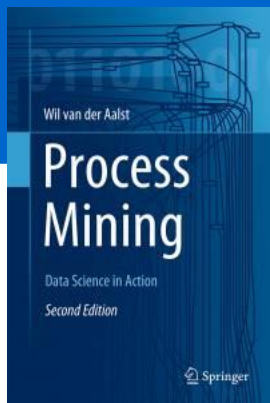


*Process Mining: Data Science in Action*

# Workflow Nets and Soundness

prof.dr.ir. Wil van der Aalst  
[www.processmining.org](http://www.processmining.org)



**TU/e**

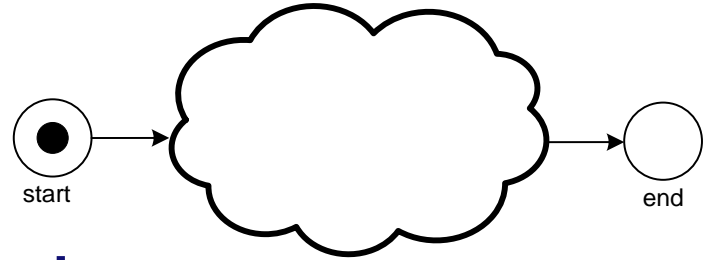
Technische Universiteit  
**Eindhoven**  
University of Technology

**Where innovation starts**



# Motivation

- For process mining we often use (or aim at) **Workflow Nets (WF-nets)**.
- **WF-nets:**
  - have a well-defined start and end
  - should be free of obvious anomalies (soundness)
- WF-nets are a subclass of Petri nets often used in the context of workflow management and business process management (systems).



# Business Process Management

**"Business Process Management (BPM) is the discipline that combines knowledge from information technology and knowledge from management sciences and applies this to operational business processes"**



Wil M. P. van der Aalst, "Business Process Management: A Comprehensive Survey," ISRN Software Engineering, vol. 2013, Article ID 507984, 37 pages, 2013. doi:10.1155/2013/507984

