GARR 2018 @ Cagliari – Italy

Applicazioni degli Smart Contracts alla certificazione di filiere di produzione e ai microservizi

Michele Marchesi [a], Roberto Tonelli[a], Andrea Pinna[b], Gavina Baralla[b], Stefano Secci, Simona Ibba[b,]

a) Department of Matehematics and Informatics, University of Cagliari, Italy

b) Department of Electric and Electronic Engineering, University of Cagliari, Italy

Blockchain

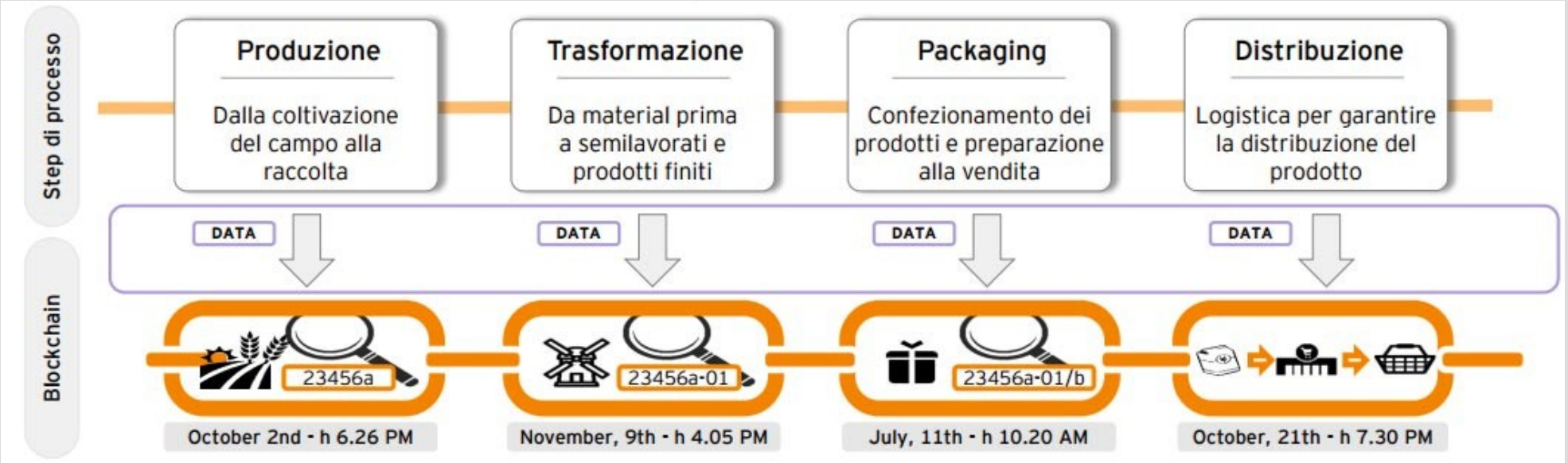
- Catena di Blocchi che fa da registro distribuito (distributed ledger) di transazioni ‘pubbliche’
- Non modificabile, trasparente, distribuita, decentralizzata, su rete peer-to-peer
- Pubblica/Privata, Permissionless/Permissioned
- Solo criptovaluta vs. Smart Conracts
- Nodi identificati da addresses
- Crittografia e hashing

