УДК: 004.03, 004.72, 004.4’2

О практических подходах к созданию доверенных защищенных цифровых платформ

D.F. Aliev, A.Yu. Shcherbakov

On Practical Approaches to Creating Trusted Secure Digital Platforms

Abstract. *This article is devoted to the consideration of a secure trusted platform for reliable exchange of documents and their storage in a distributed registry with minimal borrowing of third-party software products, based on domestic standards and indicators of programming and information security quality, using domestic cryptographic algorithms. It is shown that within the framework of the platform it is possible and reasonable to implement smart contracts and chain code. The advantages of zero-coding as a way to create a minimum viable version of a platform product, understandable and verifiable for the purpose of further development of platform solutions, are considered.*

Keywords: *platform, trust, cryptographic information protection facility (CIPF), electronic signature (ES), zero-coding, distributed ledger (blockchain), script.*

Д.Ф.Алиев[1]
А.Ю.Щербаков[2]
[1]Доктор философии в области бизнес-права (PhD), доктор делового администрирования в области финансов (DBA), кандидат экономических наук, первый проректор Федерального государственного бюджетного образовательного учреждения высшего образования «Российский государственный социальный университет».
E-mail: kharchenkoDD@rgsu.net
[2]Доктор технических наук, профессор кафедры комплексной безопасности критически важных объектов РГУ нефти и газа (НИУ) имени И.М. Губкина, заведующий кафедрой когнитивно-аналитических и нейро-прикладных технологий РГСУ, ведущий научный сотрудник Государственного университета управления, главный научный сотрудник РАН (ИТМиВТ им.С.А.Лебедева).
E-mail: x509@ras.ru

Аннотация. *Статья посвящена рассмотрению вопросов реализации защищенной доверенной платформы достоверного обмена документами и хранения их в распределенном реестре при минимальном заимствовании программной продукции третьих фирм, на основе отечественных стандартов и показателей качества программирования и информационной безопасности, с использованием отечественных криптографических алгоритмов. Показано, что в рамках платформы возможна и целесообразна реализация смарт-контрактов и чейн-кода. Рассмотрены преимущества зерокодинга как способа создания минимально жизнеспособной версии платформенного продукта, понимаемого и верифицируемого в целях дальнейшей разработки платформенных решений.*

Ключевые слова: *платформа, доверенность, средство криптографической защиты (СКЗИ), электронная подпись (ЭП), нулевое программирование (зеро-кодинг), распределенный реестр (блокчейн), скрипт.*

ВВЕДЕНИЕ

В текущей геополитической ситуации крайне важным является курс на создание унифицированных программно-аппаратных платформ для обмена документами и их хранения, основанных на следующих принципах:

- полное владение исходным кодом платформы, минимальное использование продуктов третьих фирм,
- компактность кода, возможность его доработки и технической поддержки,
- опора на отечественные криптографические алгоритмы, проектирование и реализация СКЗИ сразу по требованиям регуляторов,
- минимальное использование внешних инфраструктурных решений (удостоверяющих центров) в целях повышения надежности и доступности платформ.

При этом весьма существенным требованием является доверенность [1], т.е. возможность верификации свойств платформы как программно-аппаратного проекта на всех этапах его жизненного цикла.

Доверенность включает минимальное заимствование программной продукции третьих фирм и опору на отечественные стандарты и показатели качества программирования и информационной безопасности, в первую очередь на отечественные криптографические алгоритмы.

С другой стороны, для повышения уровня доверия к разработке необходимо использование средств разработки и расширения функционала платформы таким образом, чтобы не нарушить базовые свойства платформ, такие как локальная

замкнутость на уровне субъектов [2]. Для этого целесообразно использовать механизмы вызова сценариев (скриптов), а также в случае использования архитектуры распределенного хранения или распределенного реестра – извлечение исполняемых модулей и скриптов, их содержащих, из доверенных хранилищ с проверкой их целостности.

Предлагаемая читателю статья посвящена практике создания программной части доверенных и защищенных цифровых платформ в соответствии с указанными принципами.

В качестве примера рассмотрим квалиметрическую платформу Сешат [3].

Напомним основные бизнес-процессы платформы.

