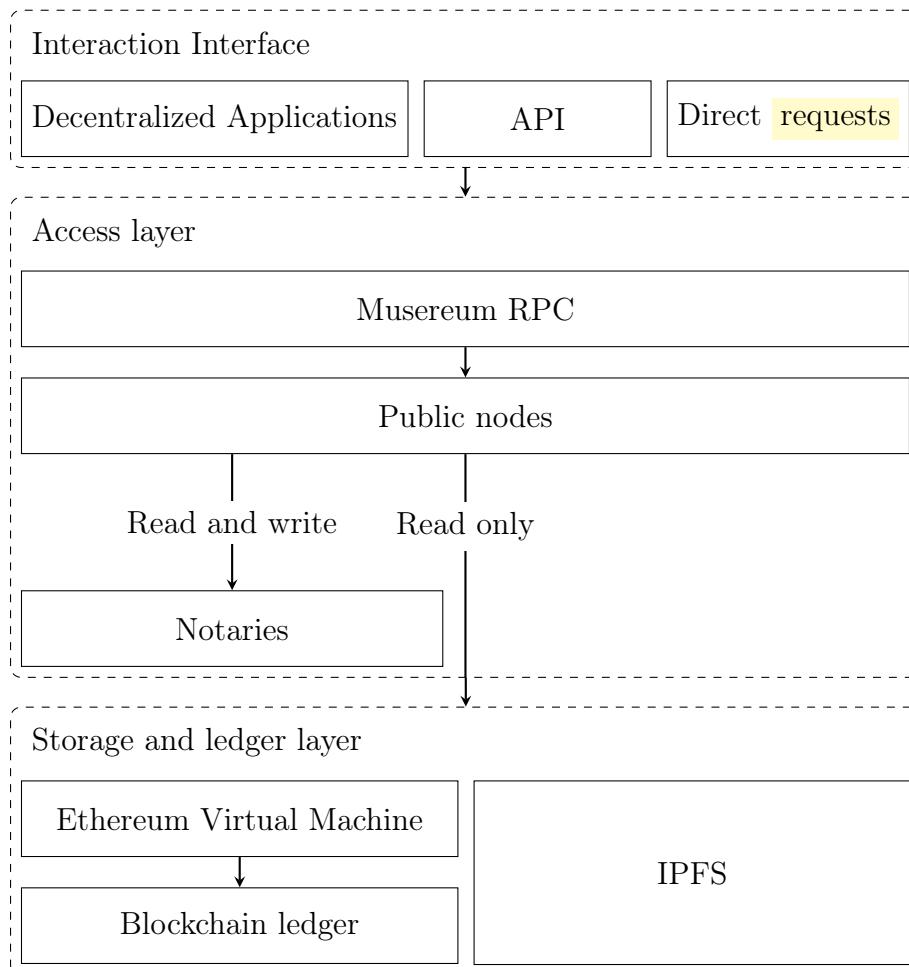
4.2 Platform architecture

The primary aim of the platform is to provide users with capabilities to:

1. Read and interpret data
2. Commit changes
3. Search data
4. Audit and receive proof of data integrity

The platform is divided into the following logic layers in order to provide above-mentioned capabilities:

- Storage and ledger
 - Blockchain
 - Decentralized storage
- Read and write layer
 - Public nodes
 - Notary nodes
- Interaction interfaces
 - Decentralized application
 - Application Programming Interface



4.2.1 Storage and record layer

Storage and record layer is the basic layer of the platform. The layer provides capability to keep a decentralized state ledger (blockchain) and to store big amounts of data.

Every layer element: blockchain, storage and virtual machine are distributed among the network (nodes) and everyone can create a personal local copy.

4.2.2 Access layer

Access layer application: providing data from the basic layer on request and receiving change requests.

Creating a node for an additional access point is not limited and available for anyone. This feature is a benefit that distinguishes Musereum platform from other centralized music platforms:

- It provides 100% network up-time
- It is Censor-proof
- Stakeholders can create a personal local copy of the network without necessity to trust third parties

4.2.3 Interaction interface layer

Interaction interface layer is the top layer of the platform. The layer receives user input and generates requests applicable with protocol. Requests are divided into two types: read and write.

Reading data from the blockchain and the decentralized storage is not limited. The only requirement is to form a request according to API agreement.

In order to write (changing state, saving data),

a user must provide valid request signature based on his or her unique private key as well as follow other writing requirements (rights, funds for protocol operation etc.).

It is not required for an organization to have its own access point (synchronize a node with the network) in order to make a request; a serverless web-application can be used as a client (see [Decentralized applications for detail](#)).

4.3 Blockchain

Musereum protocol is an open public blockchain that requires permission to generate blocks and is based on Ethereum protocol. Consensus for a single network state in a decentralized block generator structure is achieved by Proof-of-Authority (hereafter PoA) algorithm. Achieving consensus through PoA requires:

1. A common list of notaries (\mathbb{Y}) permitted to generate blocks;
2. Notary Registry Contract (ν).

The list of notaries can be changed (excluding / appointing notaries) by Notary Registry Contract based on results of active notary voting (majority).

Initially, 12 notaries are listed.

Notary **duties**:

1. Validate transactions received from system participants,

2. Write changed state to a new block ,
3. Sign block chain with a personal crypto-signature.

In order to maintain consensus, an external constant Γ_{step} is introduced, it defines the number of seconds in one time step or time between blocks. Musereum defines constant Γ_{step} as 5 or 1 block every five seconds.

$$\Gamma_{step} = 5 \quad (4.18)$$

According to PoA consensus algorithm, a notary has the right to generate one block (K) in a **timestamps** Γ_{time} . If a is a number of notaries:

$$a = |\mathbb{Y}| \quad (4.19)$$

Notary index i is selected from set \mathbb{Y} to create a block on time stamp b according to the formula:

$$b = \frac{\Gamma_{time}}{\Gamma_{step}}, \quad b \in \mathbb{N} \quad (4.20)$$

$$i = b \bmod a \quad (4.21)$$

4.3.1 Version selection

In case if notaries fail to achieve global (single) system state and a fork or creating two or more blockchains takes place (\mathbb{K}), the network defines the score ($\beta_{score}(\mathbb{K}_c)$) of every chain based on the number of notaries participating in **chain creation**:

$$h = |\mathbb{K}_c| \quad h \text{ is length of blockchain} \quad (4.22)$$

$$\beta_{score}(\mathbb{K}_c) = \Gamma_{u128max} * h - m \quad (4.23)$$

The network always selects the chain with the highest score according to β_{score} .

4.3.2 Block finalization

According to consensus, the block in the chain after which more than 50% of notaries have generated 2 or more blocks is considered irreversible.