Smart Contracts

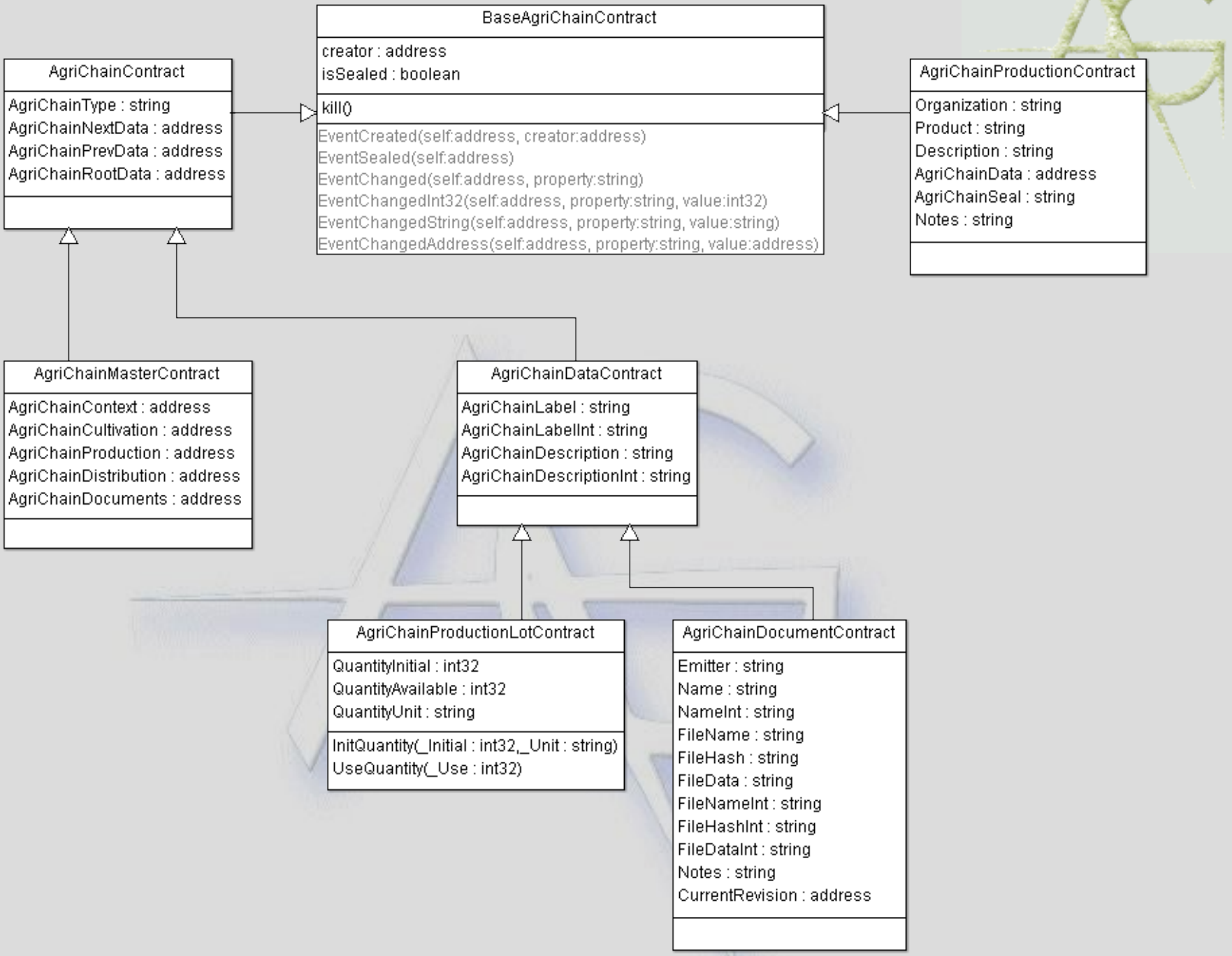
- **Accordi tra parti non ripudiabili ed eseguiti automaticamente**
- **In generale: Codice che viene eseguito su blockchain**
- **Linguaggio di programmazione Solidity per Ethereum**