**Adam Smith (1723-1790)**

**sh Frederick Taylor (1856-**

**the 1915 Henry Ford (1863-1947)**

**princinti Since 1950**

**man line**

**dict and communication**

**Infrastructure**

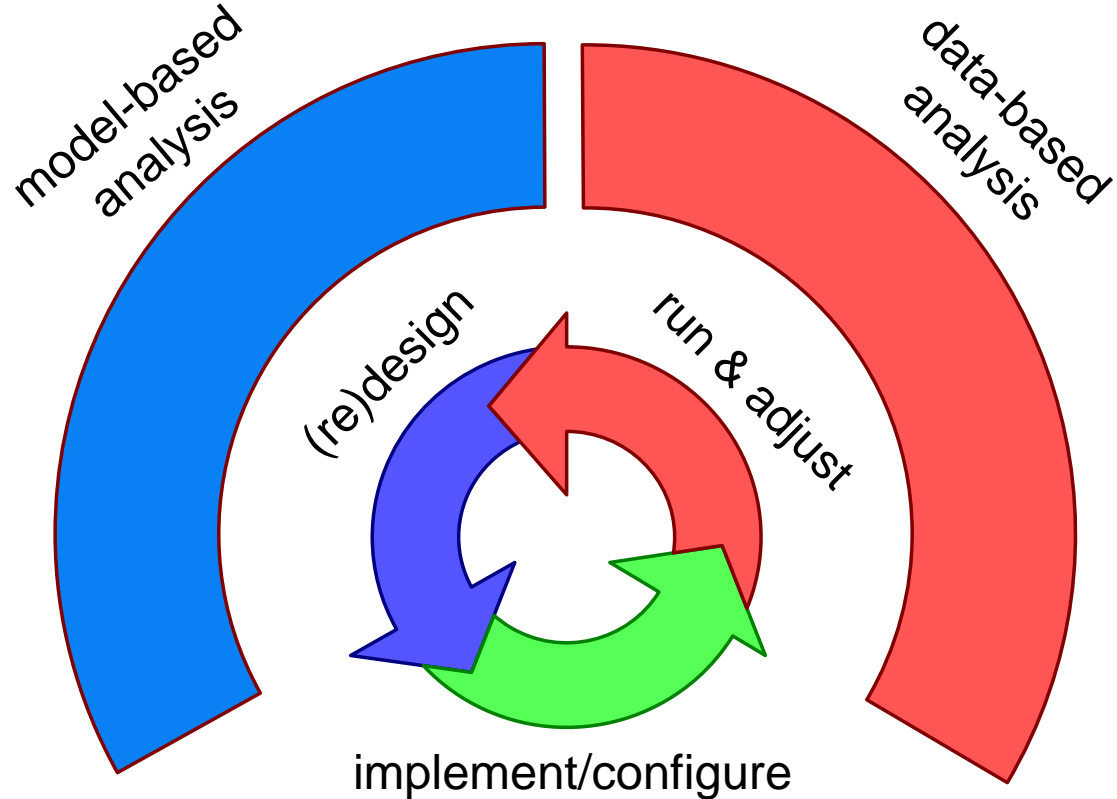
**don the most**

**factor influencing  
business processes and their  
management.**

**Data science in Action: The next level!**

**Data = the new oil**

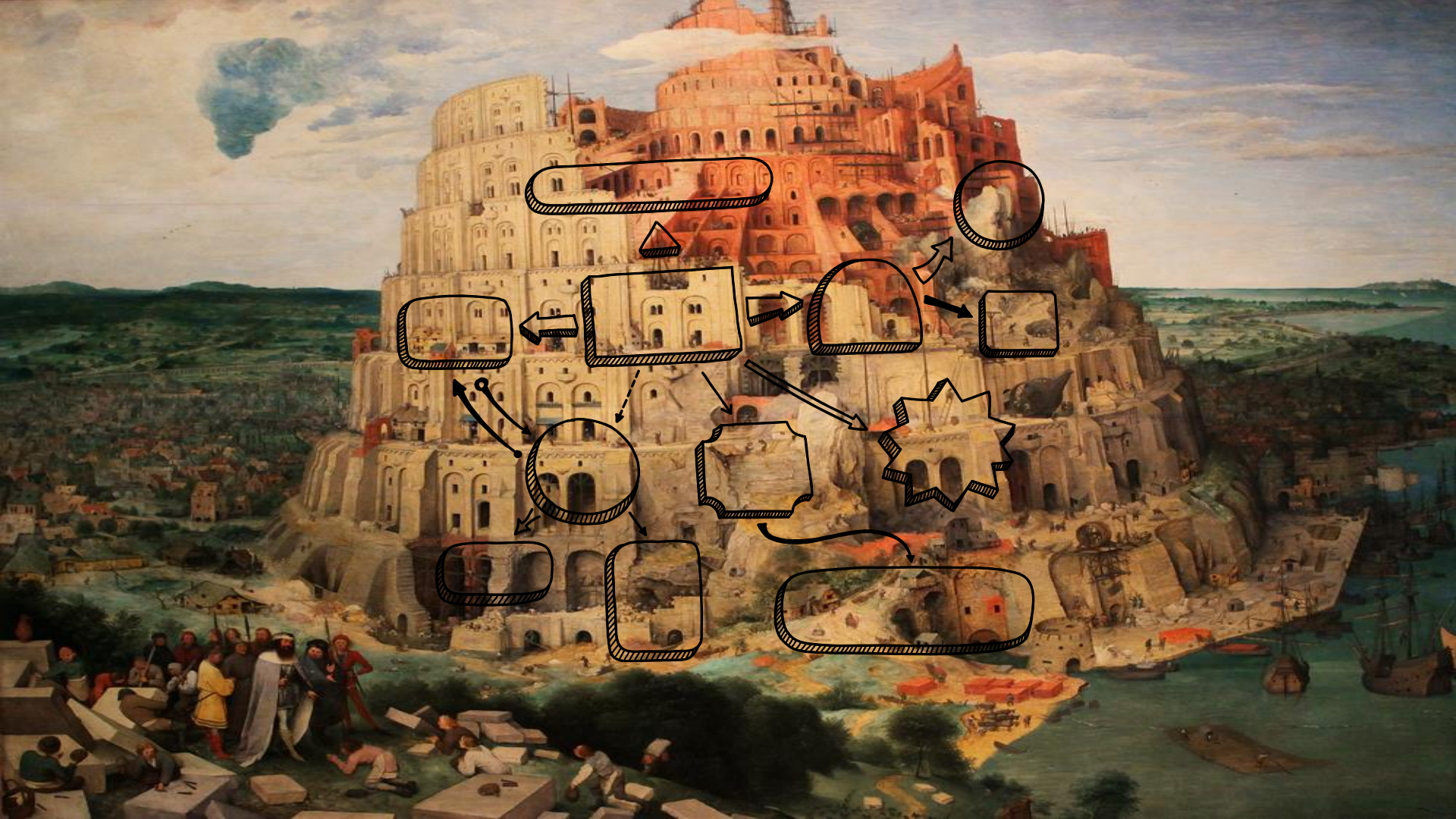
# BPM lifecycle



# What is the role of (process) models?

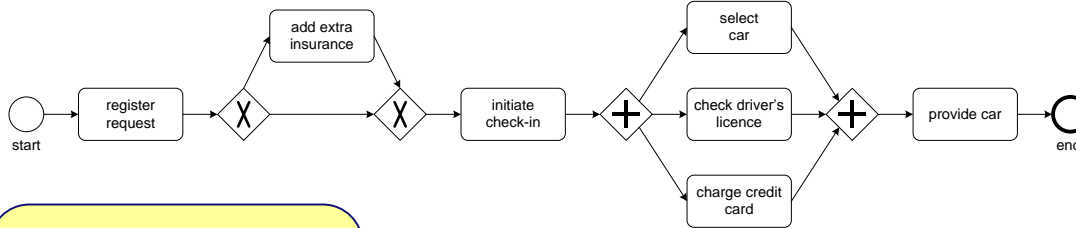
- Role of models in BPM/WFM:
  - **reason *about processes*** (redesign) and
  - **make decisions *inside processes*** (planning and control).
- **Process models** may be used to:
  - discuss responsibilities,
  - analyze compliance,
  - predict performance using simulation, and
  - configure a WFM/BPM system.





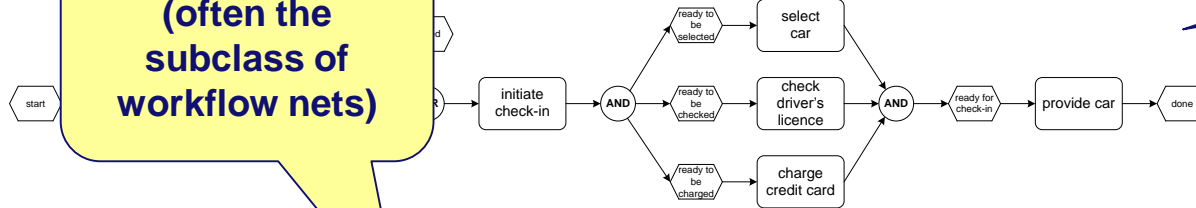
# Many notations

Business Process Model  
and Notation (BPMN)

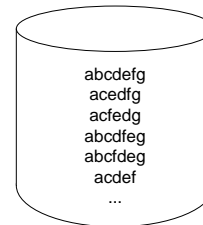
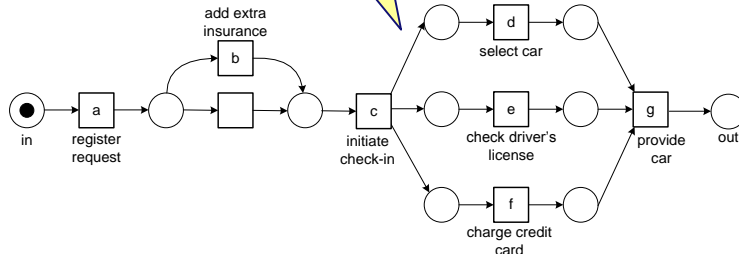


Event-Driven Process  
Chains (EPCs)

Petri nets  
(often the  
subclass of  
workflow nets)

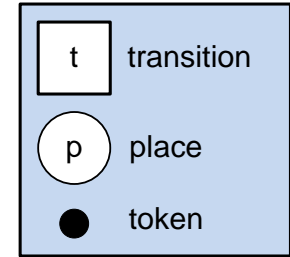
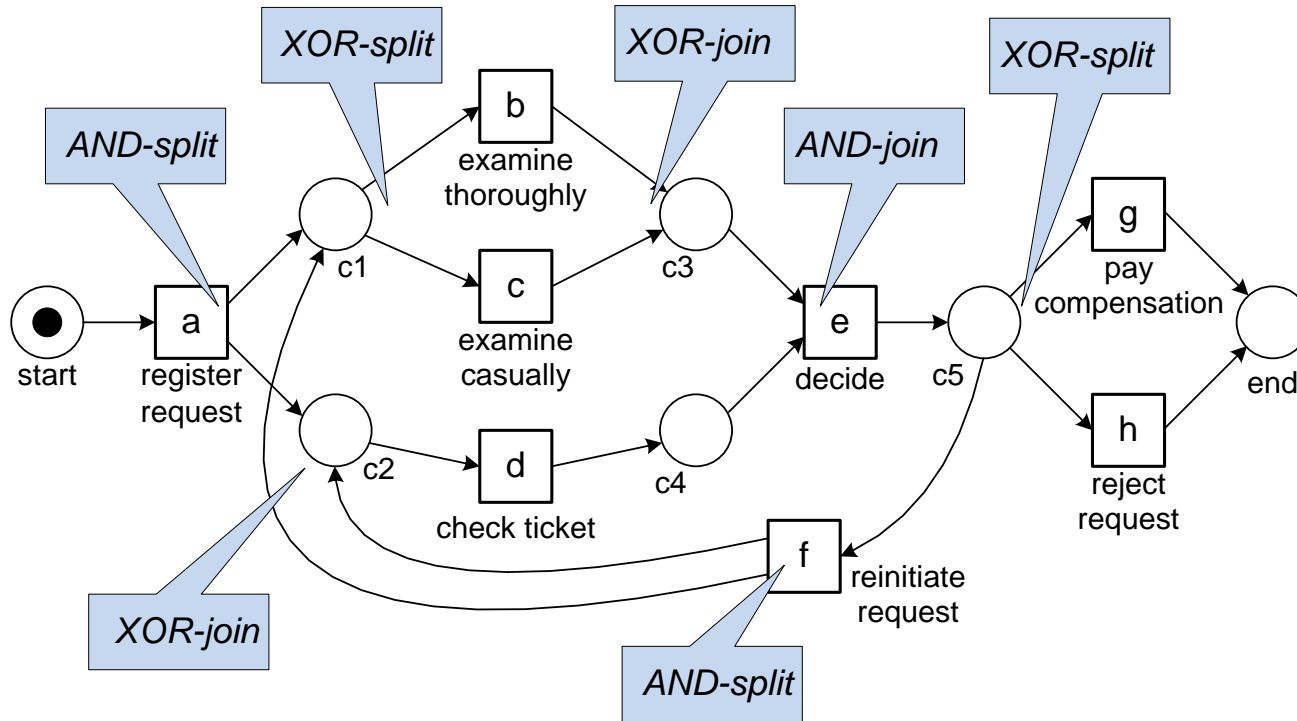


Event log  
(example/observed  
behavior)

