INFO-H420
Management of Data Science and Business Workflows
Part I: Introduction

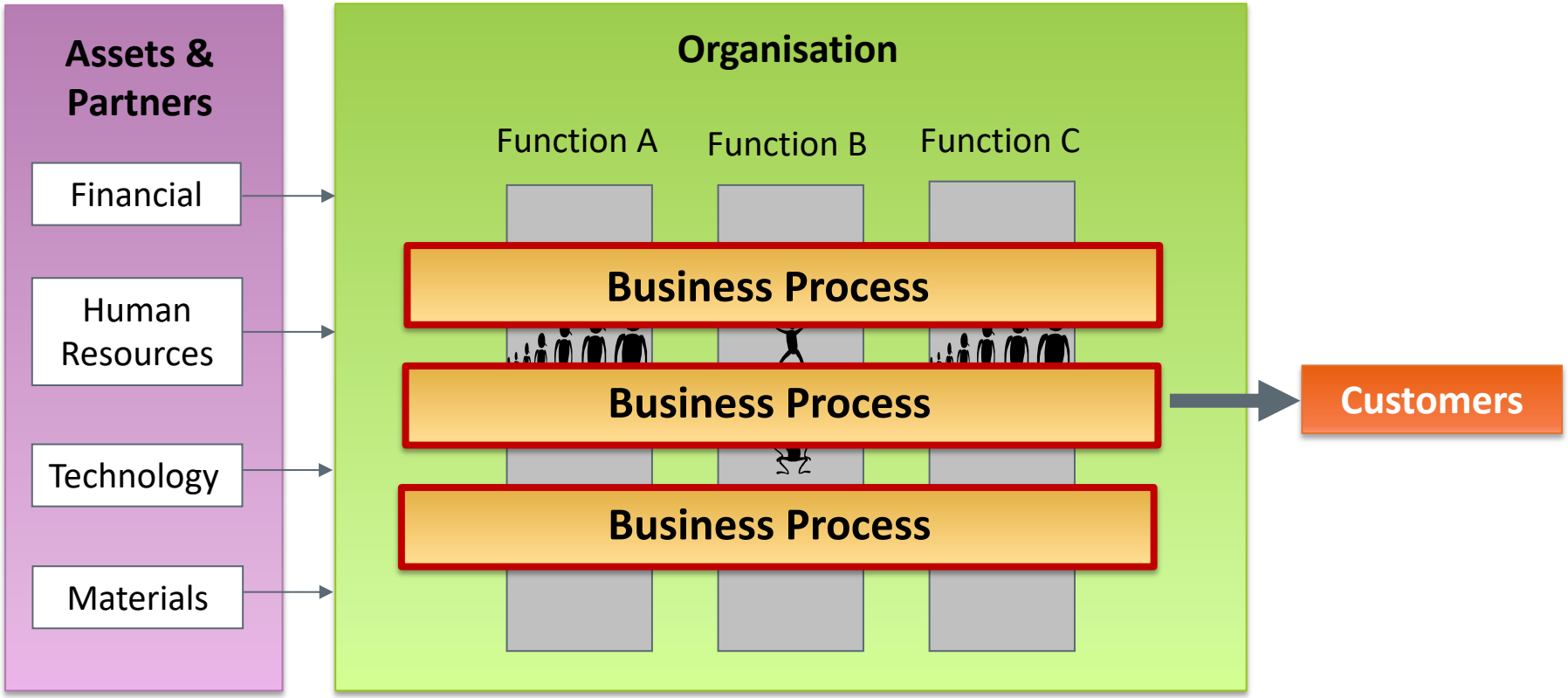
Dimitris SACHARIDIS

2023-2024

What is a Business Process?

Issue delivery receipt
Load truck
Package products
Issue invoice
Prepare shipment
Schedule payment
Schedule delivery
Check & confirm PO
Unload truck
Notify shipment
Obtain PO confirm.
Check Invoice
Request PO change
Match incoming payment

Business processes



PO received
Check & confirm PO
Package products
Load truck
Notify shipment
Issue invoice
Match payment
Payment made
P.O.
7465 8792
D. KAPIKA
PO issued
Obtain PO confirm.
Schedule delivery
Unload truck
Issue delivery receipt
Check invoice
Schedule payment
Goods arrived
P.O.
7465 8792
D. KAPIKA

Your turn…

Load dish-washer

Serve meal

Bring menu

Take order

Collect payment

Collect laundry

Brush grills

Unload dish-washer

Collect laundry

Clean kitchen surfaces

Sweep & mop

Greet & seat

Present bill

Customer arrived
Greet & seat
Take order
Bring menu
Serve meal
Present bill
Issue invoice
Customer paid
Kitchen is dirty
Load dish-washer
Clean kitchen surfaces
Brush grills
Collect laundry
Sweep & mop
Unload dish-washer
Kitchen is clean

A business process is…

a chain of events, activities and decisions
…involving a number of actors and objects,
…triggered by a need
and leading to an outcome that is of value to a customer.

