

Harnessing blockchain technology to enhance trust and traceability in wine trading among wineries

Rubén Hortelano-Haro, José A. Mateo-Cortés, María Blanca Caminero-Herráez, Enrique Arias-Antúnez, Ángel Hernández-Bravo, Diego Dujovne

Annex I. Validation test of smart contracts

The primary aim of this work is to develop a decentralized tool that aids overseers in dealing with the requirements of transparency, security, and traceability in wine stock exchanges. The proposed solution has been implemented by using smart contracts, thereby leveraging the benefits of blockchain technology to achieve these requirements.

In this annex the execution of the transactions from the developed smart contracts is described one by one, checking that the transaction flow is working as intended.

The software used for the development and testing of this work is IBM Blockchain Platform¹. In this software, each transaction is defined by two types of operations, as shown in Figure 1.

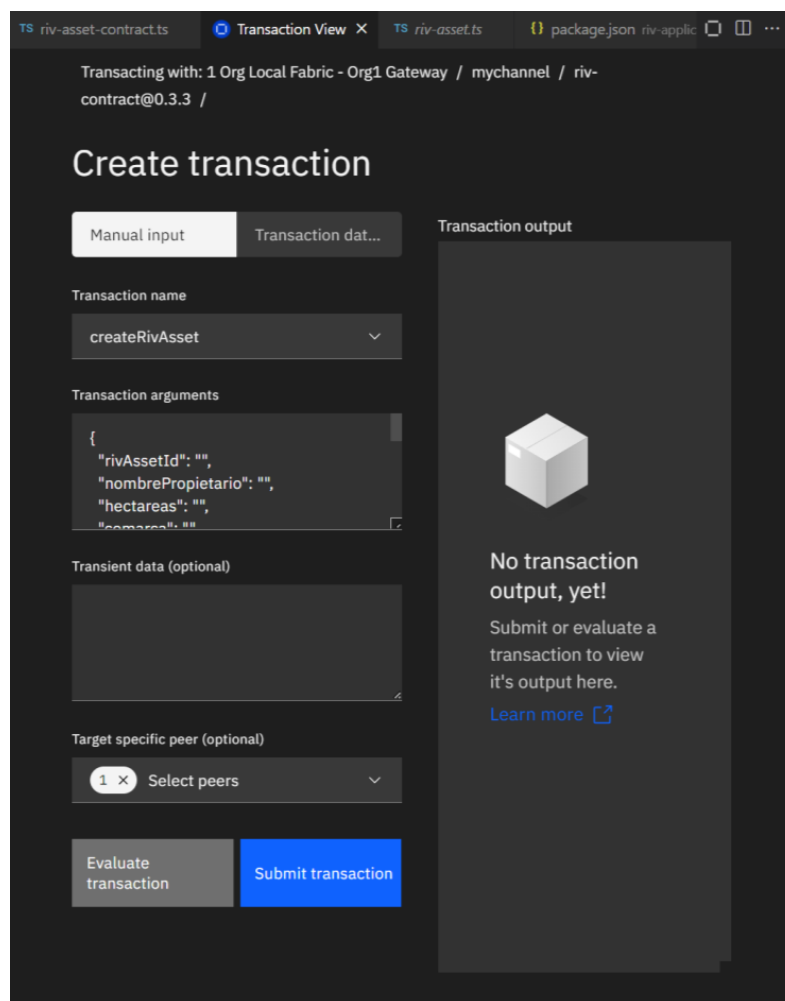


Figure 1: View of the riv-contract trade execution environment

Figure 1 shows the interface that allows us to carry out the execution of the transactions of the contract defined. To this end, we can select the transaction we want to execute. Subsequently, we observe that the next section allows us to enter the arguments, in which there is an already semi-complete JSON format. We must complete it with the data we wish to provide as a parameter. The use of this format also helps us to check that what the transaction will receive complies with this pre-execution format.

¹ <https://cloud.ibm.com/docs/blockchain?topic=blockchain-get-started-ibp>

The next point shows how we can perform the execution of the transaction in two ways. This is because if we only want to execute the transaction, but not write to the general ledger, a *transaction evaluation operation* is carried out. If, on the other hand, we want to record the transaction, we perform a *submit operation*, which writes to the general ledger and records the transaction. Finally, we can see that we have a transaction output section. Here we see the output of the transaction as feedback and, if there is a problem, an error message is shown to the user. An example of an error message is shown in Figure 2.

```
Transaction output

Error evaluating transaction: Query failed. Errors: ["Peer org1peer-api.127-0-0-1.nip.io:8080: GET_STATE failed: transaction ID: 9cac99e9e8e952ea1fc4b251b5cabe68df6ba899bc0237ea079e0d59e70f0e7f: invalid key. Empty string is not supported as a key by couchdb"]
```

Figure 2: Console output with transaction failure error

To test and verify the contract, an execution of the transactions has been carried out according to the flow described in the manuscript, verifying that it performs correctly and that we cannot skip the intermediate transactions. Note that in the following snapshots the name of the attributes and methods are shown in Spanish, their translated version is included in the manuscript.