ОСНОВНЫЕ ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ (БИЗНЕС-ПРОЦЕССЫ) ПЛАТФОРМЫ

При установке и разворачивании платформы регистрируется как минимум один администратор системы. Один из администраторов (в случае присутствия нескольких) производит верификацию регистрирующихся пользователей и экспертов (администратор верификации).

1. Регистрация пользователей

Пользователь вводит свое имя, получает анонимное имя UserX, вырабатывает свой ключ в ключевом контейнере, защищенном паролем. Далее пользователь загружает в систему свои квалификационные документы (диплом об образовании, ученом звании и степени, список трудов, некоторые труды, которые он считает важными), после чего заполняет анкету, где указывает тематики, по которым он дает рецензии или проводит экспертизы (если он претендует на роль эксперта).

Администратор верификации производит верификацию загруженных документов и личных/профессиональных данных самого пользователя и завершает его регистрацию. Пользователь получает статус – одну или несколько ролей и перечень тематик, в которых он компетентен. Параметры пользователя отражены в его профиле.

2. Загрузка статей (документов)

Статьи загружаются в систему для рецензирования или депонирования (во втором случае рецензирование не требуется, фиксируется приоритет; объектами депонирования также являются статьи или работы с требованиями сохранения конфиденциальности).

При загрузке производится индексирование статьи и установление ее принадлежности к тематике.

Также предпринимаются меры по предотвращению загрузки статей которые уже есть в системе (например, путем сравнения с загруженными ранее статьями).

После успешной загрузки администратор рецензий направляет ее на рецензирование или депонирование.

Направление на рецензирование происходит тремя возможными способами: всем экспертам тематики, избранным экспертам тематики, желающим экспертам тематики. Этот выбор делается автором при загрузке и оплачивается внутренними токенами платформы. Соответственно, эксперт тематик выставляет в своем профиле параметр, желает ли он в текущее время участвовать в рецензировании.

При загрузке и поиске работает ядро семантического сервиса – процессы индексирования и сравнения текстов.

3. Рецензирование статей

Рецензент по публичной методике выставляет оценки статье и обосновывает их в виде рецензии, которую подписывает своей электронной подписью (ЭП).

Рецензии и оценки модерируются администратором рецензий (например, в целях дополнительной анонимизации эксперта для исключения выяснения его личности и вероятного давления на него).

Возможен режим без модерации, когда собираются оценки и рецензии не менее трех экспертов и автору направляется только усредненное мнение (оценка).

Весь документооборот на платформе выполняется в виде файловых операций между соответствующими директориями пользователей и статей.

4. Начисление токенов за рецензии и статистика

После формирования рецензии и оценки эксперты получают токены в соответствии с их рейтингом.

В случае если реальных токенов (начисленных авторами или спонсорами) не хватает для оплаты работы экспертов, начисленные токены маркируются как кредитные (вексель) и учитываются по мере появления реальных токенов в хронологической очередности оплаты услуг экспертов.

Все действия в системе регистрируются в журналах.

5. Работа внешних заказчиков

Внешний заказчик регистрируется отдельно и также проходит процедуру верификации как физическое или юридическое лицо. Он также является донором (поставщиком) токенов в систему.

Он имеет возможность знакомиться со статисти-

кой поступления статей, оценками за них, а также отдельно оплачивать процедуры развернутого поиска – когда его текстовый запрос сравнивается со статьями тематики с учетом заданного порога их рейтинга (средней экспертной оценки). Поиск может происходить и только по рейтингу статей.

К внешним заказчикам также относятся эксперты государственных учреждений и сервисов, которые обращаются за фактами приоритета или плагиата, различными характеристиками научных достижений авторов или экспертов.

РЕАЛИЗАЦИЯ БИЗНЕС-ПРОЦЕССОВ В МОДУЛЯХ ПЛАТФОРМЫ

Рассмотрим реализацию описанных бизнес-процессов в модулях платформы.

Регистрация пользователя, генерация сетевого имени и создание личного ключа для подписи

Модуль gk0 – генерация личного ключа.

Результатом работы является появление в системе пользователя с уникальным анонимным именем и его ключа. Ключ размещен в ключевом контейнере с именем, совпадающим с анонимным именем пользователя, и защищен паролем пользователя.

Модуль gks – отвечает за генерацию сетевого ключа.

Данный модуль обеспечивает связь пользователя с системой. В качестве корреспондента для информационного обмена должен быть указан пользователь, зарегистрированный как администратор, который обозначает систему приема статей или подключает рецензента, или внешнюю организацию.

