

---

# \*-Chain

## Modeling Carne PRI supply chain with the \*-Chain Platform

STEFANO BISTARELLI  
FRANCESCO FALOCI ( \* francesco.faloci@unicam.it)  
MARINO MICULAN  
PAOLO MORI  
CARLO TATICCHI



# Modeling Carne PRI supply chain with the \*-Chain Platform

## Contents

1. The \*-Chain Framework (now with update too!)
2. CarnePRI – Nomenclature (use case)
3. Representing the CarnePRI Supply Chain
4. Generated Solidity Code
5. Conclusion and Future Works

---

# The \*-Chain Framework

## Main Goal

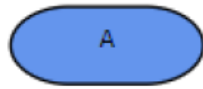
- Representing the Supply chain management system use case of **CarnePRI** nomenclature (European definition)

## Components

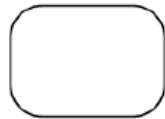
- Domain-Specific Graphical Language
- Graphic design interface
- Interfaces builder (new layout)

# The \*-Chain Framework

## Domain-Specific Graphical Language

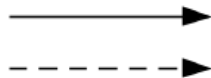


**Asset:** any "item" or object that could be subject of the analysis



Package

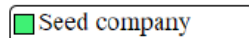
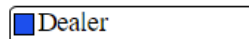
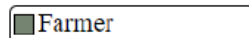
**Container:** an object that somehow contains the asset under exam.



