

Bologna – 22 January 202



# Hands-on tool interoperability using the CERBERO Interoperability framework

IBM-TNO-ABINSULA

Michiel van den Baar, [michiел.vandenbaar@tno.nl](mailto:michiел.vandenbaar@tno.nl)

Dr.Rer.Nat Julio de Oliveira Filho, [julio.deoliveirafilho@tno.nl](mailto:julio.deoliveirafilho@tno.nl)

Evgeny Shindin, [evgensh@il.ibm.com](mailto:evgensh@il.ibm.com)

Giuseppe Meloni, [giuseppe.meloni@abinsula.com](mailto:giuseppe.meloni@abinsula.com)



Horizon 2020  
European Union funding  
for Research & Innovation

# Agenda



Show the value of using CIF to the embedded systems community

Getting started	Now
CIF concepts and applicability	Now – 12:00
Hands-on tutorial	12:00 – 12:45
Feedback & questions	12:45 – 13:00
LUNCH!	

***CLF eases the process of creating interoperability between tools by defining the equivalent aspects between models***

Bologna – 22 January 2020



# CERBERO Interoperability Framework

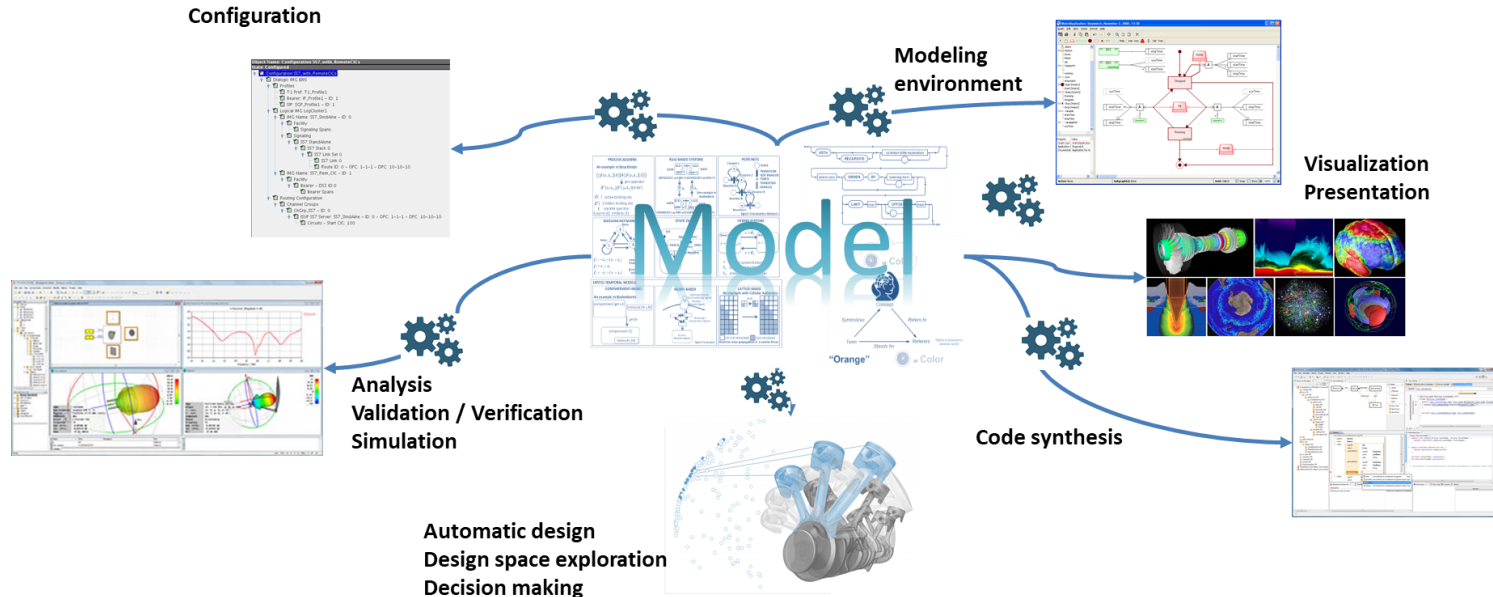
IBM-TNO-ABINSULA Contributions



Horizon 2020  
European Union funding  
for Research & Innovation

# Interoperability in model based design

In model based engineering, interoperability primarily means efficient sharing of models' information.

