How smart contracts can implement the policy objective of ‘report once’

COST 2018
Focus Session
marc.sel@pwc.com

6 September 2018
Winterthur
Contents

Introduction – objectives of the presentation
Overview of the project, regulation and demonstrator
Distributed ledgers in 1 minute
The smart contract demonstrator
  • Functionality
  • Set-up
  • Use cases
Demonstrator constraints
Further references
Introduction – objectives of the presentation

The presentation discusses a demonstrator built for the European Commission’s DG FISMA and show how smart contracts can implement 'digital doppelgängers' of OTC derivatives contracts that are traded to help achieve the policy objective of ‘report once’.

EC DG FISMA

The Directorate-General for Financial Stability, Financial Services and Capital Markets Union is the Commission department responsible for EU policy on banking and finance.

OTC derivatives

Derivatives are financial instruments whose value depends on (‘derives from’) the values of other, more basic, underlying variables e.g. a stock of a company, or an index such as S&P. Over-the-counter (OTC) derivatives are customized, bilateral agreements that transfer the risk from one party to the other (such as swaps and forwards).

Smart contracts as ‘digital doppelgängers’

Smart contracts are executable pieces of software residing on an electronic distributed ledger. In the presented demonstrator, a private Ethereum ledger was used. The smart contracts are a representation of OTC derivatives.
Overview of the project

Objectives of the project

Explore possibilities to address the problems of cost of regulatory burdens for banks and financial institutions and inadequate monitoring of risk attached to financial contracts.

Study investigated the feasibility of Distributed Ledger Technologies (DLTs) as the new way of financial institutions to meet their reporting obligations as laid down in EU financial sector legislation.

Novelties

- resulting demonstrator uses smart contracts to meet the policy objective "report once" in financial reporting (EMIR, MIFIR, COREP) in the EU
- Use of ACTUS semantics
Overview of the regulations

**EMIR** (European Market Infrastructure Regulation) provides that some classes of OTC derivative transactions have to be cleared through Central Counterparties (CCPs), and that risk mitigation techniques have be be be applied for other OTC transactions.

**MiFIR** extends the clearing obligation by CCPs to regulated markets for exchange-traded derivatives.

**COREP**, issued by EBA, specifies Guidelines for Common Reporting:
- for Capital Requirements Directive (CRD) reporting
- covers credit risk, market risk, operational risk, own funds and capital adequacy ratio
- when trading each counterparty conducts its own reporting to its own supervisor
Overview of the demonstrator

• The demonstrator shows
  • How digital representations of Over The Counter (OTC) derivatives such as bonds or swaps can be implemented as smart contracts on a blockchain. The life-cycle of the derivatives during trading follows the real assets’ life-cycle in semi real-time.
  • How blockchain-based reports can assemble and calculate the information required for compliance reporting (COREP, EMIR, MIFIR) towards supervisory bodies
• Today, this reporting information is typically dispersed and difficult to obtain and correctly aggregate. However, in the demonstrator, as all the is present on-chain, these reports contain all required information (‘report once’)
• The demonstrator is a vehicle to demonstrate Distributed Ledger Technology (DLT) concepts, it is not an implementation of a target solution
**Distributed ledgers in one minute**

Vocabulary
- Users (Alice, Bob, Romeo, …)
- Applications (cryptocurrency wallet, digital doppelgänger, …)
- Nodes (alternatively called ‘blockchain nodes’)
- Blockchain (concatenation of blocks)
- ‘Consensus’ (all nodes agree on which blocks are part of the blockchain)

Single node: Alice

Set of nodes:

All nodes have an identical copy of the blockchain
What is stored in the blockchain is considered immutable
**Distributed ledgers in one minute**

Users perform transactions through the application
Every node broadcasts its transaction outputs
Every node can create its candidate block

Which candidate block gets selected?
In most blockchain implementations, nodes try to solve a cryptopuzzle (mathematically hard problem based on hashing), and the first node that solves the puzzle has its candidate block accepted by the other nodes.
Once new block is part of the chain, it is considered as immutable, and the transactions as committed on a ledger.