В модуле формируется длинное сетевое имя, которое описывает связь между пользователем и системой и состоит из анонимного имени отправителя и получателя статей. Имя также помещается в файл Netname, который может быть использован модулем динамической генерации смарт-контрактов, который описан ниже.

Формируется ключ для передачи данных и их верификации для записи в блокчейн.

Модуль chpin

Меняет ключ доступа к контейнеру.

Модуль rsen

Выполняет подписание статьи и отправку ее в систему.

Модуль rrec

Производит прием статьи и проверку подписи под ней, при необходимости – экстракцию статьи из подписанного файла. Необходимость экстракции зависит от политики – статья помещается в реестр с подписью автора или только с подписью администратора системы.

Модуль areestr

Размещает статью в распределенном реестре.

При размещении статьи формируется квитанция, которая направляется автору и рецензентам.

В системе также реализованы модули извлечения из реестра с проверкой цепочки и модуль динамической генерации смарт-контракта.

Семантическая часть системы опирается на вызов модулей индексации и сравнения текстов, которые описаны в статье [4].

Работа платформы демонстрируется динамическим смарт-контрактом 1first.bat, который создает двух пользователей andrey и boss, затем все необходимые имена и ключи. Смарт-контракт также передает ключ в защищенном виде пользователю, меняет пароли контейнеров, подписывает статью и передает ее в систему, проверяет подпись под статьей, размещает ее в реестре и выдает пользователю квитанцию с приоритетом размещения.

Далее статья должна быть проиндексирована семантическим модулем, отнесена к некоторой тематике (сравнением индексированной статьи с шаблоном тематики), после чего автоматизированно или вручную (администратором) обработана для рецензирования (с экстракцией автора и списка литературы, по которой также можно опознать автора).

Приведем процесс разворачивания платформы в виде последовательного вызова модулей (скрипта).

```
rem First Demo SmartContract for creation one user Andrey and operator (Boss)
rem Create Main key
gk0 boss BossPas
rem Create Dir for Boss
md boss
rem Create Call for copy key to psv extanton file
mks a.bat [copy netname [*.psv
call a.bat
mks a.bat [copy [*.psv [boss\boss
call a.bat
pause
del *.psv
rem Create Andrey key
gk0 Andrey 1234
rem Create Dir for Andrey
md andrey
rem Create Call for copy key to psv extanton
mks a.bat [copy netname [*.psv
call a.bat
```

```
pause
mks a.bat [copy [*.psv [andrey\andrey
call a.bat
del *.psv
rem Create Call for Andrey Network key generation
use passw1 method
mks a.bat [gks [Andrey [boss passw1
call a.bat
rem Create Call for copy key to psw extanton
mks a.bat [copy netnames [*.psw
call a.bat
rem Send psw to Boss and to Andrey dir
mks a.bat [copy [*.psw [andrey\*.*
call a.bat
mks a.bat [copy [*.psw [boss\*.*
call a.bat
del *.psw
rem Distribute ChangePin prog
copy chpin.exe andrey\*.*
copy chpin.exe boss\*.*
rem Distribute MakeSmart prog
copy mks.exe andrey\*.*
copy mks.exe boss\*.*
rem Distribute SendReestr prog
copy rsen.exe andrey\*.*
rem Distribute RecordReestr prog
copy rrec.exe boss\*.*
rem Distribute AddReestr prog
copy areestr.exe boss\*.*
rem Normalize psw files names
cd andrey
copy *.psw a2boss
mks a.bat [chpin [a2boss ..\passw1 [1234
call a.bat
pause
cd ..
cd boss
copy *.psw frandrey
mks a.bat [chpin [frandrey ..\passw1 [boss
call a.bat
pause
cd ..
rem Send Article file (c2.txt) to boss
copy c2.txt andrey\*.*
cd andrey
rem Sign Article file
rsen a2boss 1234 c2.txt c2.sen
rem Send Signed File to boss
copy c2.sen ..\boss\*.*
cd ..
cd boss
rem Check Article signature and extract it
rrec frandrey boss c2.sen-x
rem Add Article in Reestr and create notify file
areestr.exe boss BossPas c2.txt
rem Send notify file to andrey
copy *.kvt ..\andrey\*.*
pause
```