Following the procedure explained with the example in Figure 1, the creation of the deposit is done. Thus, the submit operation has been executed with the data that appear, on the lefthand side of Figure 3. Once the transaction has been completed, we see on the output the record that has been created (centre), while on the righthand side we find that the transaction has been recorded in the ledger.

Transaction arguments	Transaction output	Blockchain
<pre>{ "rivAssetId": "B01", "nombrePropietario": "Pablo Orostivar", "hectareas": "15", "comarca": "Comarca Sur", "numeroCatastro": "35", "calleDomicilioPropietario": "C/ Inventada 24", "localidad": "Madrid", "codigoPostal": "28001", "provincia": "Madrid", "anoCampana": "2022", "fechaEntrega": "10/11/2022", "tipoUva": "Tempranillo", "pesoNeto": "1200", "grado": "1", "calidad": "1", "bodega": "Bodegas Ronaldo" }</pre>	<pre>Returned value from createRivAsset: {"nombrePropietario": "Pablo Orostivar", "hectareas": 15, "comarca": "Comarca Sur", "numeroCatastro": 35, "calleDomicilioPropietario": "C/ Inventada 24", "localidad": "Madrid", "codigoPostal": 28001, "provincia": "Madrid", "anoCampana": 2022, "fechaEntrega": "10/11/2022", "tipoUva": "Tempranillo", "grado": 1, "calidad": 1, "bodega": "Bodegas Ronaldo", "pesoNeto": 1200, "estado": "ALMACENADO", "estadoUva": "MOSTO"}</pre>	<pre>[29/6/2022 1:22:42] [INFO] submitTransaction [29/6/2022 1:22:42] [INFO] submitting transaction createRivAsset with args B01, Pablo Orostivar, 15, Comarca Sur, 35, C/ Inventada 24, Madrid, 28001, Madrid, 2022, 10/11/2022, Tempranillo, 1200, 1, 1, Bodegas Ronaldo on channel mychannel to peers org1peer-api.127-0-0-1.nip.io:8080 [29/6/2022 1:22:42] [SUCCESS] Returned value from createRivAsset: {"nombrePropietario": "Pablo Orostivar", "hectareas": 15, "comarca": "Comarca Sur", "numeroCatastro": 35, "calleDomicilioPropietario": "C/ Inventada 24", "localidad": "Madrid", "codigoPostal": 28001, "provincia": "Madrid", "anoCampana": 2022, "fechaEntrega": "10/11/2022", "tipoUva": "Tempranillo", "grado": 1, "calidad": 1, "bodega": "Bodegas Ronaldo", "pesoNeto": 1200, "estado": "ALMACENADO", "estadoUva": "MOSTO"}</pre>

Figure 3: Execution of valid deposit creation transaction (createRivAsset)

If we follow the wine transformation cycle, we arrive at the corresponding fermentation transaction. However, we must previously locate the tank, so if we try to carry out the fermentation transaction, we get a restriction error, as shown in Figure 4. We therefore proceed to carry out the deposit as indicated in Figure 5.

Once we have the deposit in place, we can perform the fermentation transaction. However, given that fermentation is a transaction where the weight decreases, if this operation is carried out with a higher weight than the existing one, it will generate an error as shown in Figure 6 ($\text{pesoFermentacion} \geq \text{rivAssetId}$, $1300 \geq 801$). The correct execution, with a weight of 800 kg ($\text{pesoFermentacion} < \text{rivAssetId}$, $800 < 801$), is shown in Figure 7.

Transaction arguments	Transaction output	Blockchain
<pre>{ "rivAssetId": "B01", "pesoFermentacion": "800" }</pre>	<p>Error evaluating transaction: Query failed. Errors: ["Peer org1peer-api.127-0-0-1.nip.io:8080: Primero debe depositar el mosto del asset en un deposito"]</p>	<pre>[29/6/2022 1:59:07] [INFO] evaluating transaction fermentationRivAsset with args B01,800 on channel mychannel to peers org1peer-api.127-0-0-1.nip. io:8080 [29/6/2022 1:59:07] [ERROR] Error evaluating transaction: Query failed. Errors: ["Peer org1peer-api.127-0-0-1.nip. io:8080: Primero debe depositar el mosto del asset en un deposito"]</pre>

Figure 4: Execution of wrong fermentation operation (fermentationRivAsset)