**Consensus is important**
Ensures that the next block in a blockchain is the one and only version of the truth.
Keeps powerful adversaries from derailing the system and successfully forking the chain.
There are many different ways to obtain consensus.
The smart contract demonstrator

Functionality of the demonstrator
Contracting parties (Alice, Bob, Eve)
• trade OTC derivatives through their usual channels
• have a user interface at their disposal to create digital doppelgängers on the distributed ledger

These digital doppelgängers are replicated to all nodes on the ledger according the consensus protocol

Every node, including the regulator (Romeo) has access to the same information in semi-real time

Every node can produce the compliance reports
Set-up of the demonstrator (conceptual)

Two types of nodes

- Raspberry PI (Ethereum node, not mining)
- Laptop (Ethereum node, mining)
**Set-up of the demonstrator**

**Specific aspects:**
Client requested to demonstrate cheap 25 € computers (Raspberry Pi’s) could also run a node

However, these:
- Do not have a screen to run a user interface
- Do not have sufficient computing power to mine

As a consequence:
Standard off-the-shelf laptops were added to run the UI and do the mining

Furthermore, a dedicated user interface was developed to visualise the consensus
Set-up of the demonstrator

Trade of OTC derivative

Alice

Bob

Private Peer-to-Peer network (Ethereum)

Blockchain state

Blockchain state

Blockchain state

Blockchain state

ConsensusUI

ConsensusUI

ConsensusUI

ConsensusUI

ConsensusUI

AliceUI

BobUI

NarratorUI

RegulatorUI

Eve

The Narrator
**OTC Derivatives as smart contracts**

A ‘Smart Contract’ is essentially a piece of software. It forms a contract capable of automatically enforcing itself, without a third party between individual participants. It is created and executed on the distributed ledger.

It contains the **contract logic**, the rules of what should happen at a certain point:

And it contains the **contract state**, in our case meaning interest rate, duration of the contract, nominal value, etc.

For this demonstrator we implemented two contracts:
- Bond (based on Actus PAM contract)
- Interest Rate Swap or IRS (based on Actus ‘Plain Vanilla‘ PVSWAP contract)
# Bond

<table>
<thead>
<tr>
<th>What</th>
<th>Implemented as</th>
</tr>
</thead>
<tbody>
<tr>
<td>A bond is a <strong>debt investment</strong></td>
<td>A Smart Contract type that can be created on the blockchain, each contract has its ContractID (CID)</td>
</tr>
<tr>
<td>in which an investor loans money to the issuer which borrows the funds,</td>
<td>RecordCreator (LEIRC) and Counterparty (LEICP), specifying a currency (CUR) and NotionalPrincipal (NT)</td>
</tr>
<tr>
<td>for a defined <strong>period of time</strong>,</td>
<td>MaturityDate (MD)</td>
</tr>
<tr>
<td>at <strong>variable or fixed</strong> interest rate,</td>
<td>e.g. National Nominal Interest Rate (IPNR)</td>
</tr>
<tr>
<td>paid at <strong>specific dates</strong>,</td>
<td>Cycle anchor date of interest payment (IPANX) and Cycle of interest payment (IPCL)</td>
</tr>
<tr>
<td>that <strong>can be transferred</strong> (sold) to another investor.</td>
<td>LEICP can be updated</td>
</tr>
</tbody>
</table>
pragma solidity^0.4.9;
contract PAM{
    enum currencies { EUR, USD, GBP, JPY }
    enum cycles {Y, H, Q, M, W, D}

    //Actus contract attributes
    string public CID; // ContractID
    string public LEIRC; // LegalEntityIDRecordCreator
    string public LEICP; // LegalEntityIDCounterparty
    uint128 public CDD; // ContractDealDate = unixtimestamp
    uint128 public IED; // InitialExchangeDate = unixtimestamp
    uint128 public MD; // MaturityDate = unixtimestamp
    uint128[2] NT; // NotionalPrincipal
    currencies public CUR; // Currency
    uint128[2] IPNR; // National Nominal Interest Rate
    string public IPDC; // Day Count Convention
    uint public IPANX; //cycle anchor date of interest payment = unixtimestamp
    cycles public IPCL; // Cycle of interest payment
    uint128[2] IPCBA; // InterestCalculationBaseAmount
    uint256 public SD; // Status Date = unixtimestamp

    //Actus state variables
    uint128[2] Nvl; // Nominal Value
    uint128[2] Nrt; // Nominal Rate
    uint public Led; // Last event Date = unixtimestamp