Интересно рассмотреть программу mks, которая динамически формирует строку в описанном выше файле.

```
int main(int argc, char *argv[])
   {
    int td[6];
    int i,j;
    char ss[16][130],itog[2048];

//  Current Time and Date
    GetTimeDate(td);
    AppLogT(LOGNAME,"Start Gks");
   if((argc<3)||(argc>12))
     {
        printf("Use format: mks file part1 part2 ...
part10\n");
      AppLogT(LOGNAME,"Bad format");
      return(-2);
     }
   strcpy(ss[0],argv[1]);
    for(i=2;i<argc;i++)
      {
       if(argv[i][0]=='[')
        {
//  printf("%d %s\n",strlen(argv[i]),argv[i]));
    for(j=0;j<strlen(argv[i]);j++) ss[i-1][j]=argv[i][j+1];
        }
       else          ReadS(argv[i],ss[i-1]);
      }
//    for(i=0;i<argc-1;i++)
//      printf("%s\n",ss[i]);
    itog[0]=0;
    for(i=1;i<argc-1;i++)
      {
       strcat(itog,ss[i]);
       strcat(itog," "  );
      }
   AppS(ss[0],itog,strlen(itog));
    return(0);
   }
```

В данном примере достаточно компактного кода мы выделяем из командной строки аргументы, начинающиеся с символа квадратной скобки и интерпретируем их как текстовые строки (имена), а остальные строки понимаем как имена файлов и переносим в итоговую строку их содержание.

Например

```
mks a.bat [gks [Andrey [boss passw1
```

Трансформируется в вызов

call a.bat

gks Andrey boss VeryLongPassForExchangKeys

В результате мы получаем возможность динамического изменения интерпретируемого файла (скрипта) и его реального исполнения в рамках модулей платформы с заданными параметрами и аргументами. При этом, как приведено в примере, мы можем вызывать сам модуль генерации строк чейн-кода с необходимыми аргументами, то есть реализовать рекурсию.

Покажем, что таким образом в рамках платформы можно реализовывать смарт-контракт или чейн-код.

Данный подход также демонстрирует процесс «нулевого» программирования, когда для реализации гибкой логики платформы используются уже встроенные в нее возможности.

ПОНЯТИЕ СМАРТ-КОНТРАКТА И ЧЕЙН-КОДА

Будем говорить, что платформа [5] поддерживает смарт-контракты, в терминах Hyper Ladger Fabric (HLF) — чейнкоды (chaincode), создаваемые на языках общего назначения, таких как Golang, JavaScript, Java, в отличие, от, например, Ethereum, в котором используется контрактно-ориентированный, ограниченный по функциональности язык Solidity (LLL, Viper и др).

С точки зрения разработчика блокчейн-приложение состоит из двух основных частей, представленных ниже.

On-chain модули — смарт-контракты (программы), работающие в изолированном окружении блокчейн-сети, определяющие правила создания и состав атрибутов транзакций. В смарт-контракте основные действия — это чтение, обновление и удаление данных из состояния блокчейн-сети. Следует подчеркнуть, что удаление данных оставляет информацию о том, что эти данные присутствовали.

Off-chain модули — приложение, взаимодействующее с блокчейн-средой через SDK (software development kit, набор средств для разработки ПО). Под взаимодействием понимается вызов функций смарт-контрактов и наблюдение за событиями смарт-контракта — внешние события могут порождать изменение данных в смарт-контракте, в то же время события в смарт-контракте могут инициировать действия во внешних системах.

Чтение данных производится обычно через “домашний” узел блокчейн сети. Для записи данных приложение отправляет запросы на узлы организаций, участвующих в “политике одобрения” определенного смарт-контракта.

Итак, разницу между чейн-кодом (цепным кодом) и смарт-контрактом можно описать следующим образом.

Чейн-код - это программа, написанная, как правило, на Go или на других языках программирования, таких как Java, которая реализует предписанный интерфейс.

Смарт-контракт - это набор кода (его функций) и данных (его состояния), который находится по определенному адресу в блокчейне Ethereum.

В некотором смысле цепной код также можно считать смарт-контрактом, потому что он обрабатывает бизнес-логику, согласованную участниками, так же, как смарт-контракт.

В нашем случае приводимый фрагмент более корректно отнести к чейн-коду.