**Operations:** each relationship that binds two objects under exam.

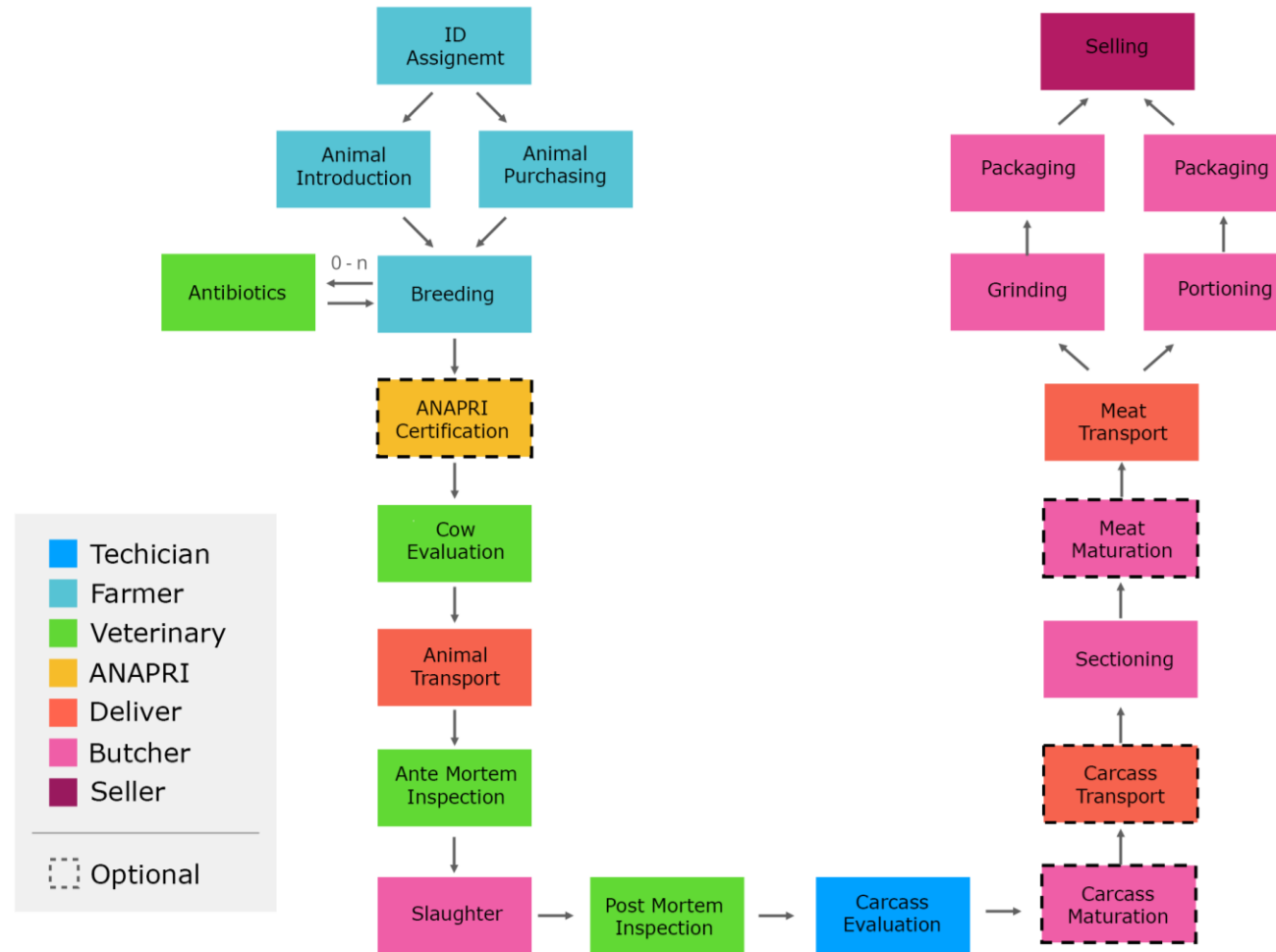


**Properties:** defining traits of the asset

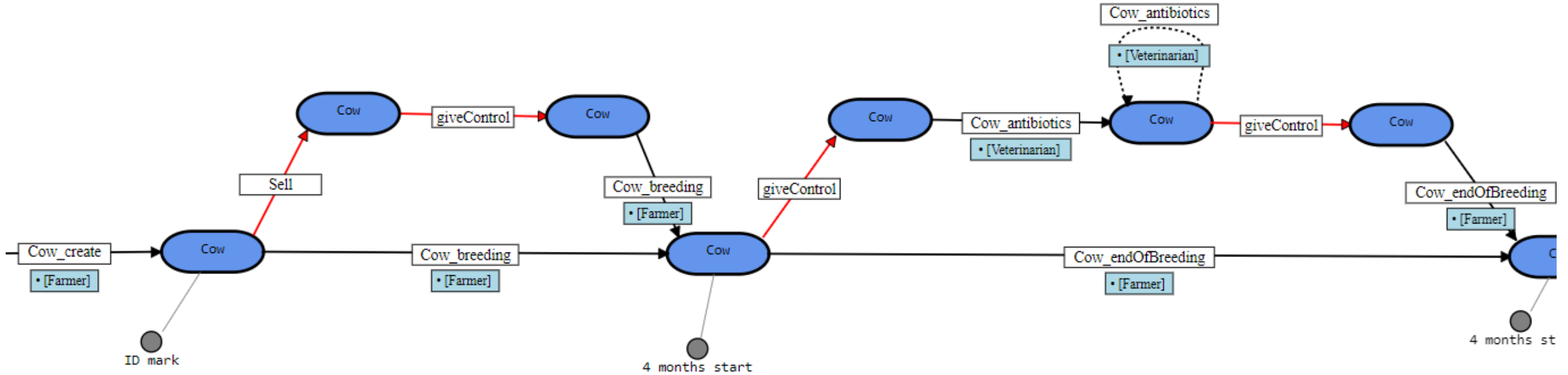


**Roles:** defining subjects who can perform a specific action.

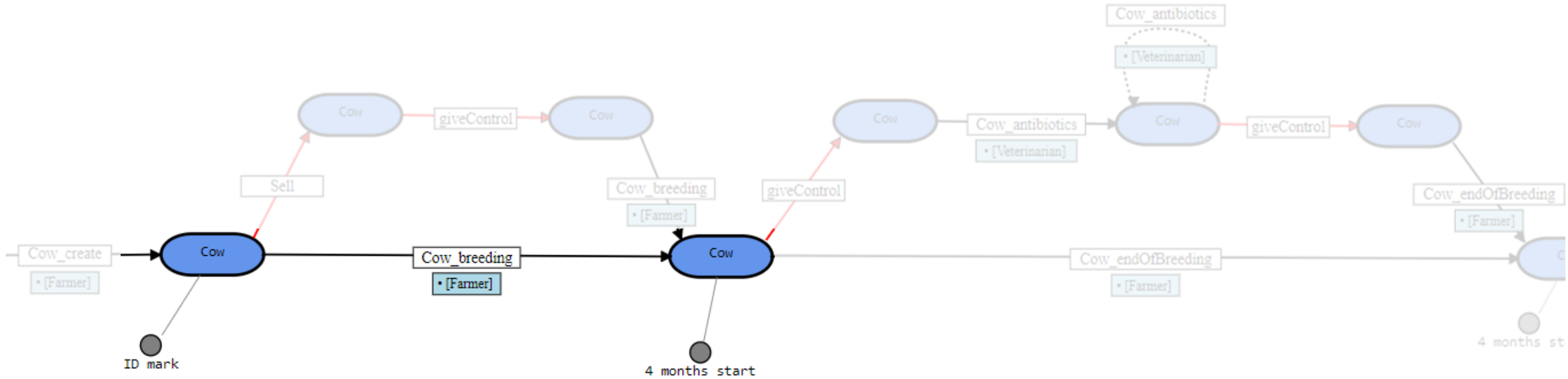
# CarnePRI supply chain



# CarnePRI supply chain: schema

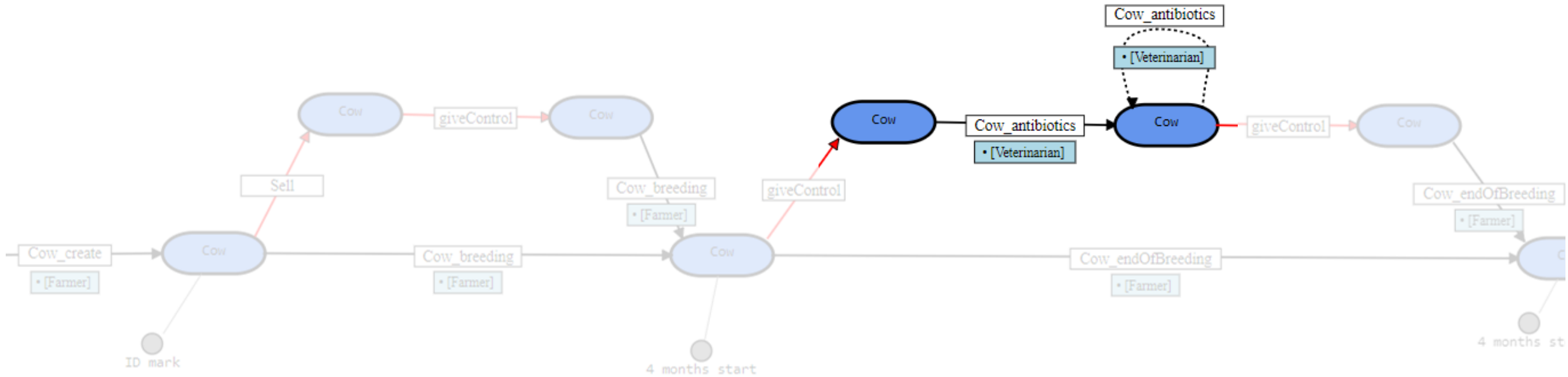


# CarnePRI supply chain



**Update** operation renews the content of properties of an asset

# CarnePRI supply chain



**Monitor** operation check and store specific or all the properties of an asset



# CarnePRI supply chain

```
1 // SPDX-License-Identifier: GPL-3.0
2 pragma solidity >= 0.8.0;
3 import "@openzeppelin/contracts/access/AccessControl.sol";
4 import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
5
6 contract contract_Cow is ERC20, AccessControl {
7     bytes32 public constant Farmer = keccak256("Farmer");
8     bytes32 public constant Veterinarian = keccak256("Veterinarian");
9     bytes32 public constant ANAPRI = keccak256("ANAPRI");
10    bytes32 public constant Deliver = keccak256("Deliver");
11    bytes32 public constant Butcher = keccak256("Butcher");
12
13    enum asset_states {Initialized, Sell_ed, Cow_breeding_ed, giveControl_ed, Cow_ant