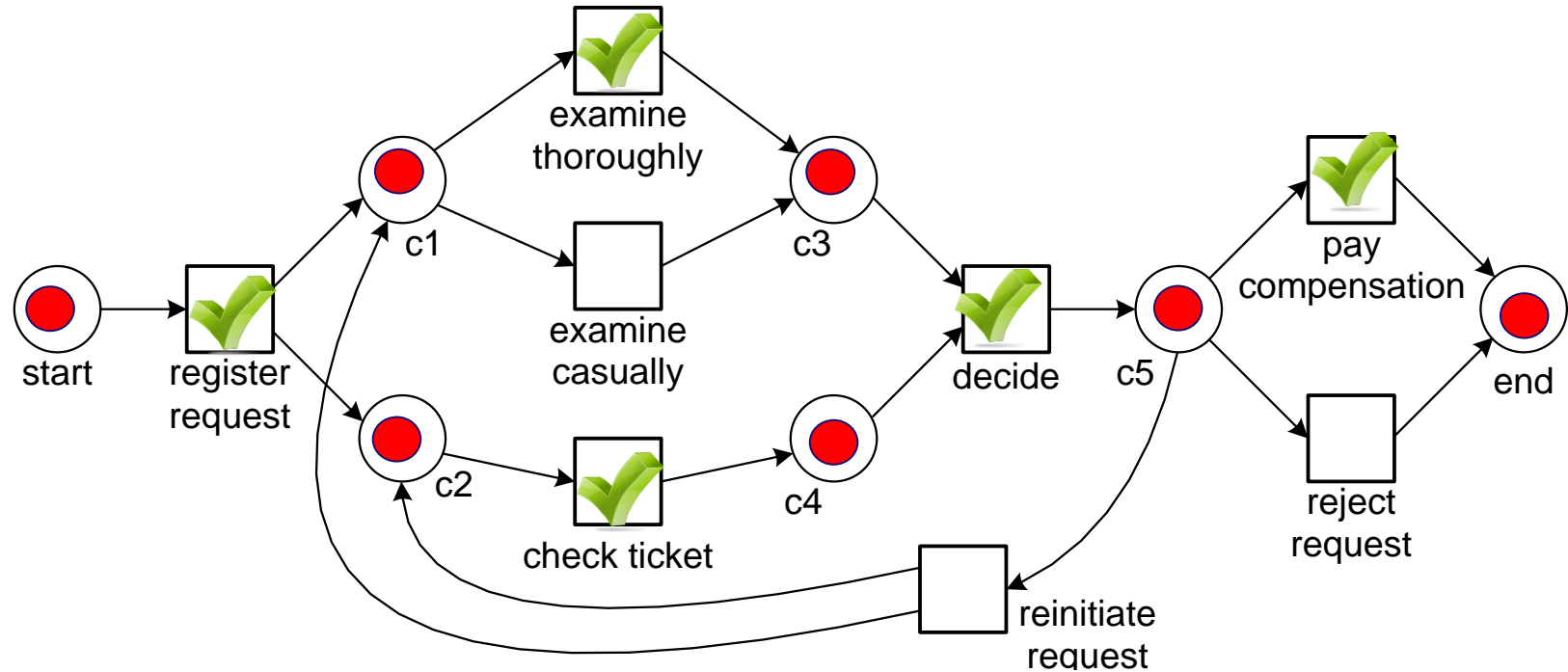




# Petri nets (as seen before)

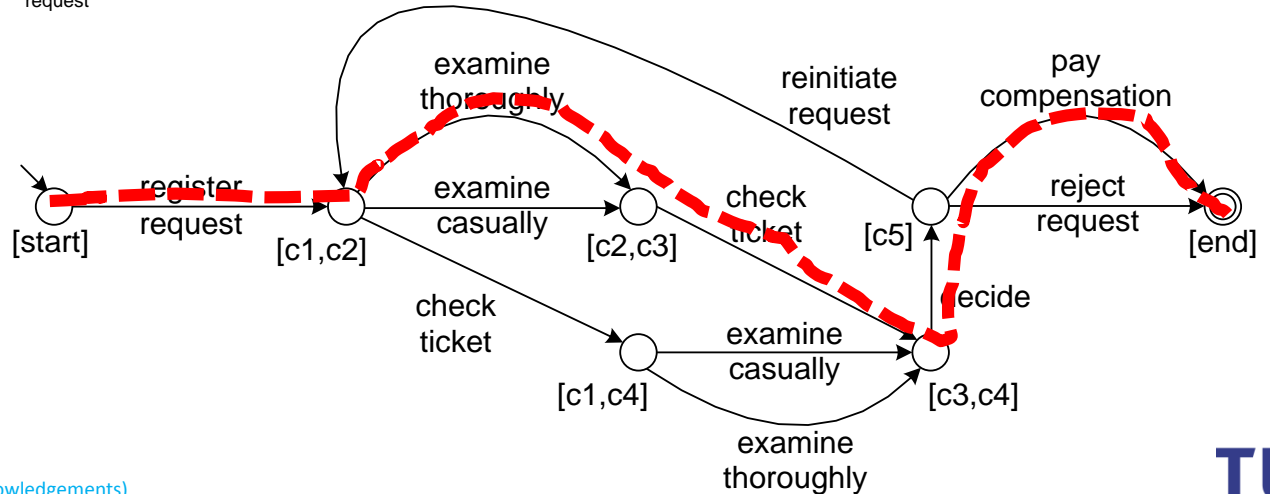
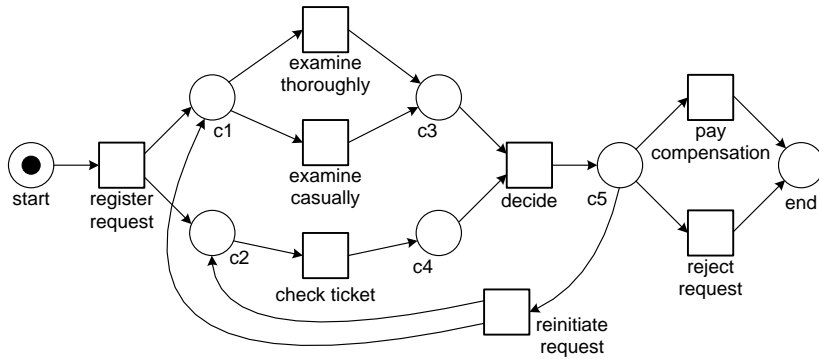