Приведем результат выполнения описанного выше чейн-кода в составе скрипта. Он демонстрирует вызов скрипта, содержащего как фиксированные приложения, описанные выше, так и динамически созданные строки скрипта. Кроме того, в данном случае мы используем также модули записи в распределенный реестр и формирование квитанции для пользователя, что его информация помещена в распределенный реестр с указанием номера звена (цепочки). Приведенный пример также относится к наукометрической платформе Сешат (Семантически-Естественная Шкала Анализа Текстов) [3].

C:\DEV\SESHAT>rem First Demo SmartContract for creation one user Andrey and operator (Boss)

C:\DEV\SESHAT>rem Create Main key

Процедура создания основного ключевого контейнера оператора системы и распределенного реестра.

C:\DEV\SESHAT>gk0 boss BossPas

Ok Test Random

UserName: boss

UserPIN: BossPas

UserX: 429398babb45cb58

Ok Test Random

Successful UserFile create!

Creation UserKey Time-> 14:07:31 21.10.2022

Обратим внимание на то, что сформировано анонимное имя для оператора платформы 429398babb45cb58.

C:\DEV\SESHAT>rem Create Dir for Boss

C:\DEV\SESHAT>md boss

C:\DEV\SESHAT>rem Create Call for copy key to psv extanton file

C:\DEV\SESHAT>mks a.bat [copy netname [*.psv

Программа (модуль) динамического формирования скрипта (чейн-кода) формирует процедуру копирования полученного контейнера в файл с расширением psw.

```
C:\DEV\SESHAT>call a.bat
Вот реальный результат вызова
C:\DEV\SESHAT>copy 429398babb45cb58 *.psv
Скопировано файлов:         1.
C:\DEV\SESHAT>mks a.bat [copy [*.psv [boss\boss
C:\DEV\SESHAT>call a.bat
C:\DEV\SESHAT>copy *.psv boss\boss
429398babb45cb58.psv
Скопировано файлов:         1.
C:\DEV\SESHAT>rem Create Andrey key
```

Аналогичная процедура для пользователя Andrey

```
C:\DEV\SESHAT>gk0 Andrey 1234
Ok Test Random
UserName: Andrey
UserPIN: 1234
UserX:1b4acb08dc2df32f
Ok Test Random
Successful UserFile create!
Creation UserKey Time-> 14:07:33 21.10.2022
C:\DEV\SESHAT>rem Create Dir for Andrey
C:\DEV\SESHAT>md andrey
C:\DEV\SESHAT>rem Create Call for copy key to psv
extanton
C:\DEV\SESHAT>mks a.bat [copy netname [*.psv
C:\DEV\SESHAT>call a.bat
C:\DEV\SESHAT>copy 1b4acb08dc2df32f *.psv
Скопировано файлов:         1.
C:\DEV\SESHAT>mks a.bat [copy [*.psv [andrey\
andrey
C:\DEV\SESHAT>call a.bat
C:\DEV\SESHAT>copy *.psv andrey\andrey
1b4acb08dc2df32f.psv
Скопировано файлов:         1.
C:\DEV\SESHAT>del *.psv
```

Следующий фрагмент создает на стороне системы ключевой контейнер для подписания передаваемых от пользователя к системе документов (статей).

```
C:\DEV\SESHAT>rem Create Call for Andrey Network
key generation use passw1 method
C:\DEV\SESHAT>mks a.bat [gks [Andrey [boss
passw1
C:\DEV\SESHAT>call a.bat
C:\DEV\SESHAT>gks Andrey boss
VeryLongPassForExchangKeys
Ok Test Random
MyName: Andrey
FriendName: boss
UserPIN: VeryLongPassForExchangKeys
MyUserX:1b4acb08dc2df32f
FriendUserX:429398babb45cb58
1b4acb08dc2df32f429398babb45cb58
Ok Test Random
User NetName:
1b4acb08dc2df32f429398babb45cb58
Successful NetFile create!
Creation NetKey Time-> 14:07:36 21.10.2022
C:\DEV\SESHAT>rem Create Call for copy key to psw
extanton
C:\DEV\SESHAT>mks a.bat [copy netnames [*.psw
C:\DEV\SESHAT>call a.bat
C:\DEV\SESHAT>copy 1b4acb08dc2df32f429398ba
bb45cb58 *.psw
Скопировано файлов:         1.
C:\DEV\SESHAT>rem Send psw to Boss and to
Andrey dir
C:\DEV\SESHAT>mks a.bat [copy [*.psw [andrey\*.*
C:\DEV\SESHAT>call a.bat
C:\DEV\SESHAT>copy *.psw andrey\*.*
1b4acb08dc2df32f429398babb45cb58.psw
Скопировано файлов:         1.
C:\DEV\SESHAT>mks a.bat [copy [*.psw [boss\*.*
C:\DEV\SESHAT>call a.bat
C:\DEV\SESHAT>copy *.psw boss\*.*
1b4acb08dc2df32f429398babb45cb58.psw
Скопировано файлов:         1.
C:\DEV\SESHAT>del *.psw
C:\DEV\SESHAT>rem Distribute ChangePin prog
```