14        Cow_ANAPRICertify_ed, Cow_Evaluation_ed, Cow_evaluation_ed, Cow_load_ed, Cow_
15
16    struct asset_Cow_history { //properties
17        string position;
18        string ID_mark;
19        string _4_months_start;
20        string _4_months_stop;
21        string transport_suitability;
22        asset_states state_of_Cow; //actual state
23    }
24
25    struct asset_Cow_struct{
26        asset_Cow_history[] Cow;
27        uint32 ID;
28    }
29    asset_Cow_struct[] public store_Cow_s; // MAIN STORAGE
30
31    function Sell(uint_ID, string memory position) public {
32        uint len = (store_Cow_s[ID].Cow.length)-1;
33        require(store_Cow_s[ID].Cow[len].state_of_Cow == asset_states.Initialized);
34        // update the state array
35        asset_Cow_history memory temp = store_Cow_s[ID].Cow[len];
36        temp.state_of_Cow = asset_states.Sell_ed;
37
38        // other code here ...
39        store_Cow_s[ID].Cow.push(temp);
40    }
41
42    function Cow_breeding(uint_ID, string memory position) public {
```

| Asset     | Deployment Cost | USD   |
|-----------|-----------------|-------|
| Cow       | 6026082         | 11.29 |
| Carcass   | 4959541         | 9.29  |
| Meat      | 3710175         | 6.95  |
| Mincemeat | 2609557         | 4.89  |
| Steak     | 2609407         | 4.89  |