Thus, the minimal time for complete block validation (C) is:

$$C = 2 * \Gamma_{step} * |\mathbb{Y}| \quad (4.24)$$

Or 120 seconds with default protocol settings: 12 notaries with 1 block every five seconds

4.3.3 Notary Witness

In order to do any activity that requires changing system state, user must form a transaction (T) signed with a unique cryptographic signature (T_{sign}).

The crypto-signature is a warrant for the network that proves owner's will to create a transaction using creator's account.

The notary in turn to generate a block checks the validity of a transaction and executes the related code:

1. Sender has the right to create a transaction for the given address

– $\rho_{allowTx}(T_{from}, T_{to})$ (see [EVM Interaction Rules](#) for detail)

2. Transaction meets formal criteria – $\rho_{initialValid}(T)$
3. Checking account owner's signature – $\rho_{checkSign}(T)$
4. Checking sufficient amount of tokens on ETM account to pay in advance for computer operation – $\rho_{upFront}(T)$
5. Execution of related EVM code did not throw an exception $\rho_{exception}(\rho_{execute}(T))$

$$\rho_{success}(T) = \rho_{allowTx}(T_{from}, T_{to}) > 0 \wedge \quad (4.25)$$

$$\rho_{initialValid}(T) > 0 \wedge \quad (4.26)$$

$$\rho_{checkSign}(T) > 0 \wedge \quad (4.27)$$

$$\rho_{upFront}(T) > 0 \wedge \quad (4.28)$$

$$\rho_{exception}(\rho_{execute}(T)) = 0 \quad (4.29)$$

Having validated the transaction, notary packs the transaction in blocks and informs the network about the block and change of the network state (\mathbb{W}_h).

$$\mathbb{W}_h = \rho_{state}(\mathbb{W}_{h-1}, \mathbb{K}_h) \quad (4.30)$$

Other notaries receive the block, check it ($\rho_{success}(T)$) and take decision to accept it

to the chain. By creating a block on \mathbb{K}_h the notary validates the block and all included transactions. Thus, the transaction may have from 1 to $|\mathbb{Y}|$ validating signatures. Minimal time for receiving $|\mathbb{Y}|$ signatures:

$$a = \Gamma_{step} * |\mathbb{Y}| \quad (4.31)$$

4.3.4 Crosschain communication and Ethereum Classic

В целях повышения безопасности и отказоустойчивости сети, Musereum периодически дублирует ключевую информацию о состоянии сети и событиях во внешний неконтролируемый нотариусами блокчейн.

В качестве блокчейна для хранения ключевой информации выбрана сеть Ethereum Classic, по следующим причинам:

- Общая капитализация сети – \$1,811,306,656
- Стоимость атаки 51% на сеть состав-

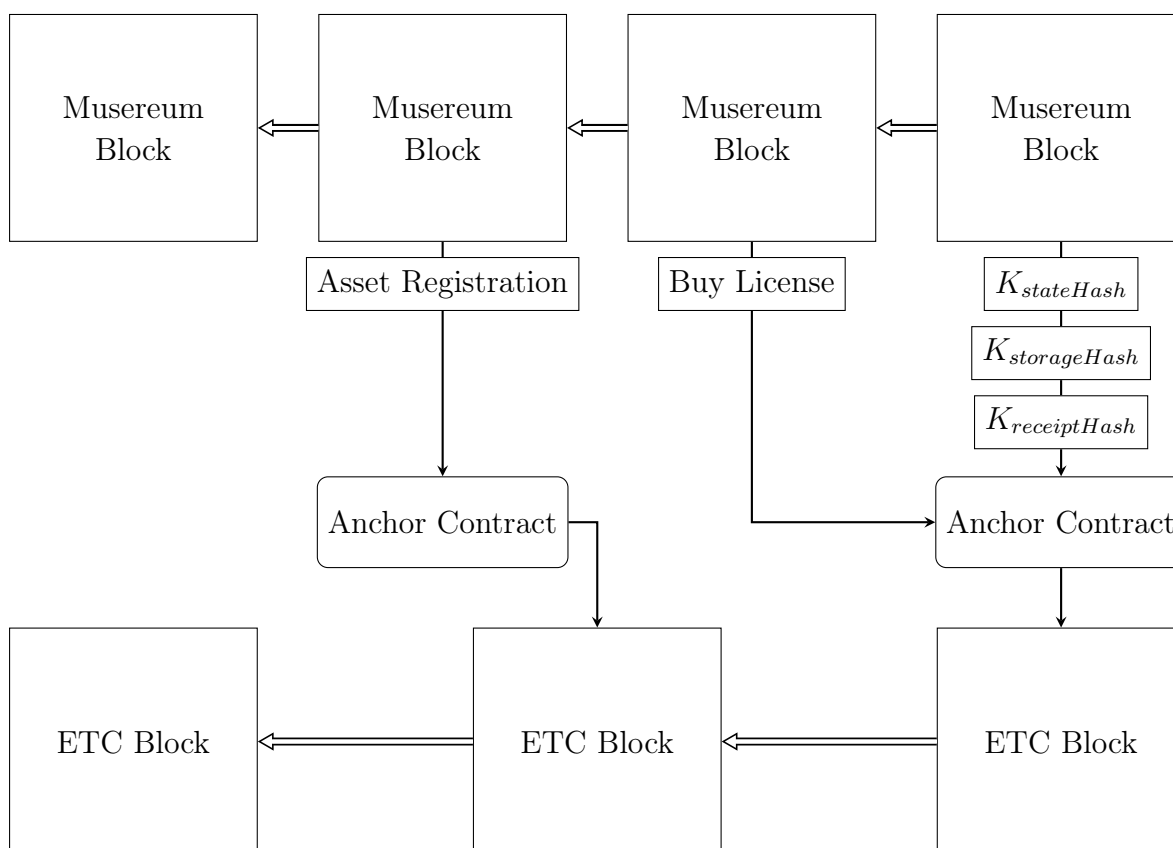
ляет более \$40,000,000

- Permission-less консенсус – PoW.

Musereum публикует в Ethereum Classic:

1. хеши каждого n блока для обеспечения доказательства исторических данных;
2. факты ключевых сделок: регистрация музыкальных активов, покупка лицензий.

Рис. 4.1: Crosschain communication



4.3.5 Block Generation Reward

Block reward (R) consists of two parts: new emission and activity fees.

Musereum platform implies emission of new ETM tokens by creation of a block. Emission is limited to 3 ETM per block.

Amount of activity fee is calculated independently for each transaction. The sender indicates in transaction difficulty unit price ($T_{gasPrice}$) that he or she is ready to spend to compensate network operation.