Следующий фрагмент распределяет исполняемые модули для вызова на стороне системы и на стороне клиента.

В первую очередь это модуль смены пароля контейнера с «длинного пароля» на разные пароли на стороне системы и клиента.

```
C:\DEV\SESHAT>copy chpin.exe andrey\*.*
Скопировано файлов:         1.
C:\DEV\SESHAT>copy chpin.exe boss\*.*
Скопировано файлов:         1.
C:\DEV\SESHAT>rem Distribute MakeSmart prog
C:\DEV\SESHAT>copy mks.exe andrey\*.*
Скопировано файлов:         1.
C:\DEV\SESHAT>copy mks.exe boss\*.*
Скопировано файлов:         1.
C:\DEV\SESHAT>rem Distribute SendReestr prog
C:\DEV\SESHAT>copy rsen.exe andrey\*.*
Скопировано файлов:         1.
C:\DEV\SESHAT>rem Distribute RecordReestr prog
C:\DEV\SESHAT>copy rrec.exe boss\*.*
Скопировано файлов:         1.
C:\DEV\SESHAT>rem Distribute AddReestr prog
C:\DEV\SESHAT>copy areestr.exe boss\*.*
```

```
    Скопировано файлов:         1.
    C:\DEV\SESHAT>rem Normalize psw files names
    C:\DEV\SESHAT>cd andrey
    C:\DEV\SESHAT\ANDREY>copy *.psw a2boss
    1b4acb08dc2df32f429398babb45cb58.psw
    Скопировано файлов:         1.
    C:\DEV\SESHAT\ANDREY>mks a.bat [chpin [a2boss
..\passw1 [1234
    C:\DEV\SESHAT\ANDREY>call a.bat
    C:\DEV\SESHAT\ANDREY>chpin          a2boss
VeryLongPassForExchangKeys 1234
    Ok Test Random
    UserKey: a2boss
    NetName:1b4acb08dc2df32f429398babb45cb58
    UserFile create at-> 14:07:36 21.10.2022
    Ok Test Random
    NetName:1b4acb08dc2df32f429398babb45cb58
    NetName:1b4acb08dc2df32f429398babb45cb58
    Successful Change PIN!
    C:\DEV\SESHAT\ANDREY>cd ..
    C:\DEV\SESHAT>cd boss
    C:\DEV\SESHAT\BOSS>copy *.psw frandrey
    1b4acb08dc2df32f429398babb45cb58.psw
    Скопировано файлов:         1.
    C:\DEV\SESHAT\BOSS>mks a.bat [chpin [frandrey
..\passw1 [boss
    C:\DEV\SESHAT\BOSS>call a.bat
    C:\DEV\SESHAT\BOSS>chpin          frandrey
VeryLongPassForExchangKeys boss
    Ok Test Random
    UserKey: frandrey
    NetName:1b4acb08dc2df32f429398babb45cb58
    UserFile create at-> 14:07:36 21.10.2022
    Ok Test Random
    NetName:1b4acb08dc2df32f429398babb45cb58
    NetName:1b4acb08dc2df32f429398babb45cb58
    Successful Change PIN!
    C:\DEV\SESHAT\BOSS>cd ..
    C:\DEV\SESHAT>rem Send Article file (c2.txt) to
boss
    C:\DEV\SESHAT>copy c2.txt andrey\*.*
    Скопировано файлов:         1.
    C:\DEV\SESHAT>cd andrey
    C:\DEV\SESHAT\ANDREY>rem Sign Article file
    C:\DEV\SESHAT\ANDREY>rsen a2boss 1234 c2.txt
c2.sen
    Ok Test Random
    Len= 285633
    UserFile: a2boss
    Successful UserConainer read!
    Creation UserConainer Time: 14:07:36 21.10.2022
    Network name: 1b4acb08dc2df32f429398babb45
cb58
    Length: 285680
    Ok OutFile write
    Ok OutFile read
    Ok OutFile check
    Registration User File in: 14:07:44 21.10.2022
    User File Name: c2.txt
    C:\DEV\SESHAT\ANDREY>rem Send Signed File to
boss
    C:\DEV\SESHAT\ANDREY>copy c2.sen ..\boss\*.*
    Скопировано файлов:         1.
    C:\DEV\SESHAT\ANDREY>cd ..
    C:\DEV\SESHAT>cd boss
    C:\DEV\SESHAT\BOSS>rem Check Article signature
and extract it
    C:\DEV\SESHAT\BOSS>rrec frandrey boss c2.sen-x
    Ok Test Random
    Extract...
    UserFile: frandrey
    Successful UserConainer read!
    Creation UserConainer Time: 14:07:36 21.10.2022
    Network name: 1b4acb08dc2df32f429398babb45
cb58
    Ok OutFile read
    Length: 285680
    File Signature: 6af3181d4119a443
    Calc Signature: 6af3181d4119a443
    RecFile valid!
    Registration User File in: 14:07:44 21.10.2022
    User File Name: c2.txt
    Extract name: 1b4acb08dc2df32f429398babb45
cb58
    Full validation!
    Extracting to c2.txt
    C:\DEV\SESHAT\BOSS>rem Add Article in Reestr
and create notify file
    Следующий фрагмент производит помещение
экстрагированной статьи в распределенный реестр
и выдачу квитанции пользователю при успешной
записи в реестр.
    C:\DEV\SESHAT\BOSS>areestr.exe boss BossPas
c2.txt
    Ok Test Random
    UserContainer: boss
    c2.txt
    User File len = 285649
    Successful UserConainer read!
    Creation UserConainer Time: 14:07:31 21.10.2022
    Ok OutFile write
    Ok OutFileIndex write
    Read Len = 285649
    TNum ok Info ok Imi ok Ok transnum Ok imit
    DNum :00000000000000000000000000000000
    TNum :3de4ffa56c77bffa60dd49a415e6100d
```