Transaction arguments	Transaction output	Blockchain
<pre>{ "rivAssetId": "B01", "depositoAlmacenaje": "1º Piso Sector 3 Barril 2" }</pre>	<p>No value returned from placeMoveRivAsset</p>	<pre>[29/6/2022 2:12:14] [INFO] submitTransaction [29/6/2022 2:12:14] [INFO] submitting transaction placeMoveRivAsset with args B01, 1º Piso Sector 3 Barril 2 on channel mychannel to peers org1peer-api.127-0-0-1.nip.io:8080 [29/6/2022 2:12:15] [SUCCESS] No value returned from placeMoveRivAsset</pre>

Figure 5: Execution of the correct place in deposit operation (placeMoveRivAsset)

Transaction arguments	Transaction output	Blockchain
<pre>{ "rivAssetId": "B01", "pesoFermentacion": "1300" }</pre>	<p>Error evaluating transaction: Query failed. Errors: ["Peer org1peer-api.127-0-0-1.nip.io:8080: Error en este proceso se pierden kilos no se guardan "]</p>	<pre>[29/6/2022 2:02:43] [INFO] evaluateTransaction [29/6/2022 2:02:43] [INFO] evaluating transaction fermentationRivAsset with args B01,1300 on channel mychannel to peers org1peer-api.127-0-0-1.nip. io:8080 [29/6/2022 2:02:43] [ERROR] Error evaluating transaction: Query failed. Errors: ["Peer org1peer-api.127-0-0-1.nip. io:8080: Error en este proceso se pierden kilos no se guardan "]</pre>

Figure 6: Wrong execution of fermentation operation (fermentationRivAsset)

Transaction arguments	Transaction output	Blockchain
<pre>{ "rivAssetId": "B01", "pesoFermentacion": "800" }</pre>	<p>No value returned from fermentationRivAsset</p>	<pre>[29/6/2022 2:24:28] [INFO] submitting transaction fermentationRivAsset with args B01,800 on channel mychannel to peers org1peer-api.127-0-0-1.nip. io:8080 [29/6/2022 2:24:28] [SUCCESS] No value returned from fermentationRivAsset</pre>

Figure 7: Execution of the correct fermentation operation (fermentationRivAsset)

Now, we want to **sell part of a wine deposit to another winery**. We must first separate what is to be sold. For this we will use the division transaction (divideRivAsset), in which we will indicate the quantity to be separated for a deposit (seleccionKilos that should be lower or equal to pesoFermentacion). If we execute the transaction with a higher weight (seleccionKilos = 801 > pesoFermentacion = 800), a restriction will prevent the transaction from being recorded in the ledger, as shown in Figure 8. The correct flow is shown in Figure 9 where seleccionKilos = 500 fulfills the constraints.

Transaction arguments	Transaction output	Blockchain
<pre>{ "rivAssetId": "B01", "seleccionKilos": "801" }</pre>	<p>Error evaluating transaction: Query failed. Errors: ["Peer org1peer-api.127-0-0-1.nip.io:8080: La división no puede llevarse a cabo en el rivAsset con B01 dado que la cantidad supera la existente"]</p>	<pre>[29/6/2022 2:36:36] [INFO] evaluateTransaction [29/6/2022 2:36:36] [INFO] evaluating transaction divideRivAsset with args B01,801 on channel mychannel to peers org1peer-api.127-0-0-1.nip.io:8080 [29/6/2022 2:36:36] [ERROR] Error evaluating transaction: Query failed. Errors: ["Peer org1peer-api.127-0-0-1.nip. io:8080: La división no puede llevarse a cabo en el rivAsset con B01 dado que la cantidad supera la existente"]</pre>

Figure 8: Execution of unsuccessful division load operation (divideRivAsset)

Transaction arguments	Transaction output	Blockchain
<pre>{ "rivAssetId": "B01", "seleccionKilos": "500" }</pre>	<p>No value returned from divideRivAsset</p>	<pre>[29/6/2022 2:47:14] [INFO] submitTransaction [29/6/2022 2:47:14] [INFO] submitting transaction divideRivAsset with args B01,500 on channel mychannel to peers org1peer-api.127-0-0-1.nip.io:8080 [29/6/2022 2:47:14] [SUCCESS] No value returned from divideRivAsset</pre>

Figure 9: Execution of successful division load operation (divideRivAsset)