Final fee is calculated according to formula:

$$x = T_{gasPrice} * \rho_{difficulty}(\rho_{execute}(T)) \quad (4.32)$$

Total block reward:

$$\text{Let } a \text{ is a list of transactions of block } \mathbb{K}_h \quad (4.33)$$

$$a = \mathbb{K}[h]_{transactions} \quad (4.34)$$

$$\mathbb{R}_h = 3 + \sum_{i=0}^{|a|} a_i \quad (4.35)$$

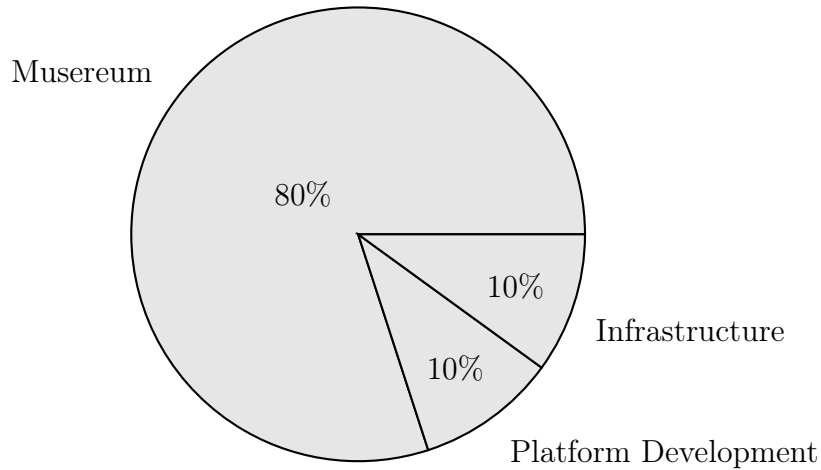
Reward distribution

Apart from Ethereum and Bitcoin protocols, Musereum network has no miners – the primary value is created by artists who select Musereum platform for publication and distribution of music.

Musereum highly appreciates artists' contribution to platform integrity and it sends the most part of block reward to fund artists.

General distribution pattern:

Рис. 4.2: Block reward distribution



- 80% – Musereum:

- 90% – SoundChain Trust (*see more in Pay-per-Play*)
- 10% – bounty programs: Fans Bounty, Clipmaker Bounty and etc.
- 10% – Platform Development Foundation
- 10% – Network Infrastructure
 - 25% – Notaries / Validators
 - 25% – Keepers
 - 25% – Affiliated specialists (registrators and smart-contract auditors)
 - 25% – Oracles

4.3.6 Ethereum VM

Ethereum is an open-source, public, blockchain-based distributed computing platform featuring smart contract (scripting) functionality. It provides a decentralized Turing-complete virtual machine, the Ethereum Virtual Machine (EVM), which can execute scripts. "Gas" an internal transaction pricing mechanism, is used to mitigate spam and allocate resources on the network.

4.3.7 Ethereum Smart-contracts

Smart contracts are deterministic exchange mechanisms controlled by digital means that can carry out the direct transaction of value between untrusted agents. They can be used to facilitate, verify, and enforce the negotiation or performance of economically-laden procedural instructions and potentially circumvent censorship, collusion, and counter-party risk. In Ethereum, smart contracts are treated as autonomous scripts or stateful decentralized applications that are stored in the Ethereum blockchain for later execution by the EVM. Instructions embedded in Ethereum contracts are paid for in ether (or more technically "gas") and can be implemented in a variety of Turing complete scripting languages.

4.3.8 Musereum rules for EVM interaction

In order to improve security of platform participants, interaction with EVM smart contracts in Musereum differs from that in parental Ethereum network.

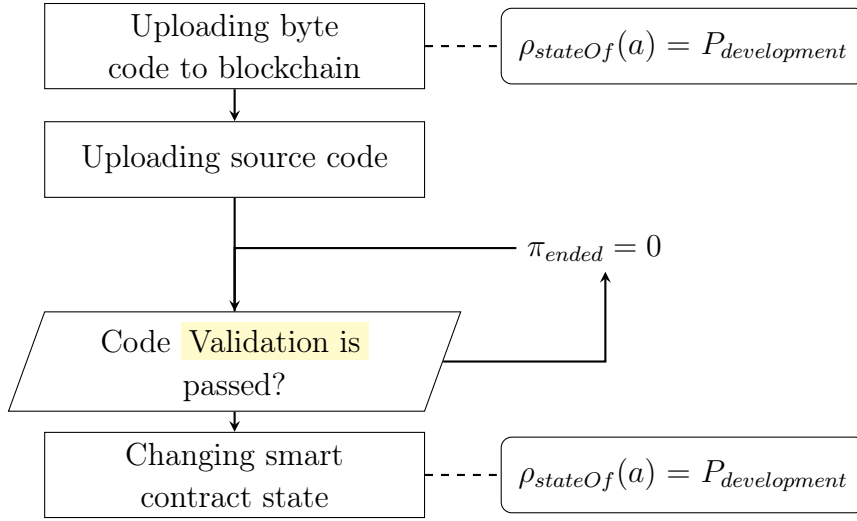
Any member of the network can upload a smart contract to the network, but such smart contract has $P_{development}$ status and it is available for interaction with affiliated addresses of network accounts. Initially this address belongs to the account of the user who uploaded the contract.

Smart contract interaction rules are regulated by allowance smart contract: ρ .

Any smart contract call initially requests allowance to make the call – $\rho_{allowTx}(T_{from}, T_{to})$

$$\begin{aligned}
 a &= \rho_{stateOf}(T_{to}) \\
 b &= \rho_{affiliateWith}(T_{to}) \\
 \rho_{allowTx}(T_{from}, T_{to}) &= \begin{cases} 1, & \text{if } a = P_{production} \\ 1, & \text{if } a = P_{development} \wedge T_{from} \in b \\ 0, & \text{otherwise} \end{cases} \quad (4.36)
 \end{aligned}$$

The author of a smart contract must provide source code for a check and pass the checking procedure in order to get $P_{production}$ status (see for details [Smart Contract Registry](#))



4.4 Decentralized Storage

4.4.1 Overview

InterPlanetary File System (IPFS) is a peer-to-peer network connecting remote servers in a single global decentralized storage platform. A significant amount of storage capacity will be required to store all the compositions in a decentralized manner and this can be achieved through IPFS.

Musereum is planning to utilize customized Clustered IPFS Swarm for the platform. Clustered IPFS nodes should create the necessary storage capacity for storing beats, compositions and videos, while providing faster access to them. Files stored in IPFS are divided into smaller chunks and each chunk is encrypted with keys. Based on our experience, storing files as chunks can significantly reduce download time and make them more secure.

The Musereum community will be involved in storing and streaming of the compositions and gain rewards in return from emission rate of ETM tokens.

4.4.2 Naming System

Musereum is planning to utilize *The Inter-Planetary Naming System* (IPNS) to provide simple way to stored files.

It allows you to store a reference to an IPFS file under the namespace of corresponding KeeperID (the hash of keeper's public key).