Certificazione di filiere di produzione

- **Caso di studio** ***Il progetto WINE Blockchain***

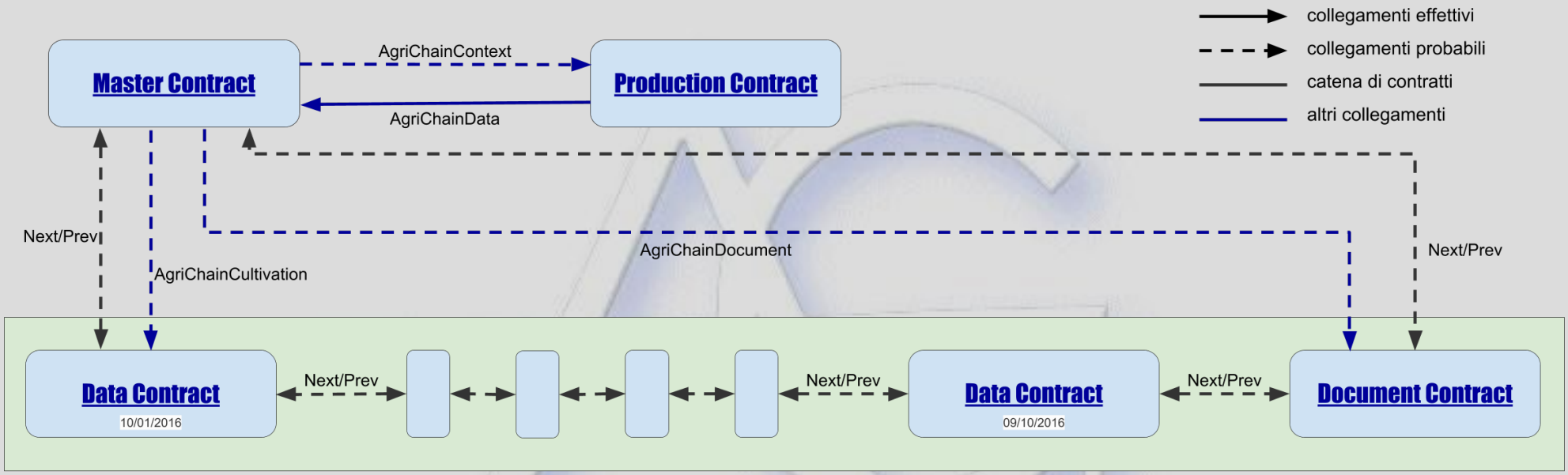


impossibilità di modificare i dati dichiarati + firma digitale del produttore (e di eventuali altri soggetti coinvolti)


BaseAgriChainContract
creator : address
isSealed : boolean
kill()
EventCreated(self:address, creator:address)
EventSealed(self:address)
EventChanged(self:address, property:string)
EventChangedInt32(self:address, property:string, value:int32)
EventChangedString(self:address, property:string, value:string)
EventChangedAddress(self:address, property:string, value:address)
AgriChainContract
AgriChainType : string
AgriChainNextData : address
AgriChainPrevData : address
AgriChainRootData : address
AgriChainProductionContract
Organization : string
Product : string
Description : string
AgriChainData : address
AgriChainSeal : string
Notes : string
AgriChainMasterContract
AgriChainContext : address
AgriChainCultivation : address
AgriChainProduction : address
AgriChainDistribution : address
AgriChainDocuments : address
AgriChainDataContract
AgriChainLabel : string
AgriChainLabelInt : string
AgriChainDescription : string
AgriChainDescriptionInt : string
AgriChainProductionLotContract
QuantityInitial : int32
QuantityAvailable : int32
QuantityUnit : string
InitQuantity(_Initial : int32,_Unit : string)
UseQuantity(_Use : int32)
AgriChainDocumentContract
Emitter : string
Name : string
NameInt : string
FileName : string
FileHash : string
FileData : string
FileNameInt : string
FileHashInt : string
FileDataInt : string
Notes : string
CurrentRevision : address

```
pragma solidity ^0.4.10;

contract BaseAgriChainContract {
    address creator;
    bool public isSealed;

    function BaseAgriChainContract() public    {  creator = msg.sender; EventCreated(this,creator); }

    modifier onlyIfNotSealed() {
        if (isSealed)
            throw;
        _;
    }

    modifier onlyBy(address _account) {
        if (msg.sender != _account)
            throw;
        _;
    }

    function kill() onlyBy(creator)   { suicide(creator); }
    function setCreator(address _creator)  onlyBy(creator)  { creator = _creator;     }
    function setSealed()  onlyBy(creator)  { isSealed = true;  EventSealed(this);   }

    event EventCreated(address self,address creator);
    event EventSealed(address self);
    event EventChanged(address self,string property);
    event EventChangedInt32(address self,string property,int32 value);
    event EventChangedString(address self,string property,string value);
    event EventChangedAddress(address self,string property,address value);
}
```