# Example run of the model

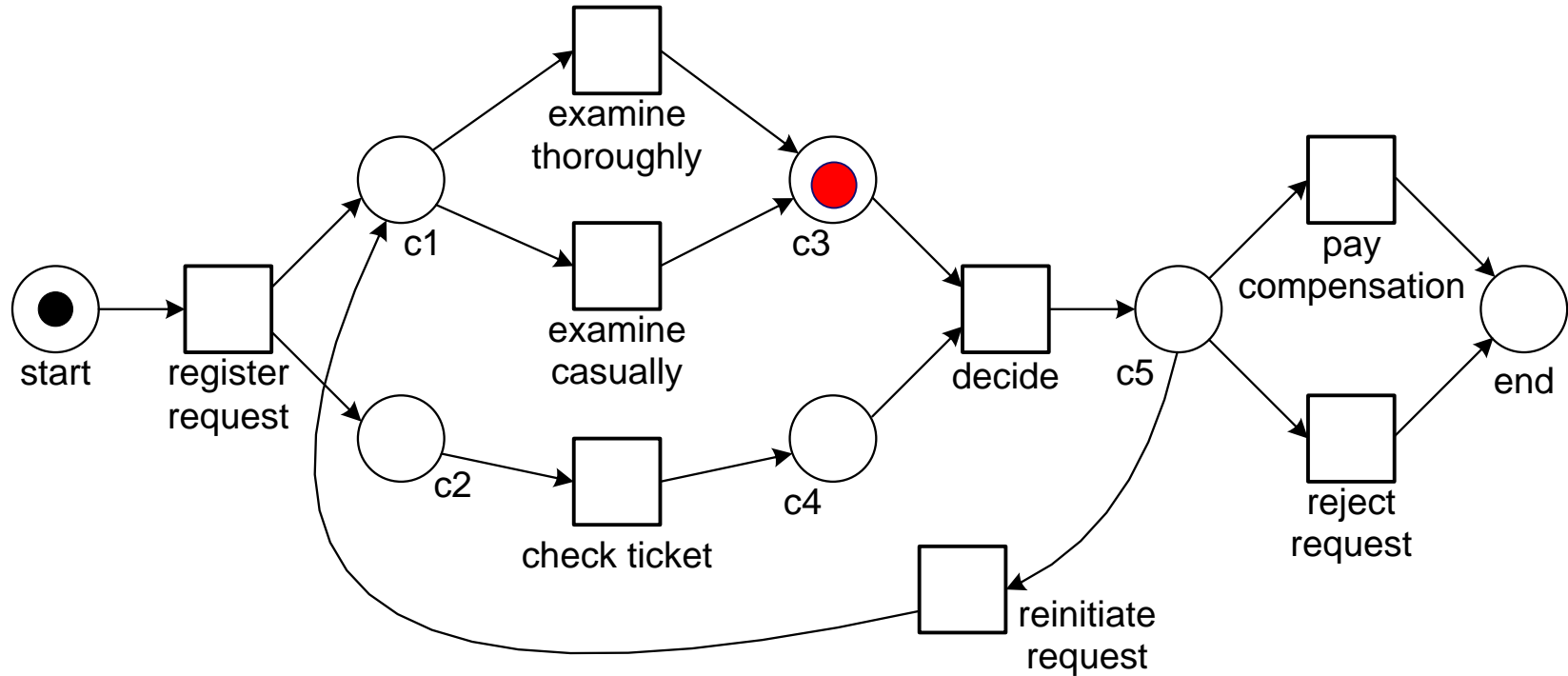


Only one of infinitely many possible firing sequences! **nd]**

# Reachability graph

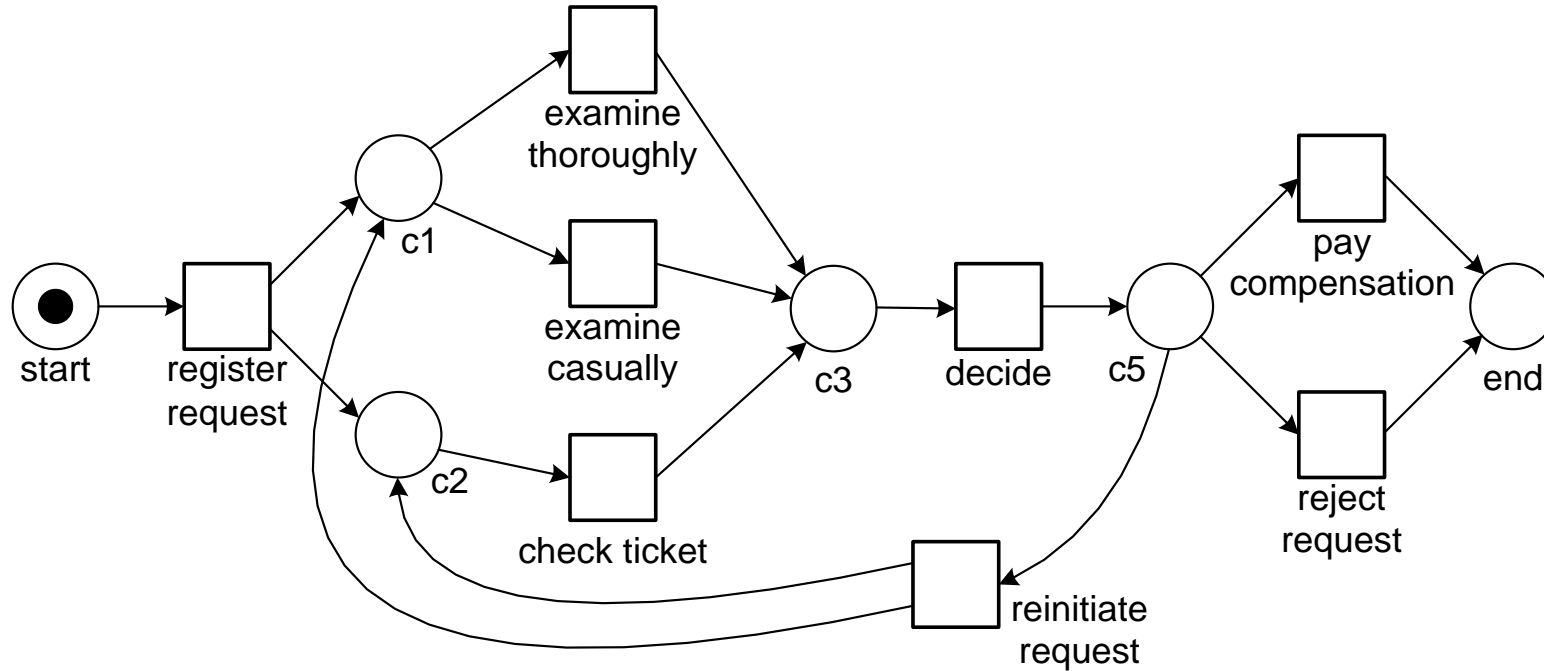


# Good model?



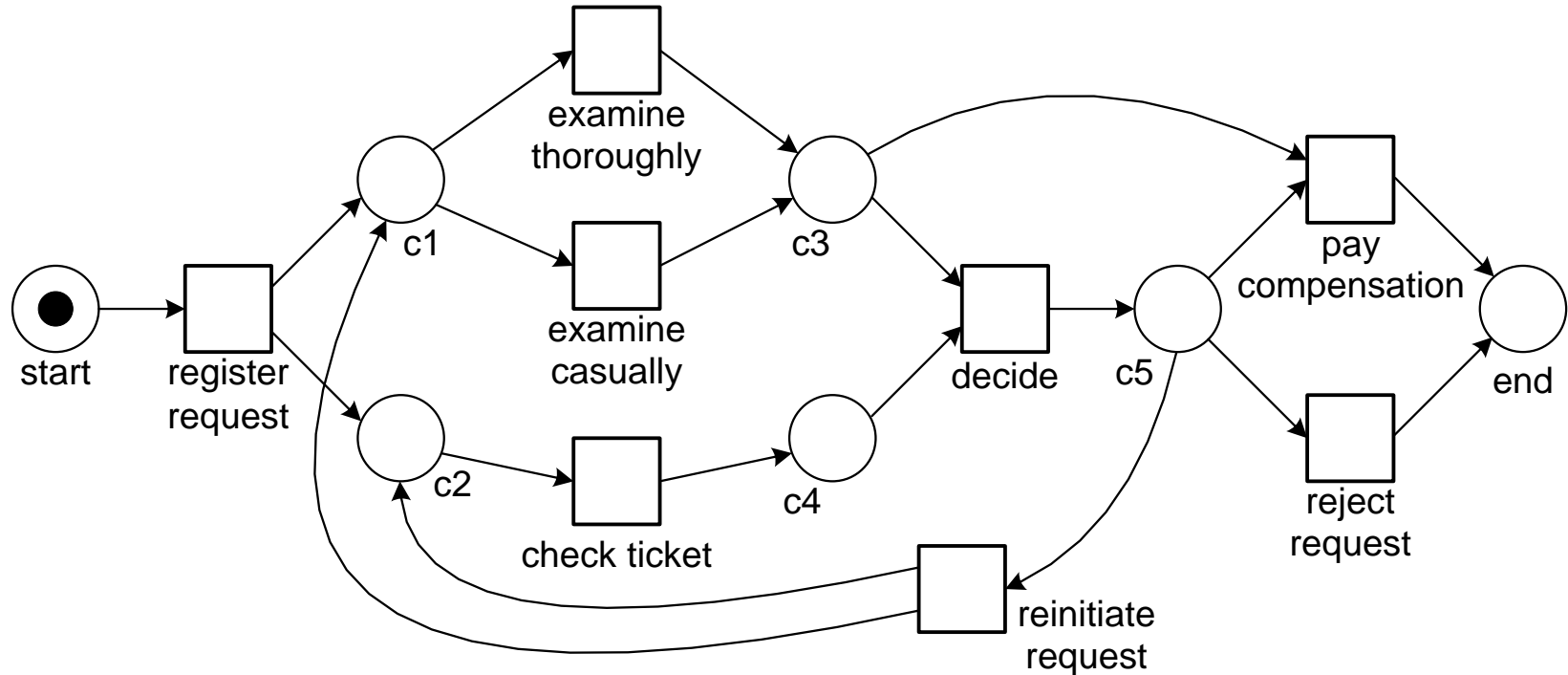
**No, deadlock possible: [c3].**

# Good model?



**No, AND-split does not have corresponding join!**

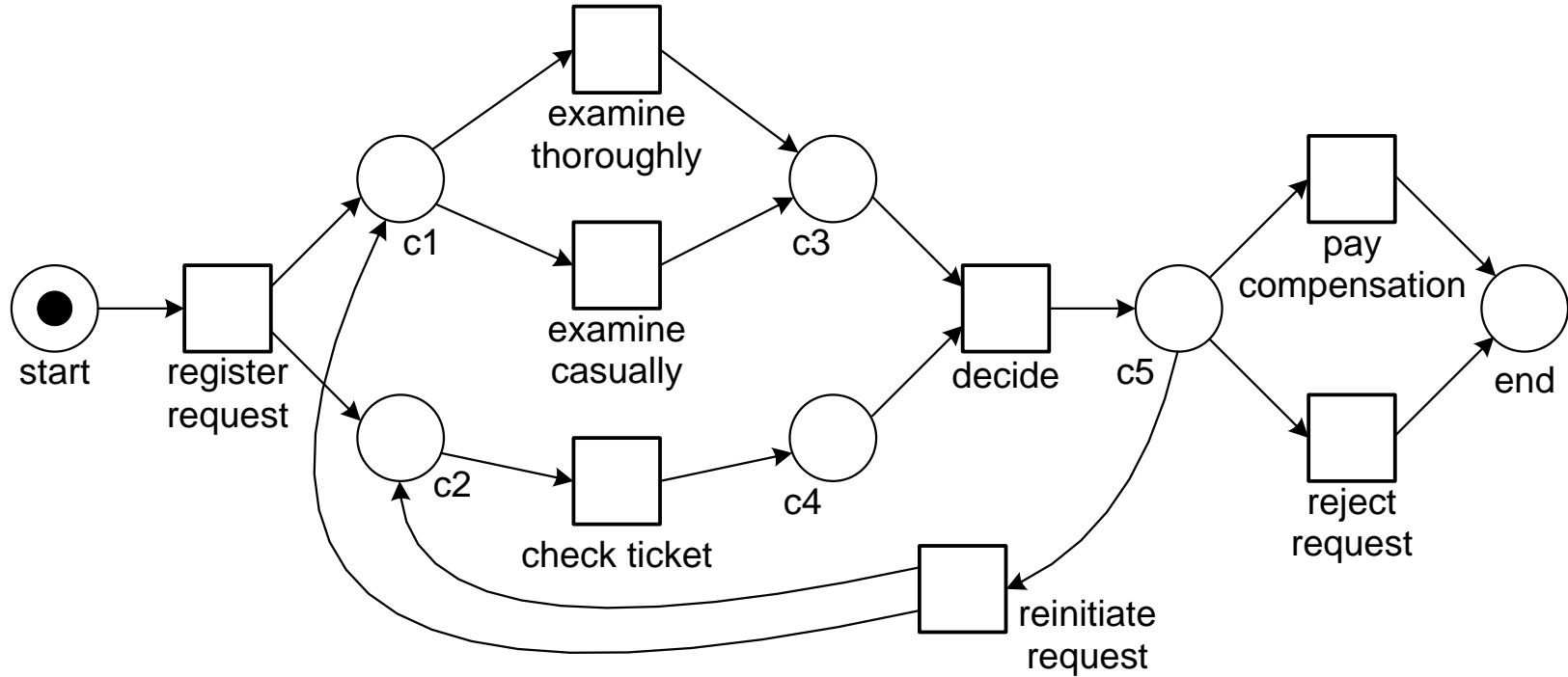
# Good model?



**No, "pay compensation" is dead!**



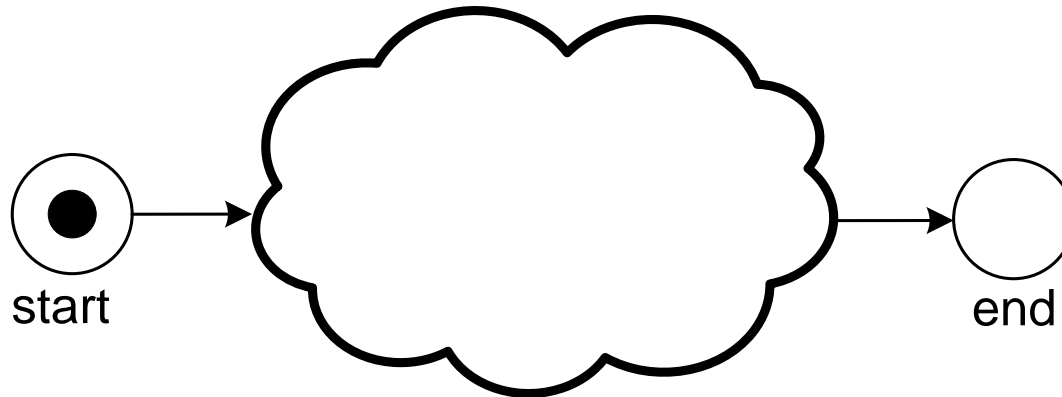
