

Умни договори

проф. д-р инж. Венета Алексиева

ОСНОВНИ МОМЕНТИ

- Определение
- Стъпки на създаване
- Стъпки на внедряване и тестване

Умен договор- определение

- Умните договори са компютърни програми, които съдържат алгоритми, споразумения за бизнес транзакции и други предварително дефинирани условия.
- Тези договори се валидират и изпълняват автоматично, след като са изпълнени посочените договорни условия.
- Терминът „умни договори“ първоначално е въведен, за да се обозначи автоматизацията на правни договори. Той обаче е получил широко признание след приемането си от блокчейн технологията.

Предимства

- Нулево доверие (Trustless):
 - Не е нужно да има доверие на другата страна.
 - Доверието е в кода.
- Скорост и икономия:
 - Процеси, които отнемат седмици и купища хартии, се случват за секунди без такси за посредници.
- Невъзможност за промяна (Immutability):
 - Веднъж записан в блокчейна, договърът не може да бъде изтрит или променен от никого, дори от този, който го е написал.

Среда за изпълнение на интелигентни договори

- Виртуална машина
 - Например виртуалната машина на Ethereum (EVM) е софтуерно-базирана платформа за изпълнение, вградена с хардуерна функционалност, която ѝ позволява да симулира операциите на машина от реалния свят.
- Контейнери (Docker)
 - Контейнерите са инструменти за виртуализация, които осигуряват среда за изолиране на процеси и ресурси, осигурявайки ефективно и сигурно изпълнение на интелигентни договори.

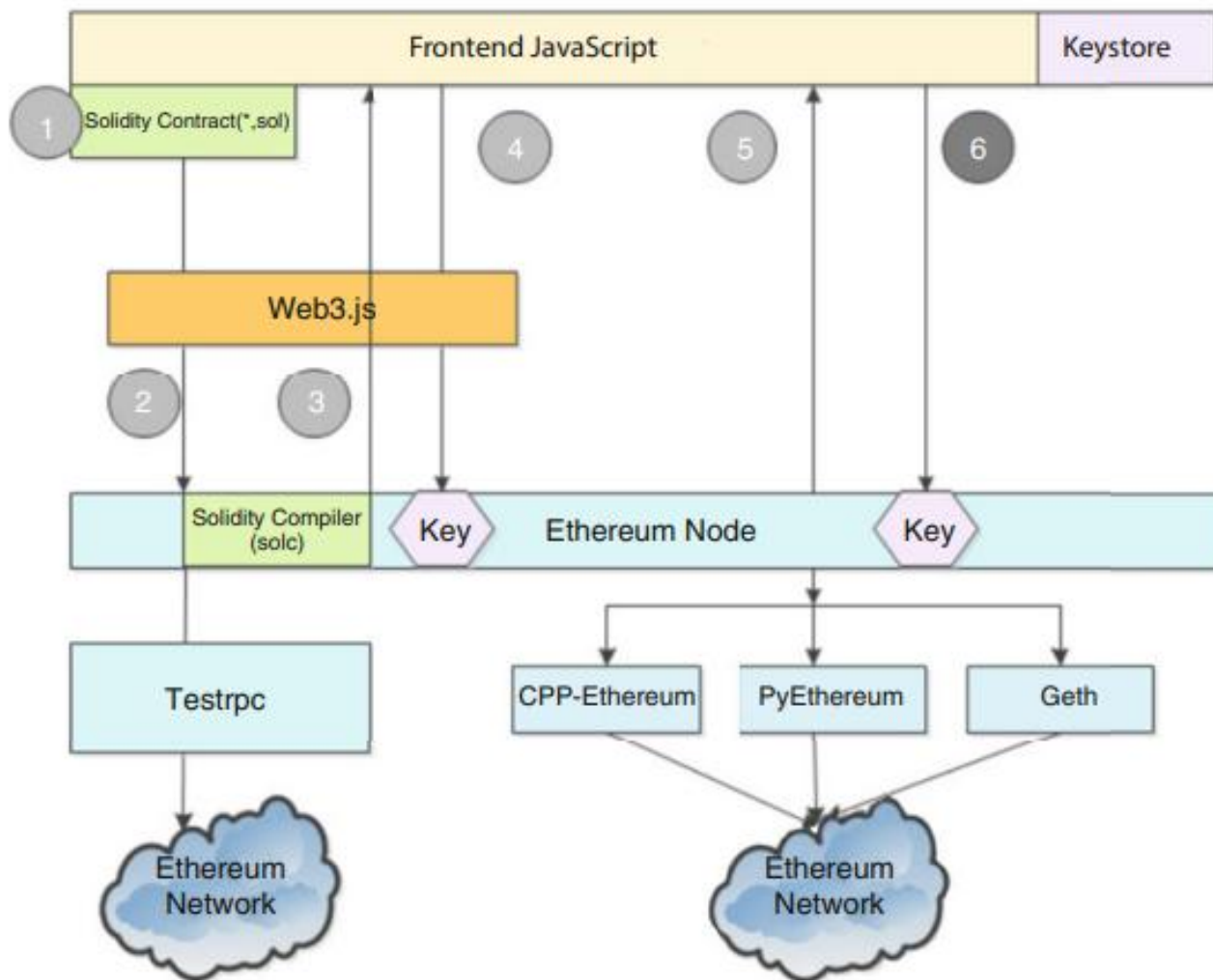
Писане и внедряване на интелигентни договори