Nel campo “AgriChainData” caricato sulla blockchain si trova nuovamente l’indirizzo dell’Agri Chain Master Contract da cui si è partiti. Questi sono gli unici contratti che si referenziano a vicenda nella blockchain, quindi la catena non è stata perfettamante implementata e per via dell’errore nel costruttore dell’AgriChainContract, non sono nemmeno stati inizializzati gli indirizzi.


collegamenti effettivi
collegamenti probabili
catena di contratti
altri collegamenti
Master Contract
AgriChainContext
Production Contract
AgriChainData
Next/Prev
AgriChainCultivation
AgriChainDocument
Next/Prev
Data Contract
10/01/2016
Next/Prev
Next/Prev
Data Contract
09/10/2016
Next/Prev
Document Contract

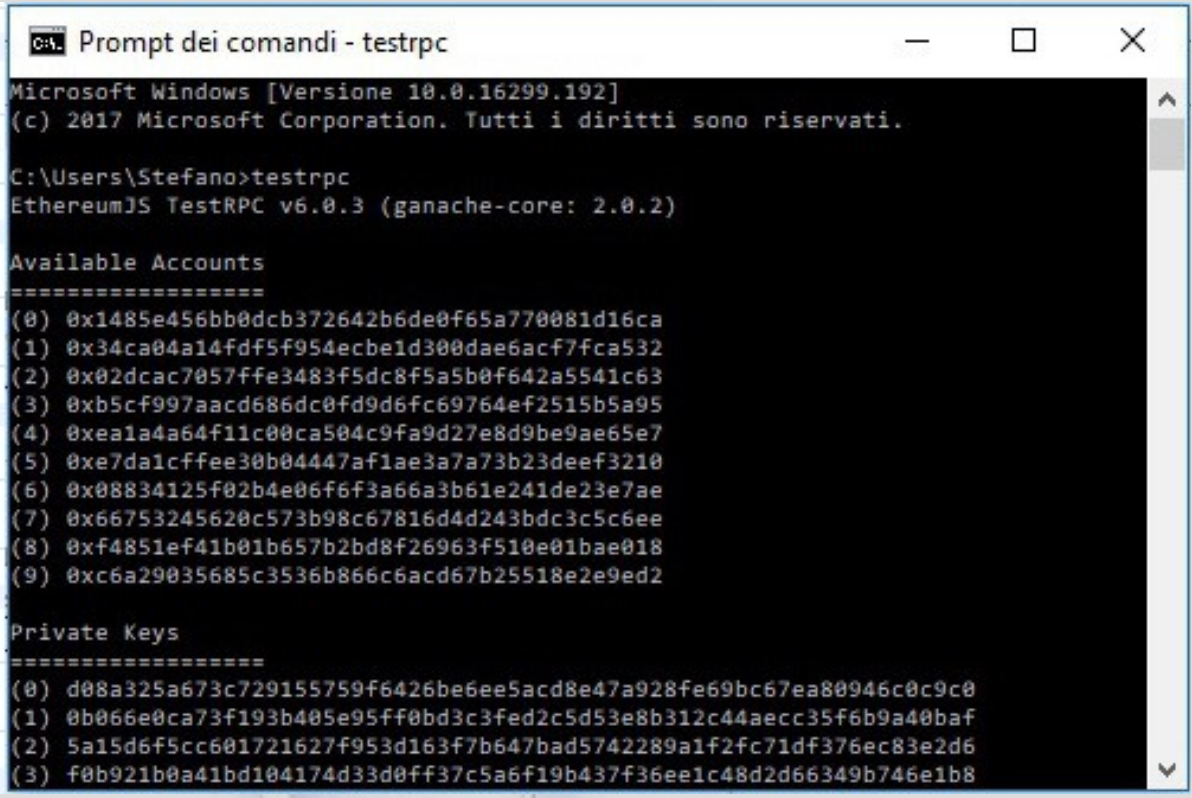
Altre problematiche riguardano alcune funzioni e costrutti utilizzati nei contratti sono stati dichiarati deprecati e perciò da sostituire. Ad es. in BaseAgriChainContract la parola chiave "throw" è oramai deprecata e di conseguenza pure il costrutto if. Per costruire dei modificatori che verifichino delle determinate condizioni vi è la possibilità di scegliere tra tre nuove funzioni che sostanzialmente fanno la stessa cosa del "throw" ma migliorano la leggibilità e hanno diversi effetti sull'utilizzo del gas. Queste funzioni sono assert(), require() e revert(): assert, in caso di fallimento del controllo, brucia tutto il gas rimanente; require, al contrario, è più indulgente; revert, rimborsa il gas rimanente al chiamante e permette di restituire un valore.