4.5 Децентрализованные приложения

Musereum DApps implement the code for record and storage layer as well as the interaction interface code. The protocol allows DApps to interact.

4.5.1 Governance dApp

Musereum starts from initial ceremony to set up a list of Notaries.

During the initial ceremony a master of ceremony creates a set of keys for each validator. He/She distributes them to validators one by one. Before each distribution of keys, he/she sends a transaction to a smart contract with a list of validators. That smart contract is used by consensus algorithm to determine if a validator has rights to participate in consensus and create blocks. The validator's smart contracts are used by other DApps, e.g. Governance DApp and Payout DApp.

A validator generates three keys in the Initial Ceremony DApp:

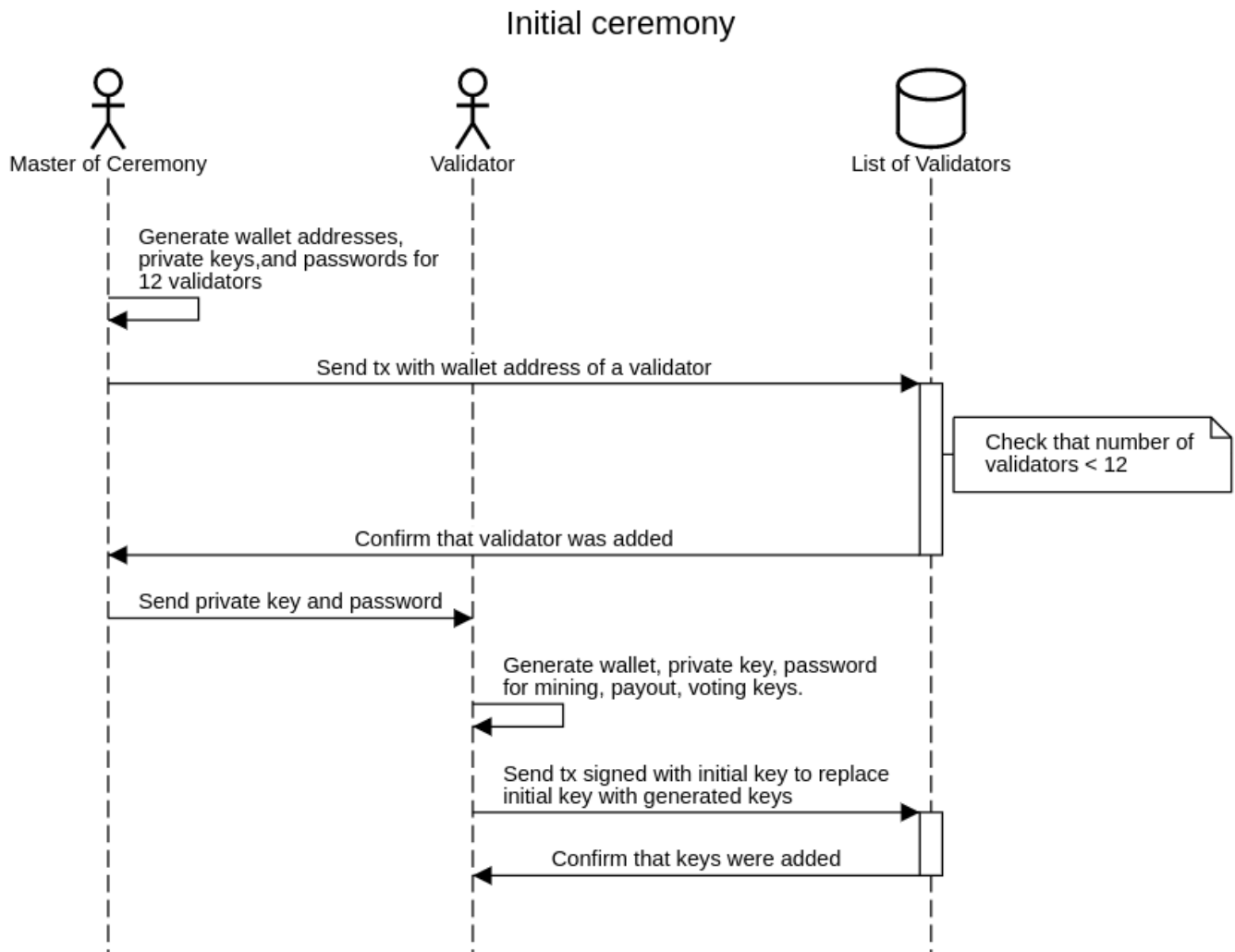
- mining key, required to participate in consensus and create blocks.
- voting key, required to create ballots and

vote on ballots.

- payout key, not required. Used in Payout DApp to send daily mined coins from the mining key to the payout key. If a mining node should be compromised, an attacker will get daily earnings or less.

All keys are generated on the client side and not transmitted over the Internet without a validator's permission and willingness. When keys are generated, the validator stores them on secure local storage, e.g. saves them to a hardware wallet and the password to a password manager. The validator signs a transaction to the validator's contract with the initial key, provided by the master of ceremony.

Initial ceremony is a required procedure to start a new network based on Oracles Network's ideas of independent validators.



Future changes of Notaries List will be committed through voting. Creation of voting ballot based on a request with assign form.

Valid notary of the Oracles Network fills out a form in DApp providing:

- mining key - mining key of a new or existing notary, which will be voted on
- affected key type - key type (mining, payout, or voting key) of a new or existing notary, which will be voted on
- memo - brief information about notary, which will be voted on
- action - add affected key to the network