Sign :845f8d751efc3207
File :c2.txt
NetName: 429398babb45cb58
AddTime: 14:07:45 21.10.0022
C:\DEV\SESHAT\BOSS>rem Send notify file to andrey
Квитанция посылается пользователю
C:\DEV\SESHAT\BOSS>copy *.kvt ..\andrey*.*
00000000.kvt
Скопировано файлов: 1.

В предлагаемом подходе к работе доверенной платформы мы можем объединить смарт-контракты и чейн-код в том случае, если исполняемые инструкции будут размещены в блокчейне и будут исполняться только после их верификации (проверки подписи или кода аутентификации) и экстракции из звена блокчейна.

ПОНЯТИЕ ЗЕРОКОДИНГА

Приведя пример интерпретируемого файла для разворачивания платформы, мы попадаем в новую парадигму разработки платформ с применением зерокодинга.

Зерокодинг [6] или no-code — это способ создавать работающие ИТ-продукты без использования или с минимальным использованием инструментов программирования, за счет визуального интерфейса программирования и/или готовых платформенных решений.

Большинство задач по реализации платформы, как мы видим, не отличаются большой уникальностью. Шаблоны no-code позволяют не писать код для таких задач с нуля, а пользоваться готовыми блоками, отталкиваясь от задачи — например, если нужно отправить запрос в базу данных или создать пользователя.

Разработчики, принявшие решение взяться за какой-либо IT-проект, оказываются перед следующей проблемой: у нас есть оформленная идея или гипотеза, при этом мы четко представляем себе нужный результат, но между идеей и результатом существует ощутимый разрыв. Чтобы запустить проект, необходимо получить инвестиции, привлечь аудиторию, продемонстрировать первый результат, что требует в первую очередь написания программного кода.