Abbiamo scelto di modificare il codice dei due modificatori presenti nel contratto usando assert per il modificatore "OnlyIfNotSealed", perché un tentativo di modificare un valore già sigillato può essere visto come un tentativo di contraffazione e quindi è da disincentivare con la perdita del gas, mentre per la clausola "OnlyBy" si è scelto require. Si è inoltre sostituito "suicide" con "selfdestruct" e corretto l'errore nel nome del costruttore del contratto AgriChainContract.

è stata realizzata una DApp per interagire, lato client, con i contratti del sistema Wine Blockchain, presentato nel capitolo 2 e analizzato nel capitolo 3.

L'applicazione si frappone, quindi, tra il contratto e l'utente creando un'interfaccia pensata per permettere a un ipotetico dipendente della cantina di registrare le varie fasi della filiera nei contratti creati e inseriti nella catena in maniera più semplice. I contratti non possono essere creati dall'applicazione ma devono essere già stati creati precedentemente e vi si può accedere utilizzando gli address degli smart contract. Una volta inseriti i dati e inviati nella blockchain l'applicazione permetterà di visualizzarli e si potrà scegliere se modificarli o "sigillarli" impedendo una futura modifica degli stessi.


AgriChainProductionContract at 0xc15...0a80c (blockchain)
AgriChainMasterContract at 0x397...463fd (blockchain)
AgriChainDataContract at 0xf34...24db8 (blockchain)
AgriChainDataContract at 0xdc8...8247e (blockchain)


Prompt dei comandi - testrpc
Microsoft Windows [Versione 10.0.16299.192]
(c) 2017 Microsoft Corporation. Tutti i diritti sono riservati.
C:\Users\Stefano>testrpc
EthereumJS TestRPC v6.0.3 (ganache-core: 2.0.2)
Available Accounts
==================
(0) 0x1485e456bb0dcb372642b6de0f65a770081d16ca
(1) 0x34ca04a14fdf5f954ecbe1d300dae6acf7fca532
(2) 0x02dcac7057ffe3483f5dc8f5a5b0f642a5541c63
(3) 0xb5cf997aacd686dc0fd9d6fc69764ef2515b5a95
(4) 0xea1a4a64f11c00ca504c9fa9d27e8d9be9ae65e7
(5) 0xe7da1cffee30b04447af1ae3a7a73b23deef3210
(6) 0x08834125f02b4e06f6f3a66a3b61e241de23e7ae
(7) 0x66753245620c573b98c67816d4d243bdc3c5c6ee
(8) 0xf4851ef41b01b657b2bd8f26963f510e01bae018
(9) 0xc6a29035685c3536b866c6acd67b25518e2e9ed2
Private Keys
==================
(0) d08a325a673c729155759f6426be6ee5acd8e47a928fe69bc67ea80946c0c9c0
(1) 0b066e0ca73f193b405e95ff0bd3c3fed2c5d53e8b312c44aecc35f6b9a40baf
(2) 5a15d6f5cc601721627f953d163f7b647bad5742289a1f2fc71df376ec83e2d6
(3) f0b921b0a41bd104174d33d0ff37c5a6f19b437f36ee1c48d2d66349b746e1b8