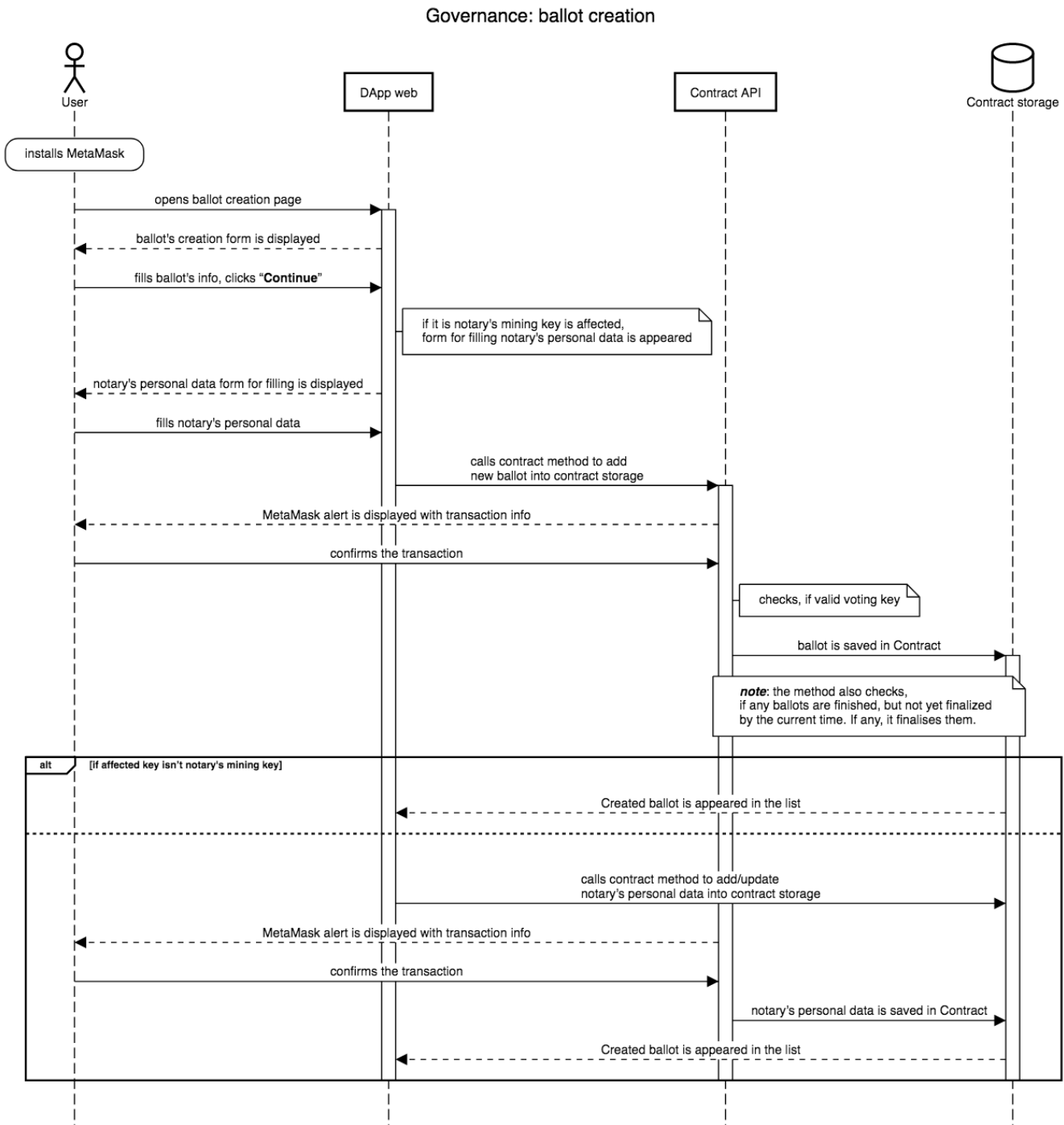
or remove it from the network

If the affected key type is mining key, the user will be asked to provide personal data of the notary (owner of this mining key) such as full name, physical address, U.S. state name, zip code, notary license ID, and notary license expiration date.

At the final step, one transaction to create a new ballot in Oracles contract will be pushed to the blockchain to add a new ballot after the user presses “Continue” button. It should be noted, that in case of a mining key, it will be two consistent transactions: to add personal data of a notary and a new ballot to contract.

User will see MetaMask popups equal to the number of transactions. After the confirmation and successful mining of the transaction by

existing validators, the user will see the created ballot in the list and be able to vote on it.



4.5.2 Contract Registry

The DApp for registering and recording smart contracts of the platform.

The DApp smart contract allows to receive:

1. The list of all registered smart contracts
2. Smart contract state associated with the address:
 - (a) **Unknown** – unknown state (default for unknown addresses in **Contract Registry**)
 - (b) **Development** – smart contract is registered but currently is under development (interaction is allowed for affiliated users only)

- (c) **Production** – smart contract is registered and **ready to interact**
- (d) **Halt** – smart contract is registered but halted (by author's decision)
- (e) **Suspend** – smart contract is registered but suspended (by auditor's decision)

3. A link to a source code repository
4. A link to ABI JSON for smart contract interaction
5. The list of affiliated auditors
6. A ballot **instance** to take decision about changing the state (*see [VotingDApp](#) for detail*)

Source code and smart contract checking

Smart contracts are validated in three ways:

- Source code validation for:
 1. Vulnerabilities
 2. Compliance with stated business logic
 3. Absence of algorithmic errors
- Compliance with law applicable to Musereum platform
- Compliance with Musereum concept

In order to approve a smart contract, all affiliated auditors must have a unanimous consent for all three points.

Validation process can be monitored in [Voting dApp](#) in corresponding **ContractValidationBallot**.

4.5.3 Musereum Name System (MNS)

The Ledger of names associated with addresses in Musereum platform is required to provide access to up-to-date protocol apps avoiding

the necessity to remember the internal address (20-character hex-address).

Name record can be created by any network

user for a random address. Number of names for the address is unlimited.

In order to protect names from improper use, the registration of a new name is charged in ETM according to the current price.

Name record is a MNS-derived smart contract:

`NameRecordContract`.

The smart contract contains the data about the name owner and associated network address and provides the following capabilities:

- for the owner to change the associated

address;

- for the owner to change the owner (transfer of right)

MNS limits available names:

- Name length cannot exceed 20 characters (or less if non-ASCII characters are used)
- Name length cannot be shorter than 4 characters

Interaction interface

`mapping (address => address) names` – (η_{names})

List of addresses associated with addresses of name registration smart contracts

`uint price` – (η_{price})

Name registration price in ETM tokens

`event NewName(address indexed name, byte20 ascii)` – $(\eta_{NewName})$

Event log of new name registration

`event AssociateWith(address indexed name, address indexed target)` – $(\eta_{AssociateWith})$

Event log of address associating with name registration smart contract

`function buyName(byte20 acsii) payable` – (η_{buy})

Method to buy name. Returns `address` if the buying is successful and `throw` in case of error.

$$\eta_{buy}(s) = \begin{cases} throw, & \text{if } s \in \eta_{NewName} \\ throw, & \text{if } |s| < 4 \\ throw, & \text{if } T_{value} < \eta_{price} \\ address, & \text{otherwise} \end{cases} \quad (4.37)$$

Smart-contract NameRecord – (η')

`address owner` – (η'_{owner})

Current name owner

`byte20 ascii` – (η'_{ascii})

Registered name (typicaly is 20 ASCII letter)

`address target` – (η'_{target})

Current association with address on Musereum platform

`function associate(address target)` – ($\eta'_{associate}$)

Change name association to the new address. Returns `true` if successful and `throw` in case of error.