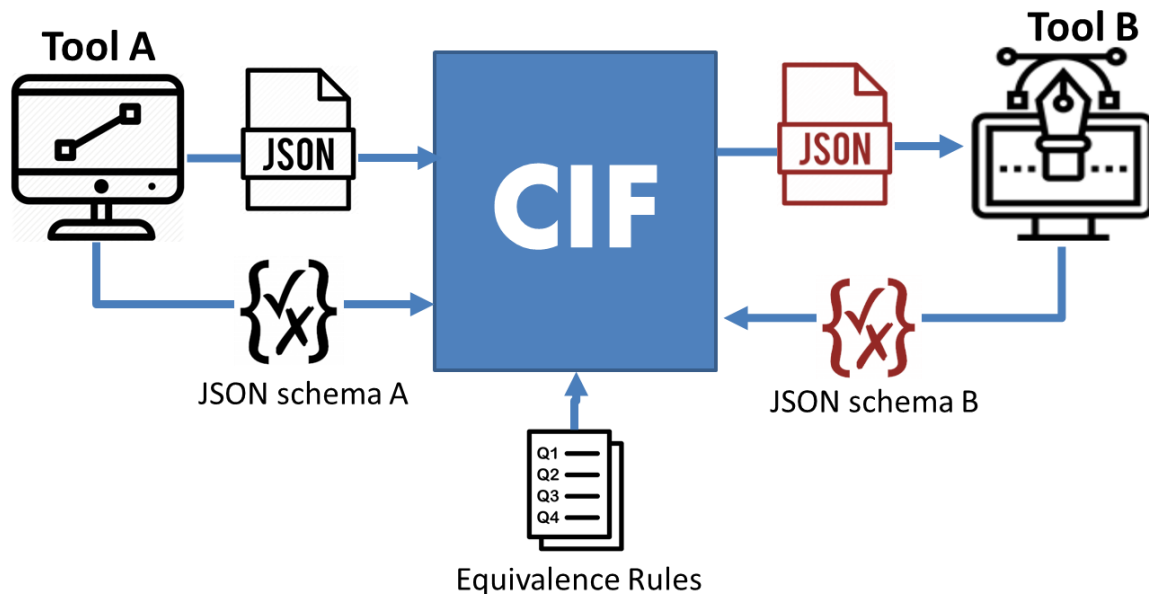


**The magic**

# Basics of CIF

*Emancipating Instances from the Tyranny of Classes in Information Modeling, Jeffrey Parsons, Yair Wand, ACM Transactions on Database Systems, Vol. 25, No. 2, June 2000*

**CERBERO provides a framework for seamless exchange of model information between (design) tools**



You write in your format, I read in mine !