Production Contract

0xc1504f22da9f8b36abf618383987f681c2a0a80c

address creatore (non pubblico)

note prodotto

organizzazione

Cantina Volpone

nome prodotto

Bianco Falanghina di Puglia IGP

descrizione

Falanghina 2016/2017

data

0x397526bd1508f56d5a1db65b325489c5072463fd

update ProductionContract

- **contratto NON sigillato**
- **organization:**
- **produt:**
- **description:**
- **notes:**
- **AgriChainSeal:**
- **AgriChainData: 0xc1504f22da9f8b36abf618383987f681c2a0a80c**

sigilla contratto

Production Contract

0xc1504f22da9f8b36abf618383987f681c2a0a80c

address creatore (non pubblico)

note prodotto

organizzazione

nome prodotto

descrizione

data

update ProductionContract

- **contratto NON sigillato**
- **organization: Cantina Volpone**
- **produt: Bianco Falanghina di Puglia IGP**
- **description: Falanghina 2016/2017**
- **notes:**
- **AgriChainSeal:**
- **AgriChainData: 0x397526bd1508f56d5a1db65b325489c5072463fd**

sigilla contratto

Microservices (MS)

- Piccole applicazioni sviluppate allo scopo di decomporre un'architettura monolitica
- Sono servizi autonomi, indipendenti, interagenti, istanziati indipendentemente, modulari, girano su server indipendenti
- Erogano servizi specifici (ad utenti o a client)
- Gli SC rispecchiano tale paradigma

Smart Contracts (SC)

- Smart Contracts rappresentano programmi, sono istanziati su blockchain, girano sui nodi in maniera indipendente ma coerente, sono attivati da transazioni
- SC tipicamente implementano task semplici e autonomi con scopo definito
- SC mettono a disposizione un'interfaccia (API) per I contratti chiamanti

Model

- Usiamo un'architettura blockchain-oriented a microservizi basati su Smart Contracts per un caso di studio: e-commerce application
- L'architettura ha due layers:
  - Il primo e' l'interfaccia tra applicazioi e blockchain
  - Il secondo e' composto da un set di SC che girano su blockchain

Primo Layer

- Fornisce l'ABI (The Ethereum Application Binary Interface)
- Un'applicazione software puo' usare le ABI dello SC per richiedere servizi
- L'ABI del Contratto specifica le funzioni che possono essere chiamate e garantisce il format dei valori di ritorno