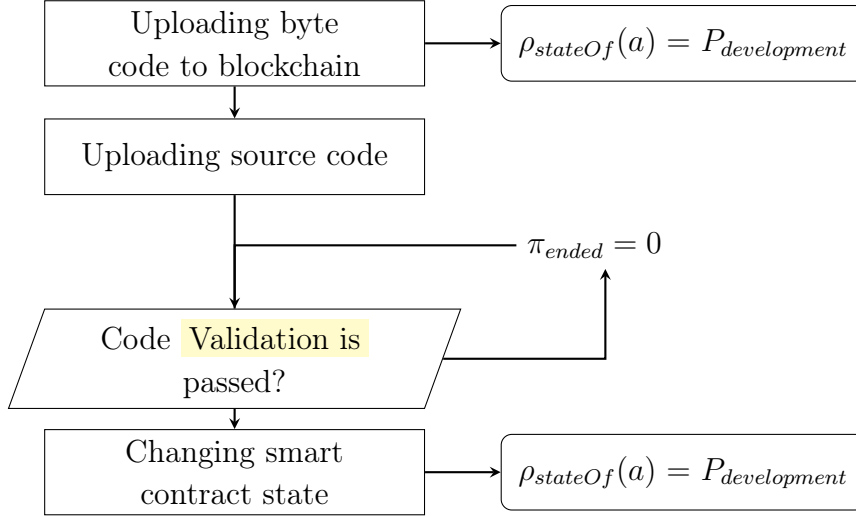
$$\eta'_{associate}(a) = \begin{cases} throw, & \text{if } T_{from} \neq \eta'_{owner} \\ throw, & \text{if } a = 0 \\ true, & \text{otherwise} \end{cases} \quad (4.38)$$

4.5.4 Voting dApp

Voting DApp consists of connected elements:

- `BallotsManager` (ϕ)
Ballots registry
- `Ballot` (π)
Abstract voting smart-contract
- `VotingRightsToken` (ϕ^τ)
Token issued to voting participants to prove the voting right
- **Voting GUI**
Web interface to interact with a ballots

In order to create a voting, the offer initiator must create a valid `Ballot` – a smart contract counting the votes and implementing `applyProposal` method to change the state based on the offer.



Смарт-контракт Ballot

An abstract smart contract describing the requirements for final implementation of voting offers.

`address votingToken` - (π_{token})

A voting-associated token providing the holder with the voting right.

`function vote(bool)` - (π_{vote})

External method to write the decision requesting the method.

`function applyProposal()` - (π_{apply})

Abstract application method based on the successful voting. It is implemented in child smart contracts.

`uint agreeVotesCount` - (π_{agree})

Number of tokens transferred for accepting the decision.

`uint rejectVotesCount` - (π_{reject})

Number of tokens transferred for rejecting the decision.

`uint endTime` - (π_{end})

Voting ending time.

`function isEnded()` - (π_{ended})

Returns 1 if voting time is over and 0 in other cases.

$$\pi_{ended} = \begin{cases} 1, & \text{if } \pi_{end} > \Gamma_{time} \\ 0, & \text{otherwise} \end{cases} \quad (4.39)$$

`function voteMajorityRule()` - (π_{rule})

Result of function $f(x) = 1/x$ where x is the ratio of "for" votes to total number of votes sufficient for accepting the decision:

$$\pi_{rule} = \begin{cases} \infty, & \text{if } x = 0 \\ \frac{1}{x}, & \text{otherwise} \end{cases} \quad (4.40)$$

`function compare()` - $(\pi_{compare})$

Returns 1 if number of votes for accepting the decision complies with voting conditions for accepting the decision.

$$\pi_{compare} = \begin{cases} 1, & \text{if } \pi_{agree} * \pi_{rule} > \pi_{agree} + \pi_{reject} \\ 0, & \text{otherwise} \end{cases} \quad (4.41)$$

`function isSuccess()` - $(\pi_{success})$

Returns 1 if voting is successful and 0 in other

cases.

$$\pi_{success} = \begin{cases} 1, & \text{if } \pi_{ended} \wedge \pi_{compare} \\ 0, & \text{otherwise} \end{cases} \quad (4.42)$$

4.5.5 Musical Assets Registry

The application is used to register musical assets on Musereum platform. The DApp consists of:

1. **MusicalAssetsRegistry** (MAR – ψ)
Root smart contract of the DApp. Performs factory and ledger functions.
2. **AssetContract** – (α)
Musical asset smart contracts produced in MAR.
3. **Decentralized labels** smart-contracts associated with **AssetContract**.
4. **AssetRegistryBallot** – ($\pi^{regAsset}$)
5. **AssetUnregistryBallot** – ($\pi^{unregAsset}$)
6. User Graphical Interface
web-applications within Musereum Wallet

Smart-contract MusicalAssetsRegistry

Smart contract tasks: Recording registered musical assets, creating musical asset registration requests through voting in **Voting dApp**.

Only system-authorized registrators

participate in the voting.

The ledger is fully public and is open for audit by any member of the network.

History data is accessible by creating a contract request for the event log:

event NewAsset(address indexed asset, uint indexed index) – ($\psi_{NewAsset}$)

Returns the musical asset adding event log to the ledger with assigned index in the ledger. *The event is triggered at the moment of creating registration request, not at the time of registration.*

$$\psi_{NewAsset} = \mathcal{E}(asset, index), \quad a \in \mathbb{A} \wedge i \in \mathbb{N} \quad (4.43)$$

event RegistryAsset(address indexed asset, uint indexed index) – ($\psi_{RegistryAsset}$)

Returns event log subset $\psi_{NewAsset}$ associated with asset addresses that have been successfully voted for adding to the public ledger.

$$\psi_{RegistryAsset} \subset \psi_{NewAsset} \quad (4.44)$$

event UnregistryAsset(address indexed asset, uint indexed index) – ($\psi_{UnregistryAsset}$)

Returns subset $\psi_{RegistryAsset}$ for musical assets that have been deleted from Musereum public ledger

based on the community decision.

$$\psi_{UnregistryAsset} \subset \psi_{RegistryAsset} \subset \psi_{NewAsset} \quad (4.45)$$

`mapping (address => uint) assets` – $(\psi_{assets}(a))$

Associated dictionary of musical assets added to the ledger.

$$\psi_{assets}(a) = \begin{cases} i, & \text{if } a \in \mathcal{S}(\psi_{NewAsset}, [asset]) \\ 0, & \text{otherwise} \end{cases}, \quad a \in \mathbb{A} \wedge i \in \mathbb{N}_1 \quad (4.46)$$

`address[] public assetsIndex` – $(\psi_{index}(i))$

The list of all musical assets added to the ledger. Asset index in the list is equal to the associated dictionary index minus 1.

$$i = \psi_{assets}(a), \quad a \in \mathbb{A} \wedge i \in \mathbb{N} \quad (4.47)$$

$$a = \psi_{index}(i - 1), \quad \text{if } i > 0 \quad (4.48)$$

Smart-contracts AssetContract

The blockchain record about registered musical asset. It contains all the data required to identify a musical asset and define the nature of the asset.