Поэтому складывается такое положение дел, при котором для достижения первых результатов нужно собрать команду программистов, создав для них идейные или материальные стимулы. Разумеется, это не только может быть сложно и рискованно, но и для первой реализации не является необходимым условием. Более того, первая реализация может показать, что идея была обречена изначально, а время, силы и деньги потрачены напрасно.

Зерокодинг быстро и без особых затрат помогает пройти путь от идеи до технического результата за счет встроенных решений, а также позволяет упрощать и решение более сложных задач в части создания мобильных приложений. К примеру, конструктор мобильных приложений Glide создает не полноценные мобильные приложения, а лишь веб-страницы, которые воспроизводят стиль взаимодействия с мобильным приложением, но этого будет вполне достаточно для демонстрации идеи на практике.

Под зерокодингом понимается и способ создания новых решений и продуктов, проверки бизнес-гипотез и автоматизации работы с помощью готовых инструментов, предназначенных для определенных элементов продукта. Готовое решение можно найти практически для любой идеи, например, маркетплейса, клуба по подписке. Такое решение всегда будет быстрее и дешевле разработки.

Для начала работы с этими инструментами, как мы видим, достаточно применить концепцию платформы. Архитектуру любого продукта можно разделить на ряд взаимосвязанных блоков или слоев.

Чаще всего их три:

- база данных;
- бизнес-логика (процессы работы с данными, принятие решений);
- клиентский слой (различные способы взаимодействия с пользователем, включая защищенные, регистрация пользователей и т.д.).

Для каждого из этих блоков появляются свои инструменты зерокодинга, решающие поставленные перед ними конкретные задачи. При этом нужно понимать, что когда мы говорим о создании продукта с помощью зерокодинга — мы говорим об MVP (Minimum Viable Product), то есть о минимально жизнеспособной версии продукта.

Итак, в результате использования зерокодинга создается понятный платформенный продукт в виде легко понимаемой, обсуждаемой и верифицированной блок-схемы, что позволяет получить как можно больше информации как о работе того или иного нововведения, так и о работе продукта в целом. Таким образом, появляется возможность «вырастить» ценность продукта эволюционным методом и в минимально возможные сроки, проходя этап проверки, внедрения и отбраковки неудачных обновлений, а также тестирования логики блоков платформы и продукта в целом.

ЗАКЛЮЧЕНИЕ

Реализация защищенной доверенной платформы обмена документами и их хранения с учетом применения зерокодинга, а также доверенного хранения скриптов и участвующих в них исполняемых модулей в распределенном реестре (блокчейне) позволяет достичь соответствия важнейшим принципам построения платформ, к которым относятся прежде всего максимальное владение исходным кодом платформы, минимальное использование продуктов третьих фирм, компактность кода, возможность его доработки и технической поддержки.

Для блокчейн-платформ выполнение этих свойств наряду с использованием симметричных криптографическх алгоритмов национальной (отечественной) реализации означает также постепенный переход к пятому поколению блокчейна [7], устойчивому к квантовому компьютингу.

СПИСОК ЛИТЕРАТУРЫ

1. Биктимиров М.Р., Щербаков А.Ю. Проблемы синтеза доверенных систем // Труды ИСА РАН. 2012. Т.53. С. 264-271.

2. Рязанова А.А. Цифровые платформы: интегративный потенциал, основные понятия и свойства // Вестник современных цифровых технолоий, № 4, с. 28-39.

3. Алиев Д.Ф., Бородулина С.А., Щербаков А.Ю. Актуальные подходы к наукометрии и квалиметрии // Вестник современных цифровых технолоий, 2022. № 12. С. 11-20.

4. Щербаков А.Ю. Методологические основы и прототип системы семантического искусственного интеллекта // Научно-технический сборник "Научно-техническая информация", сер. 2 Информационные процессы и системы, 2022. № 9. С. 1-6.

5. Разработка и тестирование смарт-контрактов Hyperledger Fabric. URL: https://habr.com/ru/post/426705/ (дата обращения: 12.11.2022)

6. Принципы зерокодинга: разработчики без опыта и приложения за несколько часов. URL: https://hightech.fm/2021/07/21/zerocode (дата обращения: 12.11.2022)

7. Пятое поколение технологии блокчейна. URL: https://cryptor-net.turbopages.org/cryptor.net/s/budushchee-nastupilo/pyatoe-pokolenie-tehnologii-blokcheyna (дата обращения: 12.11.2022)