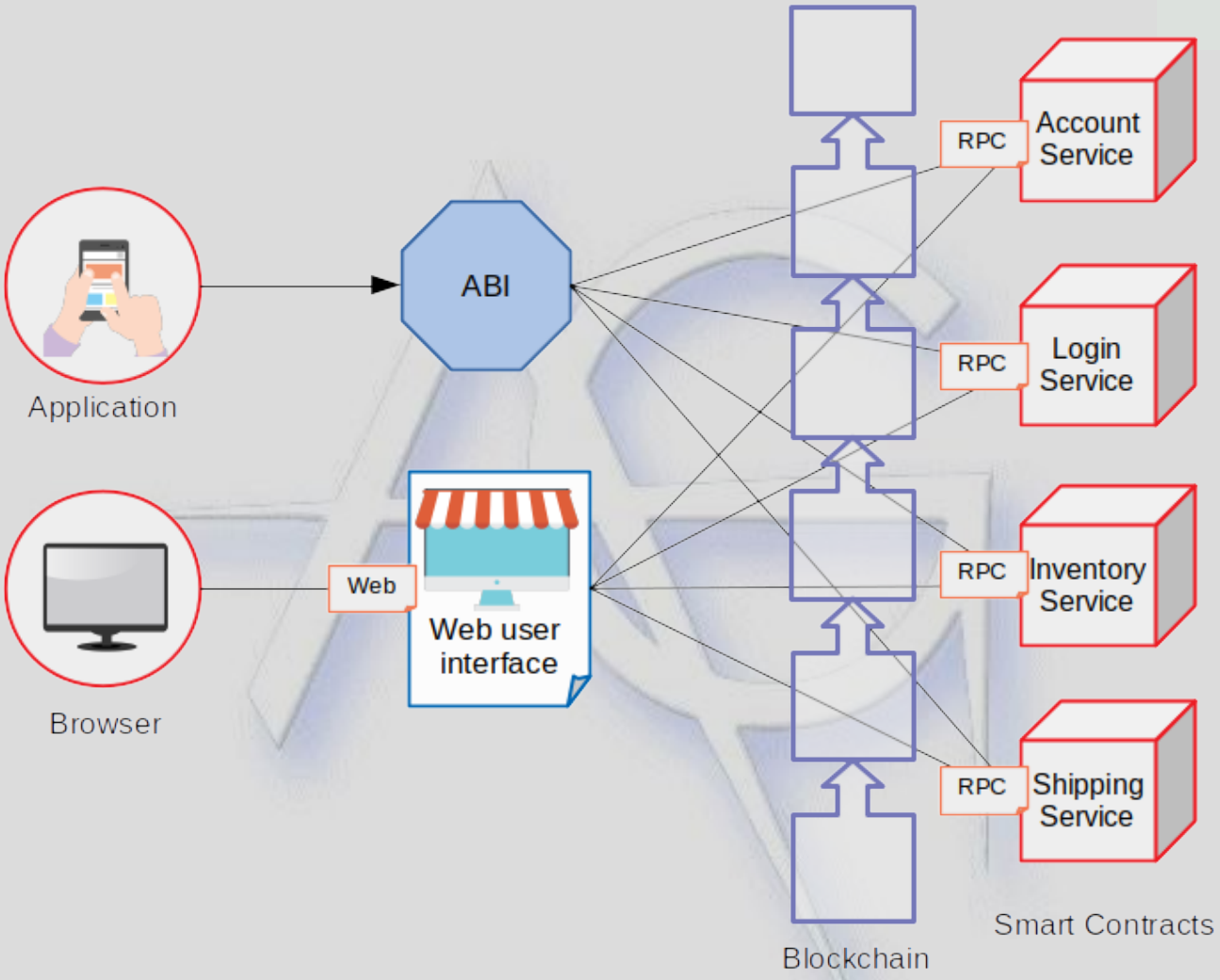
Secondo Layer

- Ogni microservizio e' implementato da uno smart contract atomico.
- La communication tra layers takes avviene con remote procedure calls (RPC), tramite la Web3 Ethereum library
- La libreria e' usata per scrivere progammi javascript per creare ed eseguire blockchain transactions e chaimate agli SC (service requests)

Functioning

- Each user is uniquely identified by an Ethereum address.
- The Account service records and manages users/clients information.
- Depending on the client profile, the system enables different functionalities or services.
- Once registered, data are stored permanently within the blockchain and all node have a copy of the blockchain

Model Architecture



- Il Data access e’ sicuro e trasparente
- Users registration e login sono microservizi gestiti da uno SC dedicato
- Il servizio inventory registra I dati su blockchain e restituisce l’informazione allapagina web dello store online