# Good model?



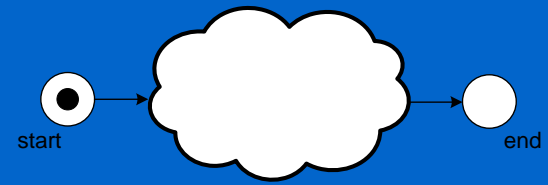
**Yes, but why?**

# WF-nets

A **WorkFlow net (WF-net)** has **one source place** (typically called *start* or *i*) and **one sink place** (typically called *end* or *o*) and **all other nodes are on a path from source to sink.**



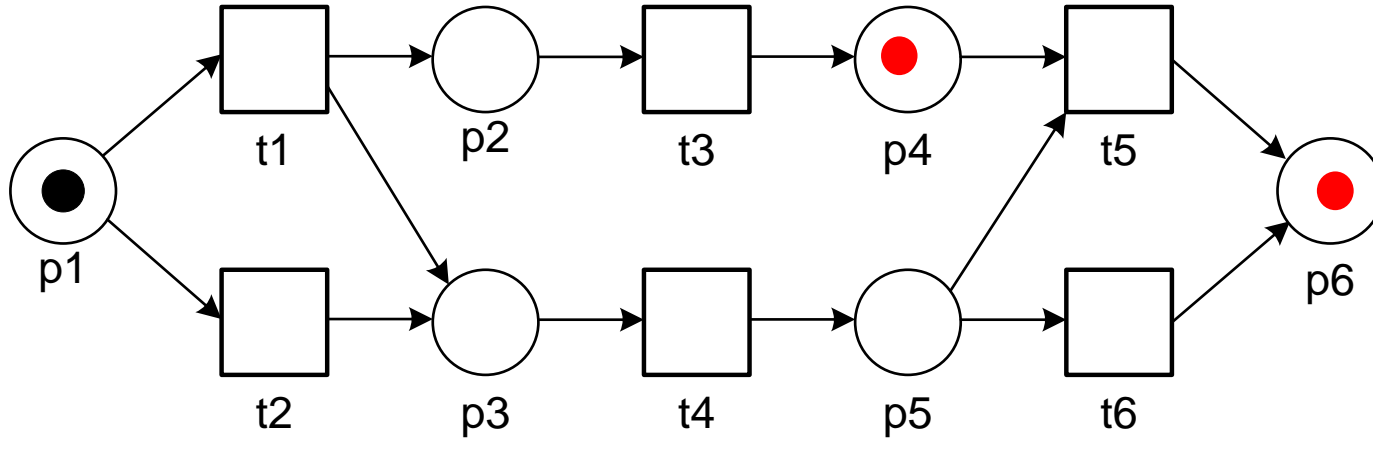
# Soundness



A WF-net is **sound** if and only if the following properties hold:

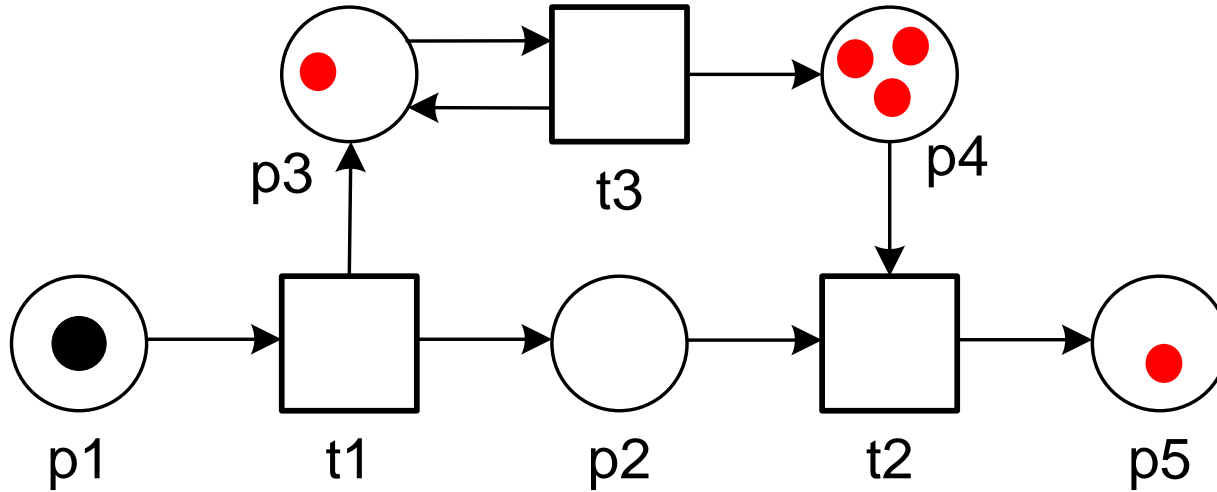
- **safeness**: places cannot hold multiple tokens at the same time,
- **proper completion**: if the sink place is marked, all other places are empty,
- **option to complete**: it is always possible to reach the marking that marks just the sink place, and
- **absence of dead parts**: for any transition there is a firing sequence enabling it.

# Sound?



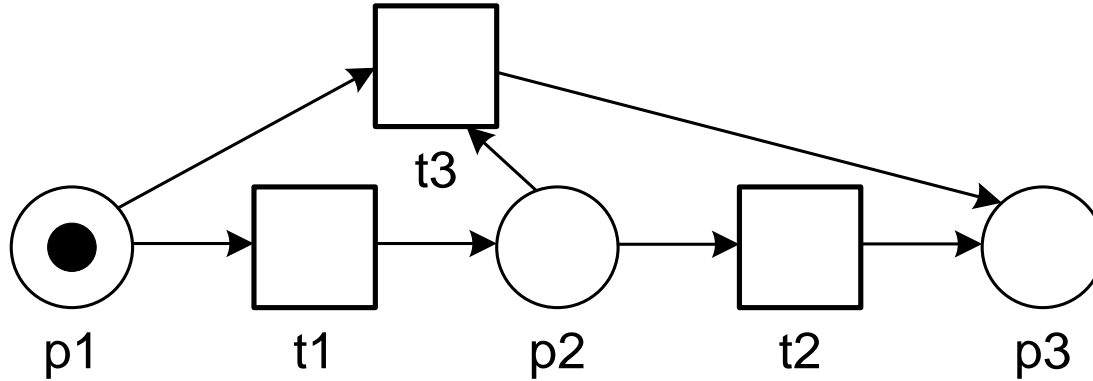
- **safeness**: places cannot hold multiple tokens at the same time
- **proper completion**: if the sink place is marked, all other places are empty
- **option to complete**: it is always possible to reach the final target marking
- **absence of dead parts**: for any transition there is a firing sequence enabling it

# Sound?



- **~~safeness~~**: places cannot hold multiple tokens at the same time
- **~~proper completion~~**: if the sink place is marked, all other places are empty
- **~~option to complete~~**: it is always possible to reach the final target marking
- **absence of dead parts**: for any transition there is a firing sequence enabling it

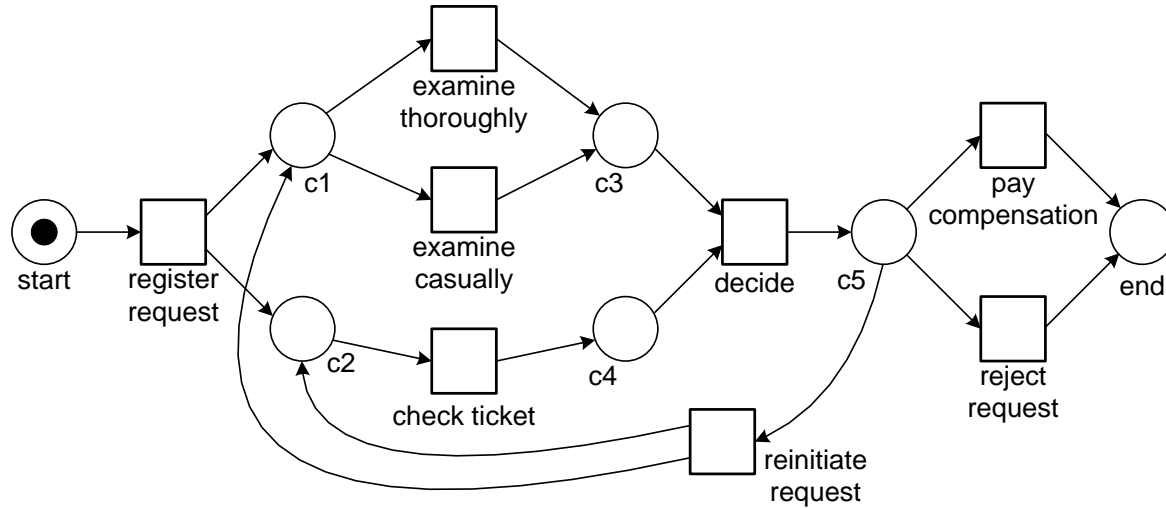
# Sound?