Because of the division, two new records have been generated, which correspond to the Id of the parent record concatenated with /H1/ and /H2/. To check that both the initial record and the records generated by the previous operation have been generated, we will use the read operation, verifying that the generated records have been created correctly (Figure 10) and that the parent has been logically deleted (Figure 11).


```
[29/6/2022 2:56:34] [INFO] evaluateTransaction
[29/6/2022 2:56:34] [INFO] evaluating transaction readRivAsset with args B01/H1/ on
channel mychannel to peers org1peer-api.127-0-0-1.nip.io:8080
[29/6/2022 2:56:34] [SUCCESS] Returned value from readRivAsset: {"anoCampana":2022,
"bodega":"Bodegas Ronaldo","calidad":1,"calleDomicilioPropietario":"C/ Inventada 24",
"codigoPostal":28001,"comarca":"Comarca Sur","estado":"ALMACENADO","estadoUva":"VINO",
"fechaEntrega":"10/11/2022","grado":1,"hectareas":15,"localidad":"Madrid",
"nombrePropietario":"Pablo Orostivar","numeroCatastro":35,"pesoNeto":500,
"provincia":"Madrid","tipoUva":"Tempranillo"}
[29/6/2022 2:56:54] [INFO] evaluateTransaction
[29/6/2022 2:56:54] [INFO] evaluating transaction readRivAsset with args B01/H2/ on
channel mychannel to peers org1peer-api.127-0-0-1.nip.io:8080
[29/6/2022 2:56:54] [SUCCESS] Returned value from readRivAsset: {"anoCampana":2022,
"bodega":"Bodegas Ronaldo","calidad":1,"calleDomicilioPropietario":"C/ Inventada 24",
"codigoPostal":28001,"comarca":"Comarca Sur","estado":"ALMACENADO","estadoUva":"VINO",
"fechaEntrega":"10/11/2022","grado":1,"hectareas":15,"localidad":"Madrid",
"nombrePropietario":"Pablo Orostivar","numeroCatastro":35,"pesoNeto":300,
"provincia":"Madrid","tipoUva":"Tempranillo"}
```

Figure 10: Read operations on the assets resulting from divideRivAsset

```
[29/6/2022 2:53:09] [INFO] evaluateTransaction
[29/6/2022 2:53:09] [INFO] evaluating transaction readRivAsset with args B01 on channel
mychannel to peers org1peer-api.127-0-0-1.nip.io:8080
[29/6/2022 2:53:09] [SUCCESS] Returned value from readRivAsset: {"anoCampana":2022,
"bodega":"Bodegas Ronaldo","calidad":1,"calleDomicilioPropietario":"C/ Inventada 24",
"codigoPostal":28001,"comarca":"Comarca Sur","estado":"ALMACENADO","estadoUva":"VINO",
"fechaEntrega":"10/11/2022","grado":1,"hectareas":15,"localidad":"Madrid",
"nombrePropietario":"Pablo Orostivar","numeroCatastro":35,"pesoNeto":800,
"provincia":"Madrid","tipoUva":"Tempranillo","depositoAlmacenaje":null,
"fechaBaja":"00:47:14 GMT+0000 (Coordinated Universal Time)"}
```

Figure 11: Asset verification after deregistration

Once the records have been separated and relocated, we proceed to the sale of a child record, in particular the one corresponding to B0/H1/. To do this, the sell transaction (soldRivAsset) will be carried out, as shown in Figure 12.

Transaction arguments	Transaction output	Blockchain
<pre>{ "rivAssetId": "B01/H1/", "bodega": "BODEGAS INIESTA" }</pre>	No value returned from soldRivAsset	<p>Transaction View</p> <pre>[29/6/2022 3:07:27] [INFO] submitTransaction [29/6/2022 3:07:27] [INFO] submitting transaction soldRivAsset with args B01/H1/,BODEGAS INIESTA on channel mychannel to peers org1peer-api.127-0-0-1.nip.io:8080 [29/6/2022 3:07:27] [SUCCESS] No value returned from soldRivAsset org1peer-api.127-0-0-1.nip. io:8080: La división no puede llevarse a cabo en el rivAsset con B01 dado que la cantidad supera la existente"]</pre>

Figure 12: Execution of the sale transaction to another winery (soldRivAsset)

On the other hand, with the other record generated (B0/H2/), we are going to perform the bottling transaction (bottleRivAsset), which ends by deleting the record through a logical deletion and ends the production flow (Figure 13).

Transaction arguments	Transaction output	Blockchain
<pre>{ "rivAssetId": "B01/H2/" }</pre>	No value returned from bottleRivAsset	<pre>[29/6/2022 3:16:55] [INFO] submitting transaction bottleRivAsset with args B01/H2/ on channel mychannel to peers org1peer-api.127-0-0-1.nip.io:8080 [29/6/2022 3:16:56] [SUCCESS] No value returned from bottleRivAsset</pre>

Figure 13: Execution of the bottling operation (bottleRivAsset)

With the result of these tests, we can see that each of the executed transactions comply with the predefined requirements, in addition to verifying the expected results. Thus, this validates the functionality of our designed contract.