- Интегрирана среда за разработка
 - За Ethereum - Remix IDE
 - За HyperLedger - VS Code
- Език за програмиране
 - За Ethereum – Solidity
 - За HyperLedger - Go (Golang), JavaScript / TypeScript (Node.js) или Java
- Дигитален портфейл
 - За Ethereum - MetaMask
 - За HyperLedger – не, но има :
 - Fabric Developer Tools / Microfab: Лек инструмент за бързо стартиране на локална Hyperledger мрежа за целите на тестването, без да е необходимо ръчно да се конфигурират сложни топологии.
 - Fabric SDKs: Библиотеки за комуникация между вашето клиентско приложение и умния договор (наличен за Node.js, Java и Go)

СТЪПКИ

- Написване на интелигентния договор
 - Договорът се пише с помощта на език за програмиране на интелигентни договори.
- Компилиране на интелигентния договор
- Генериране и тестване на байткод
 - Ако договорът премине теста, компилаторът връща bytecode на Frontend.
- Разгръщане в блокчейн
 - Байткодът на интелигентния договор се разгръща в блокчейна.
- Адрес на договора и извличане на ABI (за Ethereum)
 - След успешно разгръщане, блокчейнът връща адреса на договора и двоичния интерфейс на приложението (ABI) (това определя как фронтендът ще взаимодейства с интелигентния договор) към фронтенда.
- Извикване на интелигентен договор
 - Фронтендът взаимодейства с разгръщания договор, като го извиква, използвайки адреса на договора и ABI, предоставени от блокчейна след разгръщането.

Архитектура



За HyperLedger

- 1. Инициализиране на Go проекта
 - Създава се нова папка за проекта и се инициализира Go модула
- 2. Създава се пълен код на умния договор във файл `mycontract.go`:
 - дефиниране на актив,
 - създаване,
 - четене,
 - проверка за съществуване
 - трансфер на собственост.
- 3. Подготовка за компилация
 - Преди да се инсталира кода в Hyperledger Fabric, трябва да се провери, че всичко се компилира правилно на локалната машина

За HyperLedger – как действа

- `contractapi.TransactionContextInterface`
 - Това е контекстът (`ctx`), през който минава всяка транзакция.
 - Той дава достъп до `ctx.GetStub()`, което е интерфейсът за реалното четене и писане в блокчейн базата данни (често CouchDB или LevelDB).
- `ctx.GetStub().PutState(key, value)`
 - Записва двойка ключ-стойност (стойността задължително е под формата на масив от байтове `[]byte`).
- `ctx.GetStub().GetState(key)`
 - Връща байтовете, записани за този ключ.

Внедряване на договора

- Подготовка на средата и стартиране на мрежата
 - Ако мрежата не е стартирана, се пуска заедно с CouchDB (базата данни, която позволява JSON заявки):
 - След това се зареждат променливите на средата за Администратор на Org1
- Пакетиране на умния договор (Package)
 - Трябва да се създаде архив .tar.gz на Go проекта.
 - Задава се реалния път до папката, където се намира asset_contract.go файл
- Инсталиране на договора на Peer възлите (Install)
 - Договорът трябва да се инсталира на възлите на всяка организация, която ще участва в одобрението му.
- Одобряване и дефиниране (Approve & Commit)
 - За да се активира договорът в канала mychannel, организациите трябва да го подобрят с еднакви параметри, след което той се записва окончателно (commit).
- Тестване на умния договор (Invoke / Query)

Основни CLI команди за управление (След внедряване)

- Проверка на инсталираните договори
 - Показва всички договори, инсталирани на конкретния реер възел, заедно с техния Package ID
- Проверка за одобрение (Check Commit Readiness)
 - Показва коя организация е одобрила договора и дали той е готов за активиране в канала
- Списък с активните договори в канала
 - Показва кои умни договори са успешно пуснати в действие в съответния канал
- Обновяване на умния договор (Upgrade / Update)
 - Повтаряне на процеса за новата версия

Въпроси ?

Благодаря за вниманието !