- **safeness**: places cannot hold multiple tokens at the same time
- **proper completion**: if the sink place is marked, all other places are empty
- **option to complete**: it is always possible to reach the final target marking
- ~~absence of dead parts~~: for any transition there is a firing sequence enabling it



# Sound?



- **safeness:** places cannot hold multiple tokens at the same time
- **proper completion:** if the sink place is marked, all other places are empty
- **option to complete:** it is always possible to reach the final target marking
- **absence of dead parts:** for any transition there is a firing sequence enabling it

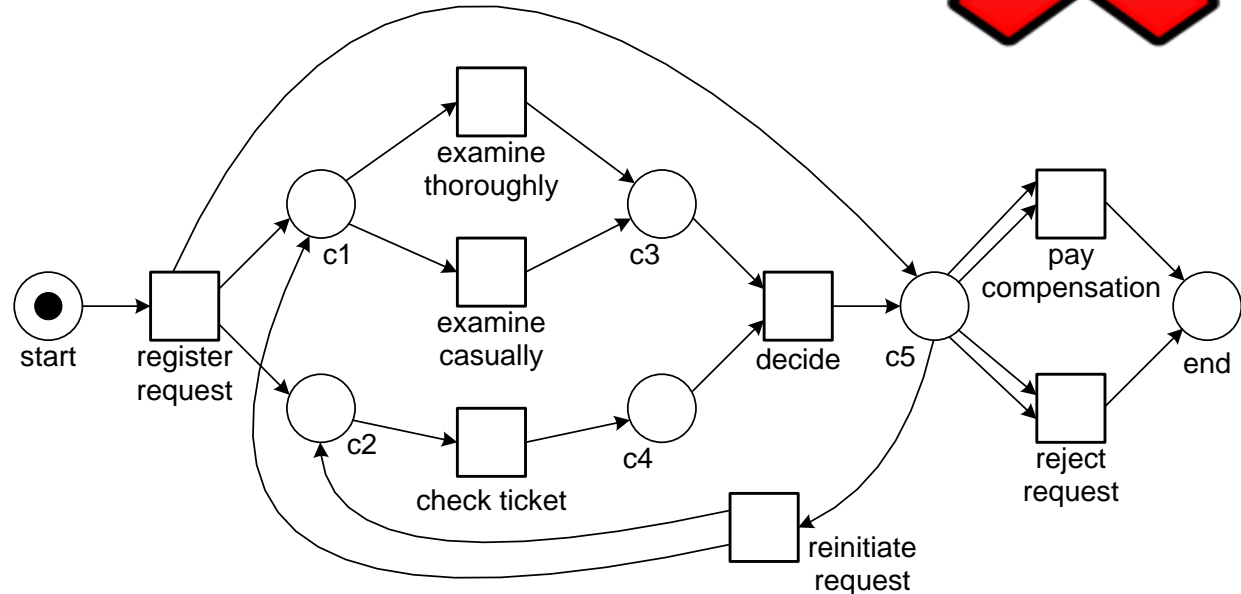


**Checking soundness  
may be far from  
trivial for larger  
examples.**

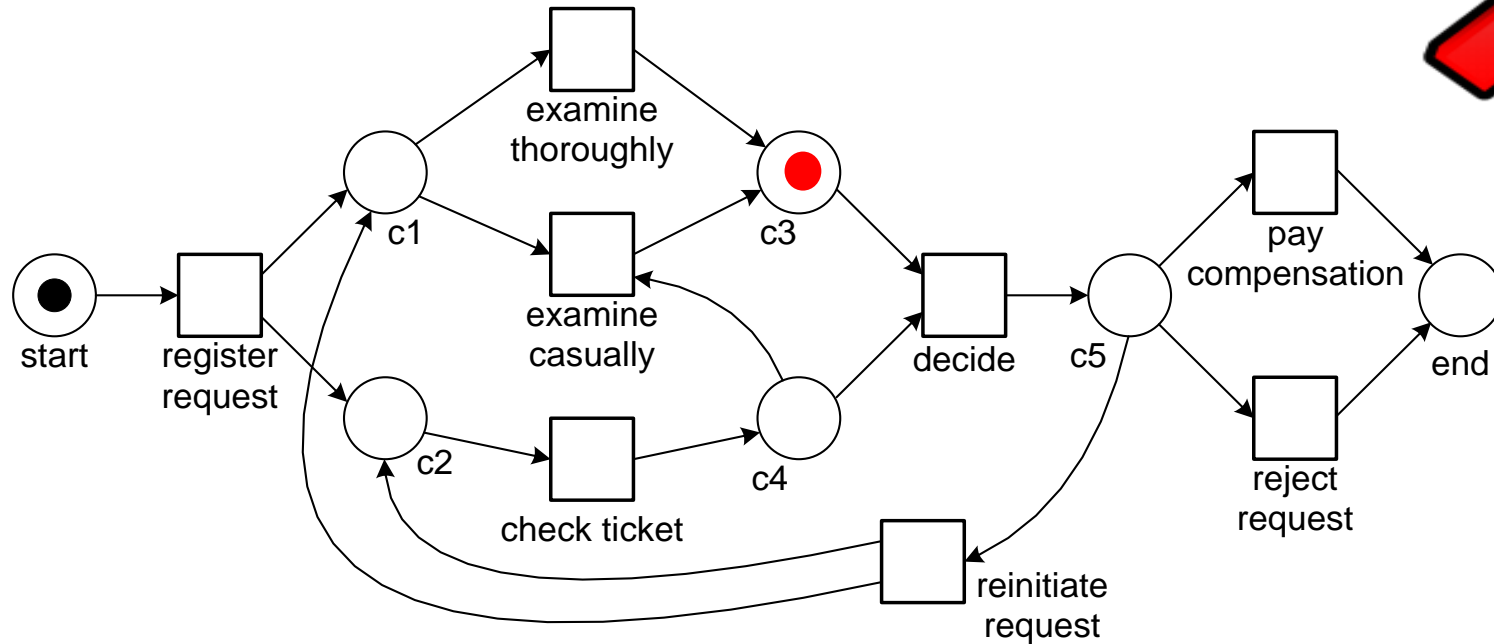
# Not sound: Unsafe

Examples of  
reachable markings:

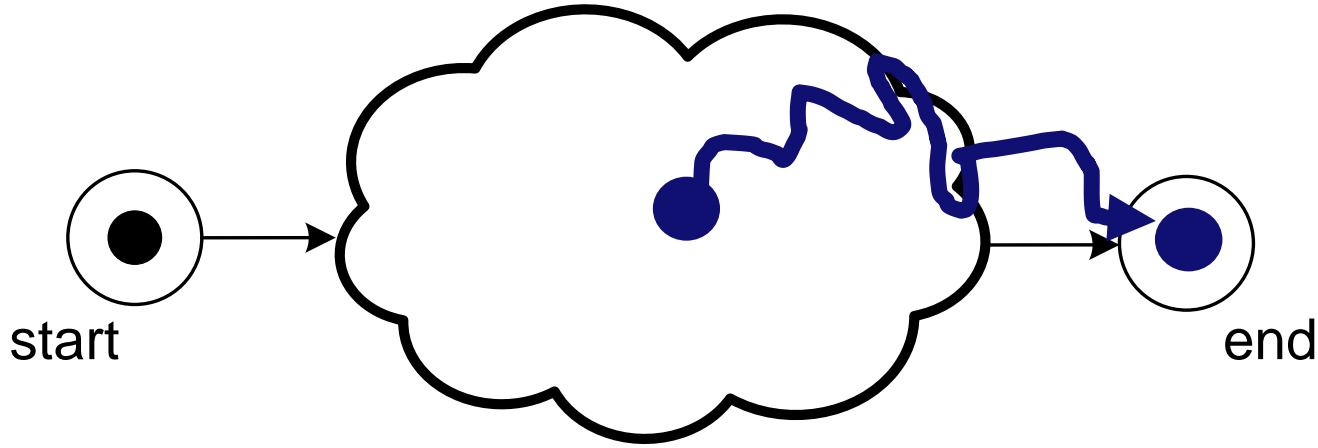
- $[c5^2]$
- $[c1, c2, c3, c4]$
- $[c1^2, c2^2]$
- etc.