`string name` – (α_{name})

Name associated with the asset

`Multihash meta` – (α_{meta})

Hash of meta-data associated with the asset (stored in the decentralized storage)

$$\alpha_{meta} = \mathcal{O}(hash, hashFunction, size), \quad \alpha_{meta}[hash] \in \mathbb{B}_{32} \wedge \quad (4.49)$$

$$\alpha_{meta}[hashFunction] \in \mathbb{B} \wedge \quad (4.50)$$

$$\alpha_{meta}[size] \in \mathbb{B} \quad (4.51)$$

`event SetAssetType(uint8 indexed typeId)` – $(\alpha_{SetAssetType})$

Returns the asset defining event log containing asset type listing index (`AssetTypes` – α^{types}).

$$\alpha^{types} \in \mathbb{B} \quad (4.52)$$

$$\alpha_{SetAssetType} = \mathcal{E}(typeId), \quad i \in \alpha^{types} \quad (4.53)$$

`mapping (uint8 => bool) type` – $(\alpha_{type}(i))$

Associated dictionary of asset types.

$$\alpha_{type}(i) = \begin{cases} 1, & \text{if } i \in \alpha_{SetAssetType} \\ 0, & \text{otherwise} \end{cases}, \quad i \in \alpha^{types} \quad (4.54)$$

Смарт-контракт AssetRegistryBallot

Implementation of `Ballot` smart-contract to vote for accepting assets in Musereum ledger.

Смарт-контракт AssetUnregistryBallot

Implementation of `Ballot` smart-contract to vote for excluding assets from Musereum ledger.

4.5.6 Decentralized Autonomous Labels

Decentralized autonomous label is a governing smart contract with associated musical asset. DAL is able to:

1. Define the label charter that regulates decision making rules;
2. Issue and sell licenses for commercial use of associated musical asset;
3. Accumulate and distribute musical asset income,
4. Distribution and recording the rights for a musical asset with a digital proof-token.

The musical asset is managed on the basis of a token holder democratic voting.

DAL capabilities are implemented with the set of smart contracts:

- `DALRegistry` – (δ^R)
The ledger of registered decentralized autonomous labels
- `DALContract` – (δ)
Smart-contract of a decentralized autonomous label

- `DALToken` – (τ^δ)
Smart-contract of a token proving the holder's right to participate in a decentralized label
- Set of voting smart-contracts for `Ballot` implementation:
 - `CharterChangeVotingRulesBallot`
Voting offer to change voting rules and make changes to the charter
 - `CharterChangeProposalBallot`
Offer to make changes to the charter
 - `AddLicenseBallot`
Offer to create a new license
 - `CloseLicenseBallot`
Offer to stop selling the license
 - `ReplaceLicenseBallot`
Offer to replace/make changes to the license
- `DALAssetLicense` – (θ)
License smart contract for commercial use
- `DALAssetLicenseRule` – (θ')
License-derived smart contract to limit the use

Смарт-контракт DALRegistry

The ledger of registered autonomous labels is required to keep record and provide access to the label for the manager of associated musical asset.

```
mapping (address => address) assetLabels - ( $\delta_{labels}^R(a)$ )
```

Associated dictionary of created decentralized labels.

```
event CreateLabel(address indexed asset,  
                  address indexed label,  
                  uint indexed index) - ( $\delta_{CreateLabel}^R$ )
```

Returns set of decentralized label creation events. The event consists of: Address of created label, address of associated musical asset and assigned ledger index of the label.

$$\begin{aligned} \delta_{CreateLabel}^R \in \mathcal{E}(asset, label, index) \quad & asset \in \mathcal{S}(\psi_{RegistryAsset}, [asset]) \subset \mathbb{A}^\alpha \wedge \\ & label \in \mathbb{A}^\delta \wedge \\ & index \in \mathbb{N}_1 \end{aligned} \quad (4.55)$$

```
event CreateLabelBallot(address indexed label,  
                        address indexed ballot) - ( $\delta_{CreateLabelBallot}^R$ )
```

Returns set of decentralized label voting creation events.

The event consists of: Addresses of decentralized label and addresses of created voting.

$$\begin{aligned} \delta_{CreateLabelBallot}^R \in \mathcal{E}[label, ballot] \\ label \in \mathcal{S}(\delta_{CreateLabel}^R, [label]) \subset \mathbb{A}^\delta \wedge \\ ballot \in \mathcal{S}(\phi_{CreatedBallot}, [ballot]) \subset \mathbb{A}^\pi \end{aligned} \quad (4.56)$$

```
event FinishLabelVoting(address indexed label,  
                        address indexed ballot,  
                        bool indexed result) - ( $\delta_{FinishLabelVoting}^R$ )
```

Returns subset $\delta_{CreateBallot}^R$ for complete d voting with clear indication of success or failure of the offer.

$$\begin{aligned} f_Z(\mathbb{E}') &= \{\mathbb{E}'_i : \forall \mathbb{E}'_i[ballot][ended] > 0\} \\ \delta_{FinishLabelVoting}^R &= f_Z(\delta_{CreateLabelBallot}^R) \end{aligned} \quad (4.57)$$

```
event NewLabelLicense(address indexed label, address indexed license) - ( $\delta_{NewLicense}^R$ )
```

Returns set of events created by license decentralized label.

```
event RemoveLabelLicense(address indexed label, address indexed license) - ( $\delta_{RemoveLicense}^R$ )
```

Returns subset $\delta_{NewLicense}^R$ for licenses removed from sale.

Смарт-контракт DALContract

Smart contract for managing musical asset. Asset and associated entities are managed through proposing an offer for voting with preliminary distribution of `VotingRightsToken` among all community participants based on the balance of associated `DALToken`.

Decentralized label is created automatically based on the registration of the associated asset in the ledger `MusicalAssetsRegistry`.

1,000,000,000 proof-tokens are issued upon creation of a decentralized label; at that, in order to simplify the calculations the actual

number in associated user interfaces is divided by 10^7 .

Thus, sole label ownership is equal to 100.0000000 tokens, the minimal vote in the company is 0.00000001%.

`DALToken` holders' rights::

1. Participating in royalty distribution
2. Participating in voting for offers
3. Cession of tokens to other platform participants by exchanging for ETM or other tokens

Смарт-контракт DALToken

Decentralized label token is a [ERC-20](#) compatible token.

Required changes are added to standard ERC-20 token code to provide operation of distribution system:

1. One-to-one link with associated label in `label` field is maintained

2. `withdrawRevenueFor()` calls in `transfer(...)` и `transferFrom(...)` functions for sender and receiver are introduced to [improve distribution algorithm](#)
3. `makeVoteToken()` method is implemented to create `VotingRightsToken` associated with current balances

Royalty distribution

Royalty is ETM units received on the balance of linked decentralized autonomous label upon commercial activity of associated musical asset.