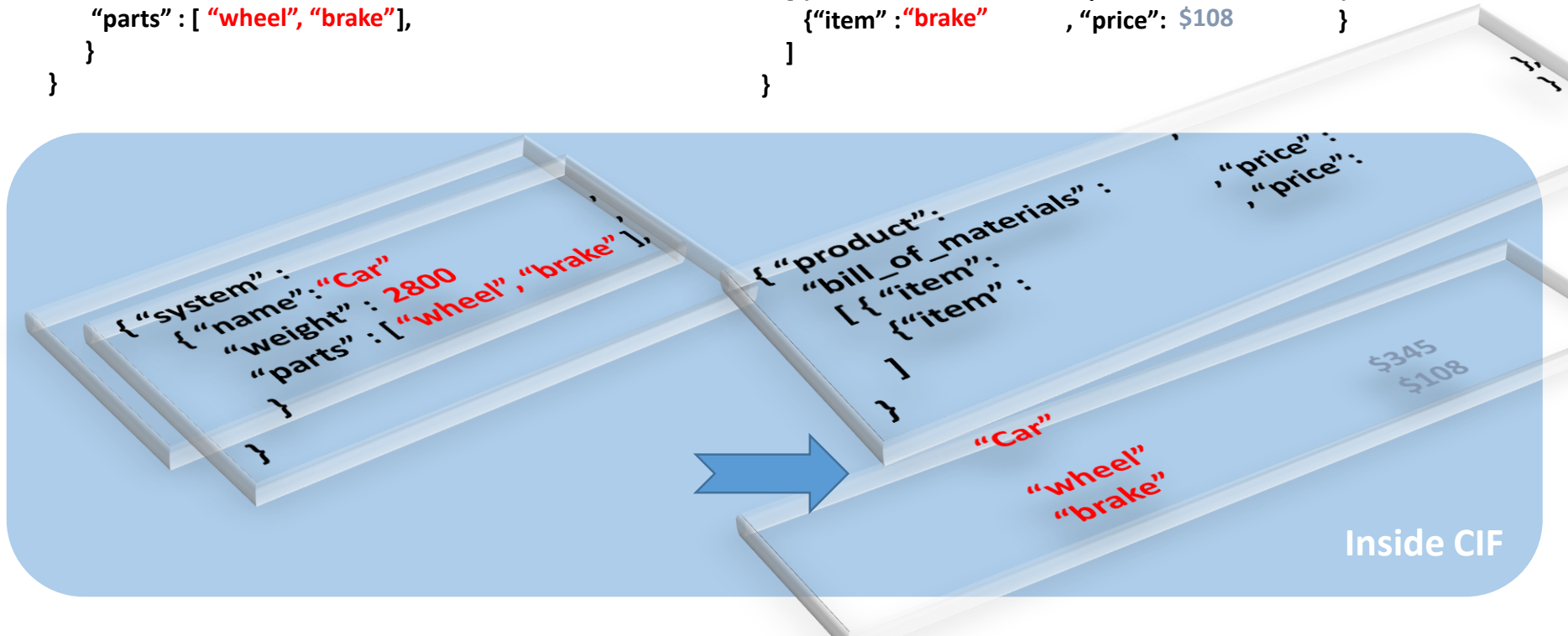
# Basics of CIF

## JSON – tool A

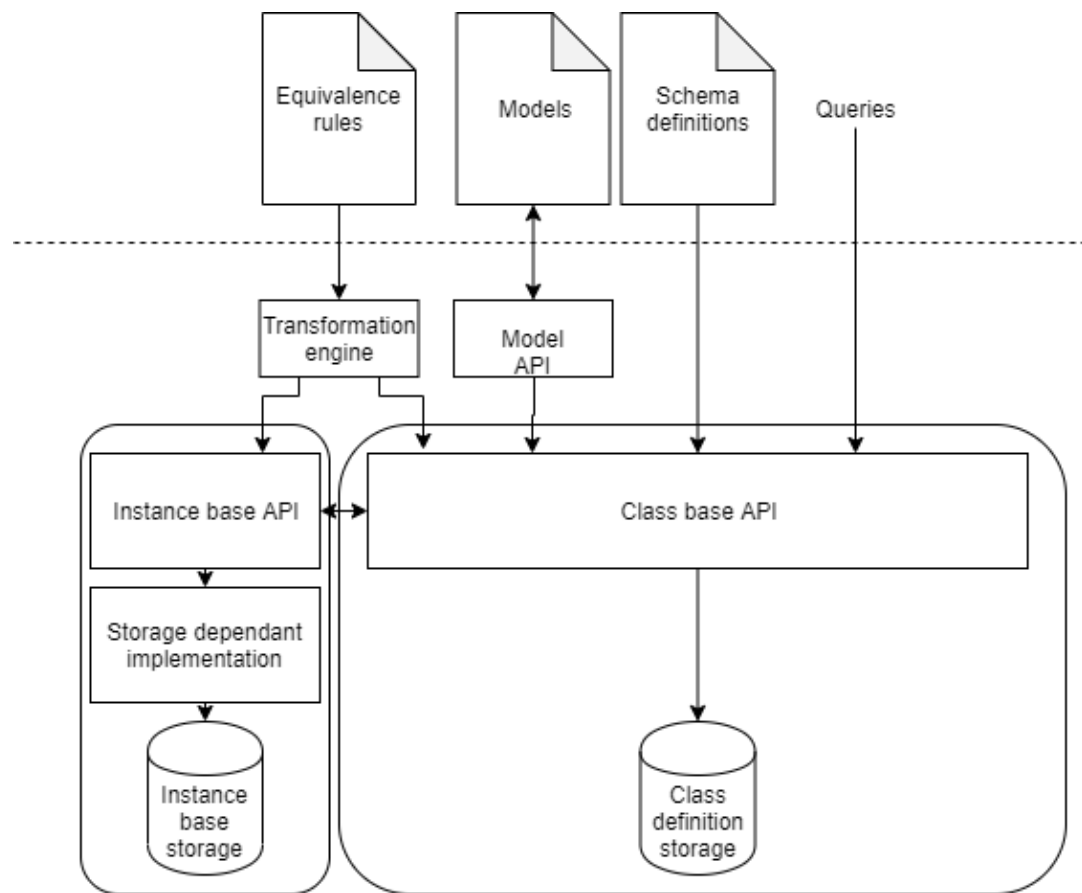
```
{ "system":  
  { "name": "Car",  
    "weight": 2800,  
    "parts": [ "wheel", "brake"],  
  }  
}
```