| Truck     | 2618557         | 4.90  |
| Package   | 2609544         | 4.89  |
| erc20     | 1246301         | 2.33  |
| Total     | 26389314        | 49.43 |

The total cost for a single product is maximum **749100** (gas)

# CarnePRI supply chain

0x2C25c493B6d2Fc3B56f7e    \*\*\*\*\*    Login

**0x2C25c493B6d2Fc3B56f7e9D1095934aA91838c42 [Farmer]**

Product name    Create product

COWE57  
COWE58  
COWE59  
COWE60

```
sequenceDiagram
    actor Farmer as [Farmer]
    participant C1 as Cow
    participant C2 as Cow
    participant C3 as Cow
    participant C4 as Cow
    actor Vet as [Veterinarian]

    Farmer->>C1: Cow_create
    C1->>C2: Sell
    C1->>C3: Cow_breeding
    Farmer->>C3: Cow_breeding
    C3->>C4: giveControl
    C4->>Vet: Cow_antibiotics
    C3->>C4: giveControl
```

**Cow**

- Function Sell()
- Function Cow\_breeding()
- Function giveControl()
- Function Cow\_breeding()

# CarnePRI supply chain

- Analyze and translate **other real case** supply chains
- Implement **Macros** of operations
- **Extend** framework tools to other smart contract languages
- **Compare** our framework to other competitors

---

# THANKS FOR THE ATTENTION

/end

STEFANO BISTARELLI  
FRANCESCO FALOCI ( \* [francesco.faloci@unicam.it](mailto:francesco.faloci@unicam.it))  
MARINO MICULAN  
PAOLO MORI  
CARLO TATICCHI