Every label participant can implement the right to receive royalty share at any moment of time according to the current amount of tokens.

Royalty distribution	
$a = T_{from}$	Initial withdrawal transaction
$d = \delta(T_{to})$	DAL instance
$r = \tau^\delta(d_{token})$	Rights tokens of corresponding DAL
$h = \mathbb{K}_n$	Height of latest block
$b_h = r_{balanceOf}(a)$	Balance of DAL rights tokens at block h
$t_h = r_{totalSupply}$	Total supply of rights token (typically is 10^9)
$s_h = \frac{b_h}{t_h}$	Stake of a at moment h
$I_h = x$	I_h income of a DAL at moment h
$T_h = \sum_{i=1}^h I_i$	Total lifetime income of DAL
$W_h = s * (T_h - \sum_{i=1}^{h-1} W_i)$	Available withdrawal amount at moment h

Distribution algorithm optimization

An additional associated list `incomeAtLatestWithdrawalAddress` (required by business logic and only keeping the value at the moment of the latest fund withdrawal for associated address) is introduced in the smart contract to optimize it and avoid calculating W_h for every unique h .

for associated holders.

If rights are transferred to a third party (token transfer), it triggers automatic calculation algorithm of available royalty for the current and the future holder.

This method allows to calculate royalty only

Голосование за принятие решений

Устал децентрализованного лейбла

Продажа лицензий и смарт-контракт DALAssetLicense

Musereum licensing has the capability to conduct the following types of licensing.

Sale of license for commercial use of associated musical asset is the primary task of a

decentralized label, at that receiving royalty is the basic motivation for existence. Все лицензии выпускаются в виде экземпляра смарт-контракта `DALAssetLicense` через интерфейс управления `DALContract`.

All licenses are issued as an instance of `DALAssetLicense` smart-contract through `DALContract` management interface. The task of `DALAssetLicense` root contract is to define the nature of license, acquisition rules

and limitations of use. `DALAssetLicense` is a container that provides holder with an unlimited license for commercial use, at that limitations are described in derivative smart contracts: `DALAssetLicenseRule`.

DALAssetLicenseRule

It is an abstract smart contract defining the limitation of a parental license; there are 7 basic types of limitations:

1. `EnumerateRule` – enumerate limitation
2. `RangeRule` – range limitation
3. `ValueRule` – value limitation
4. `AddressRule` – address(es) limitation
5. `TextRule` – descriptive limitation
6. `AllOfRule` – grouping limitation according to type: all from the group
7. `AnyOfRule` – grouping limitation according to type: any from the group

Final limitation can be selected from ready options or it can be created by a decentralized label for a certain license.

Сравнение ограничений

$$a = \theta'(original), \quad b = \theta'(target)$$

$$\theta'_{fitWith}(b) = \begin{cases} 1, & \text{if } a_{type} = b_{type} \wedge f_z(a, b) \\ 0, & \text{otherwise} \end{cases} \quad (4.59)$$

EnumerateRule

$$f_z(a, b) = \begin{cases} a_{items} \subset b_{items}, & \text{if } a_{all} \\ |\{i : \forall a_{items}[i] \in b_{items}\}| > 0, & \text{otherwise} \end{cases} \quad (4.60)$$

RangeRule

$$f_z(a, b) = a_{min} > b_{min} \wedge b_{max} > a_{max} \quad (4.61)$$

ValueRule

$$f_z(a, b) = \begin{cases} a_{value} = b_{value}, & \text{if } a_{equal} \\ a_{value} < b_{value}, & \text{if } a_{less} \\ a_{value} > b_{value}, & \text{if } a_{more} \end{cases} \quad (4.62)$$

AddressRule

$$f_z(a, b) = a_{addresses} \in b_{addresses} \quad (4.63)$$

AllOfRule

$$f_z(a, b) = ||\{\top : a_{items}[i]_{fitWith}(b) = \top\}|| = ||a|| \quad (4.64)$$

AnyOfRule

$$f_z(a, b) = ||\{\top : a_{items}[i]_{fitWith}(b) = \top\}|| > 0 \quad (4.65)$$

Таблица 4.1: Предзаготовленные типовые ограничения лицензий

Наименование	Описание
EnumerateRule	
LocationRule	Geographical use limitation and derivatives: CountryRule and CityRule
ChannelRule	Playback channel limitation
MediaRule	Media type limitation
GenreRule	Media genre limitation (used for cinemas, theatres etc.)
UsageRule	Usage type limitation
RangeRule	
PlayTimesRule	Playback quantity/period limitation
LifetimeRule	Validity of license limitation
NumberOfCopyRule	Number of copies limitation
PowerOfRule	Project budget and (or) buyer income limitation
ValueRule	
PriceRule	License price

Устав децентрализованного лейбла

Любой децентрализованный лейбл

4.5.7 Soundchain

Soundchain is the first project that aims to embrace Musereum Foundation. The primary application of Soundchain is to allocate royalty to decentralized labels for listening to musical assets.

Soundchain operation is determined by

numerous smart-contracts and interaction interfaces, consider PayPerPlay smart-contract that is used to:

- record every listening to a musical asset;
- generate the royalty payment list from

Let n is a moment of time and

Let r is a money available to pay-per-play for

Set of plays a , where a_i is a tuple:

$$a = \{a_i \mid \forall i \in a, a_i \equiv \mathcal{O}[listener, asset, times, sign]\} \quad (4.66)$$

$$b = \{b_i \mid \forall i \in a, b_i = f_{pay}(a_i), b_i \in (0; 1)\} \quad (4.67)$$

$$c = \sum_{i=0}^{|b|} b_i, \quad d = \frac{r}{c} \quad (4.68)$$

Finally p is a set of payments:

$$p = \{p_i \mid \forall i \in b, p_i = d * b_i\}, \quad r = \sum_{i=0}^{|p|} p_i \quad (4.69)$$

Now we need to define what is f_{pay} function. The function result depends on two variables: Listener ratio and label ratio:

$$f_{pay}(a) = \min(f_{listenerCoef}(a)), \quad (4.70)$$

$$f_{labelCoef}(\delta_{labels}^R(a_{asset})) \quad (4.71)$$

Listener ratio determines how much trust the platform puts in the listener:

1. Account personification: Full name, location;
2. Phone number confirmation (Proof-of-SMS);
3. Geographical location confirmation: verification of documents;
4. The amount of financial incentive in the future of the platform: Amount of ETMs, amount of held tokens of a decentralized organizations.

When calculating, label ratio takes into account:

1. License sale total revenue of associated asset;
2. License sale revenue for the latest week;
3. Soundchain artificial label evaluation ratio (additional depositing, fund burning etc.)

(4.72)