# Not sound: No option to complete in [c3]

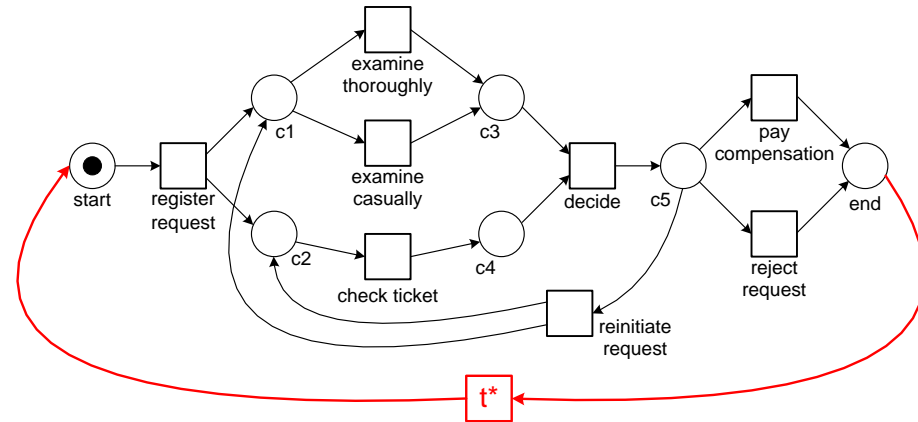
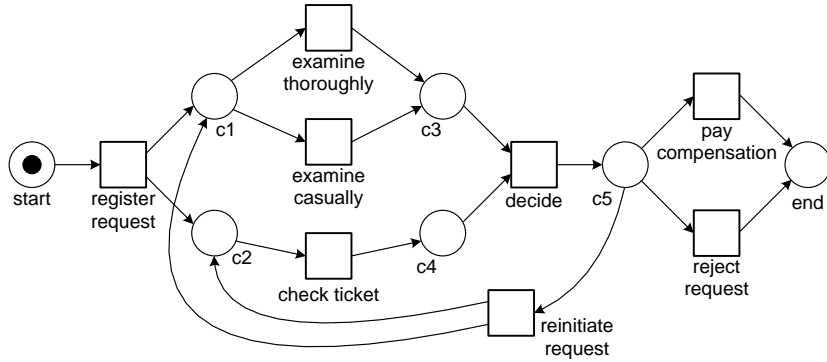


# No need to check proper completion: It is implied by other properties



**option to complete** (it is always possible to reach the marking that marks just the sink place) implies **proper completion** (if the sink place is marked all other places are empty)

# Link between soundness and classical Petri net properties



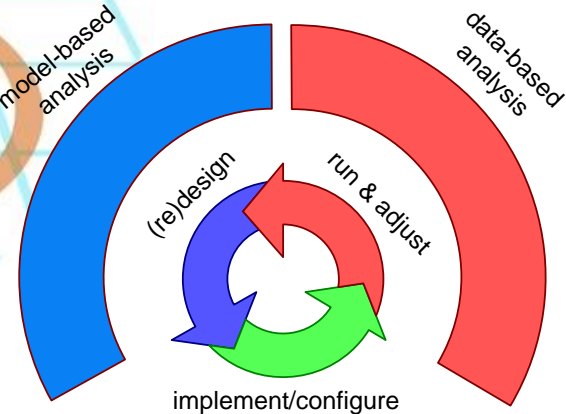
**A WF-net is sound if and only if the corresponding "short-circuited" Petri net is live and bounded!**



# Main types of model-based analysis

**verification (like  
soundness checking)**

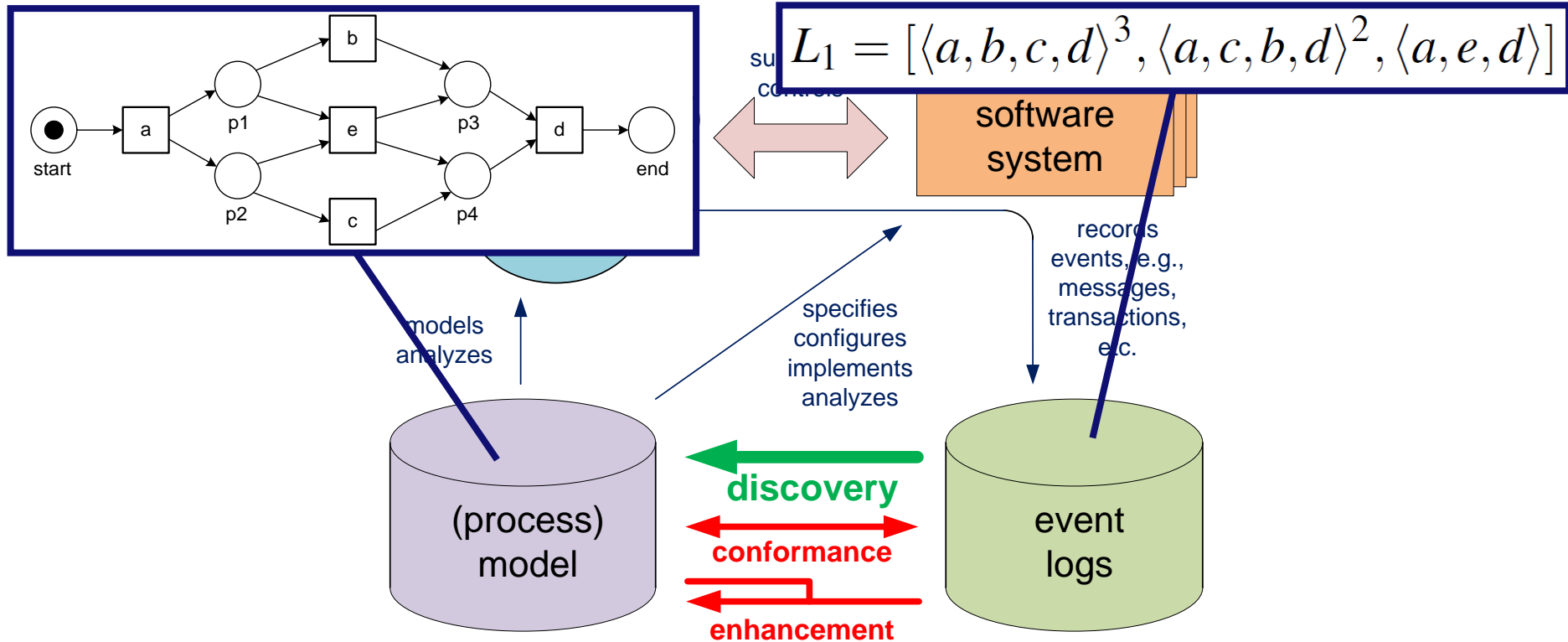
**performance analysis  
(simulation)**



# Limitations of model-based analysis

- Verification and performance analysis heavily rely on the **availability of high quality models**.
- When the models and reality have little in common, model-based analysis does not make much sense!
- There is often a **poor alignment** between hand-made models and reality.
- **Process mining aims to address these problems by establishing a direct connection between the models and actual event data about the process.**

# Next: Using the **Alpha Algorithm** to discover WF-nets from event logs



## *Part I: Introduction*

### **Chapter 1**

Data Science  
in Action

### **Chapter 2**

Process Mining:  
The Missing Link

## *Part II: Preliminaries*

### **Chapter 3**

Process Modeling  
and Analysis

### **Chapter 4**

Data Mining

## *Part III: From Event Logs to Process Models*

### **Chapter 5**

Getting the Data

### **Chapter 6**

Process Discovery:  
An Introduction

### **Chapter 7**

Advanced Process  
Discovery Techniques

## *Part IV: Beyond Process Discovery*

### **Chapter 8**

Conformance  
Checking

### **Chapter 9**

Mining Additional  
Perspectives

### **Chapter 10**

Operational Support

## *Part V: Putting Process Mining to Work*

### **Chapter 11**

Process Mining  
Software

### **Chapter 12**

Process Mining in the  
Large

### **Chapter 13**

Analyzing “Lasagna  
Processes”

### **Chapter 14**

Analyzing “Spaghetti  
Processes”

## *Part VI: Reflection*

### **Chapter 15**

Cartography and  
Navigation

### **Chapter 16**

Epilogue