## JSON – tool B

```
{ "product": "Car",  
  "bill_of_materials":  
    [ { "item": "wheel", "price": $345 },  
      { "item": "brake", "price": $108 }  
    ]  
}
```



# Basics of CIF



# Basics of CIF: schemas

- Information about the structure of data
- A class in OOP contains structure
- UML, petri nets, etc. contain structure
- CIF currently can:
  - Check, register and unregister schemas
  - Interpret schemas in JSON

```
{
  "namespace": "preesm",
  "class": "graph",
  "schema": {
    "properties": [
      {
        "name": "name",
        "optional": false,
        "set": false,
        "value": {
          "type": "str",
          "optional": false,
          "collection": null,
          "constrains": [],
          "default": null,
          "object": null
        }
      },
      {
        "name": "actors",
        "optional": false,
        "set": true,
        "value": {...
      }
    ],
    {
      "name": "edges",
      "optional": false,
      "set": true,
      "value": {...
    }
  ]
}
```

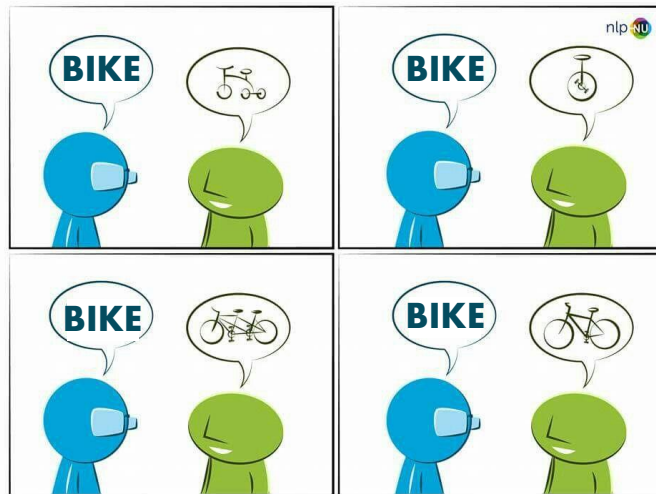
# Basics of CIF: models

- Information in your own format
- CIF can:
  - Write a model (from CIF)
  - Read a model (to CIF)

```
{
  "name": "03-parallel_sobel",
  "actors": [{
    {
      "id": "Read_YUV",
      "name": "Read_YUV",
      "kind": "vertex",
      "nbRepeat": "1"
    }, {
      "id": "display",
      "name": "display",
      "kind": "vertex",
      "nbRepeat": "1"
    }, {
      "id": "Sobel",
      "name": "Sobel",
      "kind": "vertex",
      "nbRepeat": "8"
    }, {
      "id": "Split",
      "name": "Split",
      "kind": "vertex",
      "nbRepeat": "1"
    }, {
      "id": "Merge",
      "name": "Merge",
      "kind": "vertex",
      "nbRepeat": "1"
    }
  ],
  "edges": [{
    "source": {
      "id": "Read_YUV",
```