    //Operational variables
    bool public initialpaymentdone;
    int public interestpayments;
    bool public defaulted;
    bool public active;
    uint128 periodCounter; // the total number of periods that have passed

    function initialise(string _CID, string _LEIRC, // idem for _LEICP, _IED, _MD, _NT, etc
    {
        periodCounter = 0;
        active = true;
        CID = _CID;
        LEIRC = _LEIRC;
        LEICP = _LEICP;
        IED = _IED; // initial payment day
        MD = _MD; // end date of the contract
        NT = _NT;
        CUR = currencies(_CUR);
        IPNR = _IPNR;
        IPDC = "30/360";
        IPANX = _IPANX;
        IPCL = cycles(_IPCL);
        IPCBA = _IPCBA;
        SD = now;

        Nvl = _NT;
        Nrt = _Nrt;
        Led = SD;
    }

    function nextPeriod() constant returns (bool){
        Led = SD;
        SD = now;
        periodCounter += 1;
    }

    function setDefault(){
        defaulted = true;
    }
# Swap

<table>
<thead>
<tr>
<th><strong>What</strong></th>
<th><strong>Implemented as</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>A swap is a transaction between two parties</td>
<td>A Smart Contract type that can be created on the blockchain, each contract has its ContractID (CID). Parties are the RecordCreator (LEIRC) and Counterparty (LEICP),</td>
</tr>
<tr>
<td>These two parties agree to exchange cash flows in the future</td>
<td>Agreeing on the value for the swap, coded in nominalValue. Agreeing on fixed parameters, coded in: fixedRate (interest rate) and fixedPayout (value), Agreeing on variable parameters, coded in variableRate (interest rate) and variablePayout (value),</td>
</tr>
<tr>
<td>They agree about the specific dates when the cash flows are to be paid</td>
<td>There is a start date SD, and a maturity date MD. There is a constant defined that specifies the periods per year, PPY = 4, and a counter maintaining the contract’s period,</td>
</tr>
<tr>
<td>that <strong>can be transferred</strong> (sold) to another investor.</td>
<td>LEICP can be updated</td>
</tr>
</tbody>
</table>
pragma solidity^0.4.9;
contract PAM{
    enum currencies { EUR, USD, GBP, JPY }
    enum cycles {Y, H, Q, M, W, D}

    //Actus contract attributes
    //Actus state variables
    //Operational variables

    // Getter, setter and helper functions ....

    function nextPeriod() returns (int128){
        periodCounter += 1;
        fixedPayout = [[[nominalValue[0]]*fixedRate/100] / 4,nominalValue[1]];  
        variablePayout = [[[nominalValue[0]]*variableRate/100]/4,nominalValue[1]]; 
        payoutValue = [[[fixedPayout[0]]/fixedPayout[1]] - (variablePayout[0] / variablePayout[1]),1]; 
    }
}

.......
Set-up of the demonstrator

Actors:

- **Alice**, **Bob** and **Eve** are contracting parties, using their respective *AliceUI* and *BobUI* (UI: UserInterface)
- **Romeo**, party with the reporting application, using the *RegulatorUI*
- **The Narrator**, tells the stories, uses the *NarratorUI*
  - Story 1: Bond, where Alice lends money to Bob and later merges with Eve to form AliceEve
  - Story 2: Interest Rate Swap, where Bob creates contract towards AliceEve and creates a new ‘PlainVanillaSwap’ (PVSWAP) contract with her
  - Story 3: AliceEve defaults on the PVSWAP contract
  - Showing the contents of the DLT, generating reports (EMIR, MIFIR, COREP) and demonstrating consensus takes place during the stories
Demonstrator Story 1: Bond

• In this story, Alice and Bob conclude a bond contract.
• On the blockchain a digital doppelgänger of this contract is created, and through consensus this appears in each node.
• The Narrator simulates that time is advanced to the next quarter.
• Romeo generates a regulatory report, a COREP report:
  - All data available on-chain according ACTUS semantics
  - Hence all calculations can be performed in real-time
  - And report is available immediately
New bond contract

Contract ID: Contract 1

Owner: Alice

Counterparty: Bob

Initial exchange date: 09/22/2017

Maturity date: 09/22/2020

Notional principal: 34500 EUR

National nominal interest rate (%): 1

Interest cycles start date: 09/22/2017

Interest cycles: Monthly
2 Contract visualisation

2 Consensus visualisation (consensus not yet achieved)
3 Consensus visualisation (consensus achieved)
### 4 Contract visualisation

Current interest rate: 2%

<table>
<thead>
<tr>
<th>Type</th>
<th>Contract ID</th>
<th>Owner</th>
<th>Counterparty</th>
<th>Initial exchange date</th>
<th>Maturity date</th>
<th>Nominal value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAM</td>
<td>Contract 1</td>
<td>Alice</td>
<td>Bob</td>
<td>22-9-2017</td>
<td>22-9-2020</td>
<td>34500</td>
</tr>
</tbody>
</table>

Current quarter: Q2 2017

- Next quarter
### Credit Risk: Credit and counterparty credit risks and free deliveries: standardised approach to capital requirements

<table>
<thead>
<tr>
<th>Number</th>
<th>Field name</th>
<th>Details to be reported</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Exposure class</td>
<td>Indicates the specific class in which</td>
<td>Exposure to Institution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the exposure will be classified</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>according to Article 112 of CRR</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>SME</td>
<td>Indicates if the exposure value is an</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>exposure towards a Small or Medium-sized</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enterprise, this field indicates if the</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SME is subject to the SME supporting</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>factor according to the requirements of</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Article 561 of the CRR</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Exposure type</td>
<td>Indicates the exposure type</td>
<td>Derivatives and long settlement transactions</td>
</tr>
<tr>
<td>6</td>
<td>On-balance sheet or off-balance sheet exposure</td>
<td>Indicates if the exposure appears on the company's balance sheet or not</td>
<td>On-balance sheet</td>
</tr>
<tr>
<td>7</td>
<td>Centrally cleared?</td>
<td>Indicates if the derivative is centrally</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cleared through a Qualifying Central</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Counterparty</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Securities Financing</td>
<td>Indicates if the transaction is a</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Transactions</td>
<td>Securities Financing Transaction (SFT),</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>as defined in paragraph 17 of the Basel</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Committee document The Application of</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Basel II to Trading Activities and the</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Treatment of Double Default Effects,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>includes: (i) Repurchase and reverse</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>repurchase agreements defined in Article</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 (80) of CRR as well as securities or</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>commodities lending and borrowing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>transactions, (ii) margin lending</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>transactions as defined in Article 272</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) of CRR</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Contractually Cross Product</td>
<td>Indicates that due to the existence of a</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Netting?</td>
<td>non-contractual cross product netting (as</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>defined in Article 272 (11) of CRR</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>nth default credit derivative?</td>
<td>Indicates if the exposure related to a</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>derivative where the nth default among</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>the exposures shall trigger payment</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Value adjustments and</td>
<td>The exposure value without taking into</td>
<td>1,180.72</td>
</tr>
<tr>
<td></td>
<td>provisions associated with</td>
<td>account value adjustments and provisions,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the original exposure</td>
<td>conversion factors and the effect of</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>credit risk mitigation techniques.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Exposure net of value</td>
<td>Value adjustments and provisions for</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>adjustments</td>
<td>credit losses made in accordance with the</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>accounting framework to which the</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>reporting entity is subject to.</td>
<td></td>
</tr>
</tbody>
</table>

Net exposure which is the difference between (8) and (9).

1,180.72
Demonstrator constraints

In the current demonstrator implementation:

- In the current set-up of the demonstrator, for simplification reasons, contracting information is reflected in the DLT, and all participants have access to it (may not be desirable for a production system)
- The contracting which takes place between parties is supposed to take place outside the blockchain universe (via phone or via another application)
- Only these contracts are then created on the blockchain
- The identities of the parties is not cryptographically confirmed (production system would do this e.g. with electronic authentication)
- The identity and state of the parties (e.g. legal entity definition, identity attributes, defaulted or not) is not maintained on the blockchain
- The authenticity of the contract itself is not cryptographically confirmed (production system would do this e.g. with electronic signatures)
- Valuation is done within the reporting application, taking into account the data on the blockchain as well as external data (currently hardcoded)
Further references

Open access publication describing the demonstrator in detail at https://zenodo.org/record/884497#.W4zoTLhLeUn

A movie of the demonstrator in action is available at https://www.pwc.be/fismablockchain

ACTUS: http://actusfrf.org

Also:

http://www.pwc.be/blockchain
http://www.marcesl.eu