# Basics of CIF: equivalence

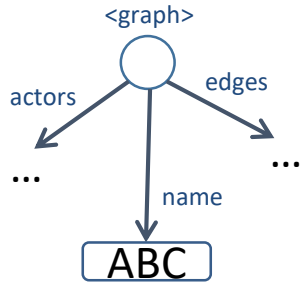
- Rules which define which classes and properties are the same.
- CIF:
  - Has a rule syntax
  - Has class and property equivalence
  - Can check rules, interpret rules
  - Expand an existing model from a set of rules (called simple merge)



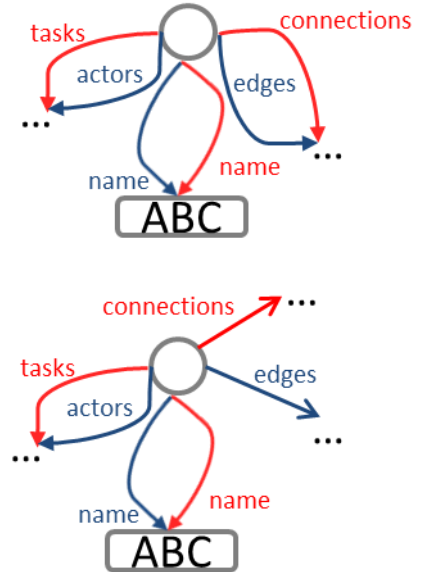
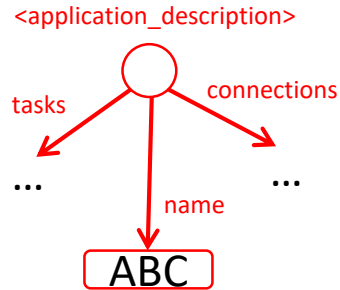
```
preesm:graph == dynaa:application_description
  IMPLYING
    (preesm:graph.name == dynaa:application_description.name,
     preesm:graph.actors == dynaa:application_description.tasks,
     preesm:graph.edges == dynaa:application_description.connections);
```

# Basics of CIF: equivalence

PREESM



DynAA



```
preesm:tool_description == dynaa:application_description
IMPLYING
    (preesm:tool_description.name == dynaa:application_description.name,
     preesm:tool_description.actors == dynaa:application_description.tasks,
     preesm:tool_description.edges == dynaa:application_description.connections);
```

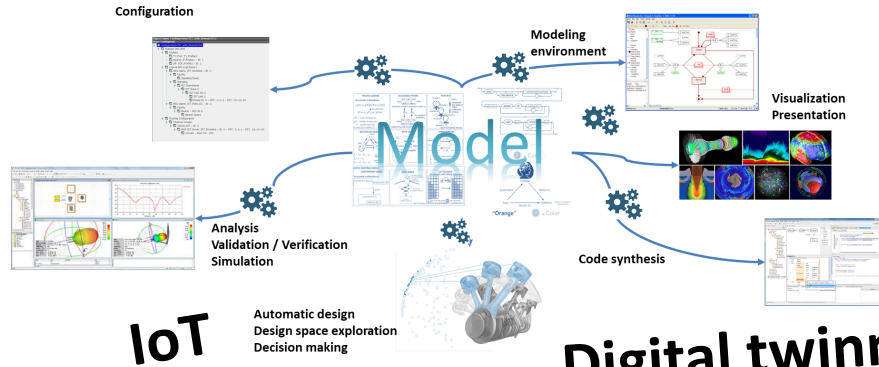
# Future applications of CIF

World modelling

Video stream processing

Standard sets of  
equivalence for  
standard interfaces

Open source  
community



CIF compatible  
tooling

Digital twinning

CIF as a service

Real time interoperability

Model based design

# Tutorial

A nighttime photograph of a city street. In the foreground, a modern building with a curved, glass-enclosed walkway or ramp is visible, illuminated with warm lights. The background shows a city street with light trails from moving vehicles, including a prominent green light trail. Buildings with lit windows are visible in the background.

› **THANK YOU FOR YOUR  
ATTENTION**

Take a look:  
**[TNO.NL/TNO-INSIGHTS](https://www.tno.nl/tno-insights)**

**TNO** innovation  
for life