Examples:

- Order-to-Cash
- Procure-to-Pay (aka Purchase-to-Pay)
- Application-to-Approval
- Issue-to-Resolution

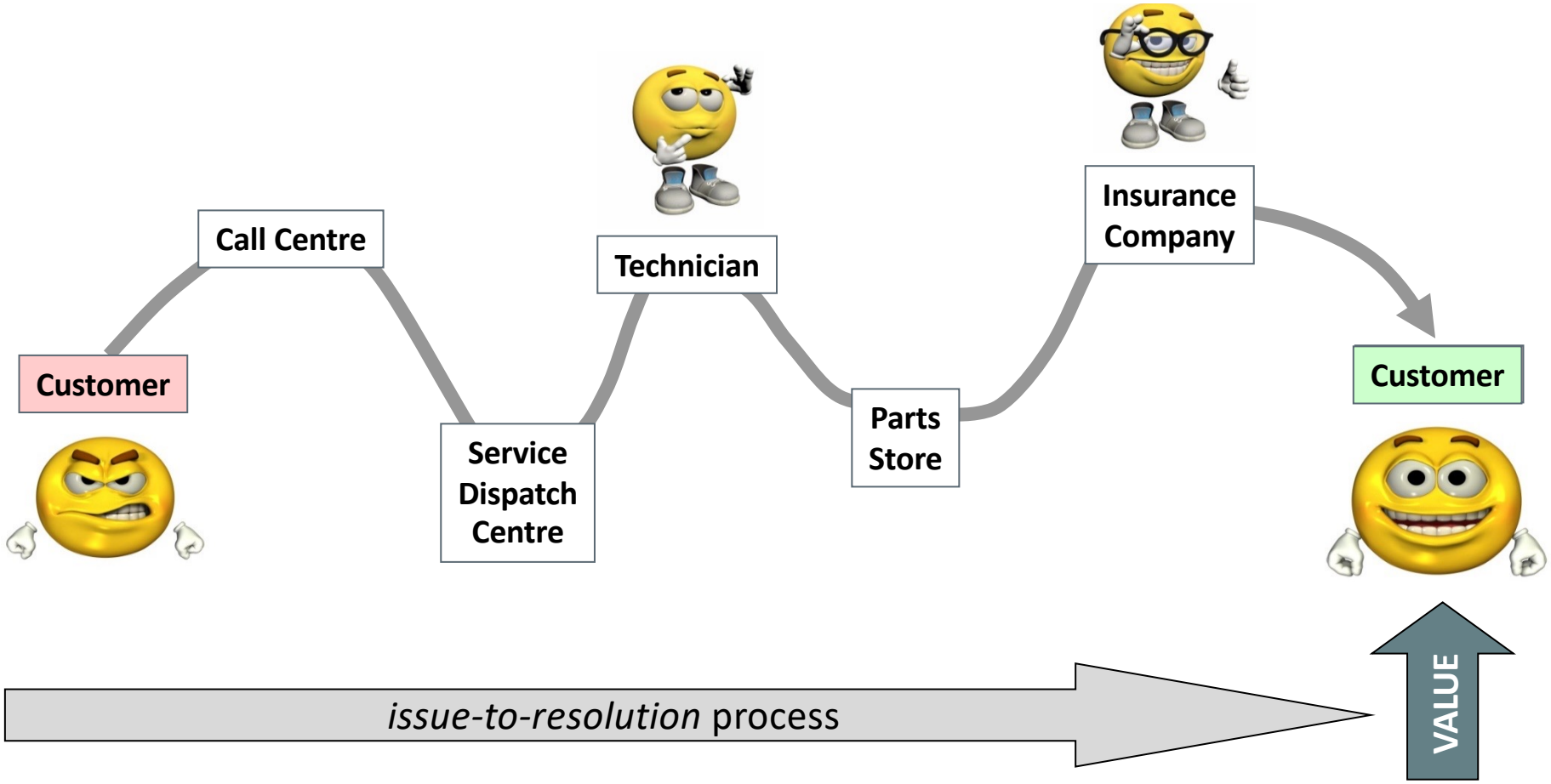
“My washing machine doesn’t work…”

Negative outcomes (value-reducing):

- Fault not repaired in a timely manner
- Fault repaired but customer pays more than expected

Positive outcomes (value-adding):

- Fault repaired immediately with minor intervention
- Fault repaired, covered by warranty



Your turn

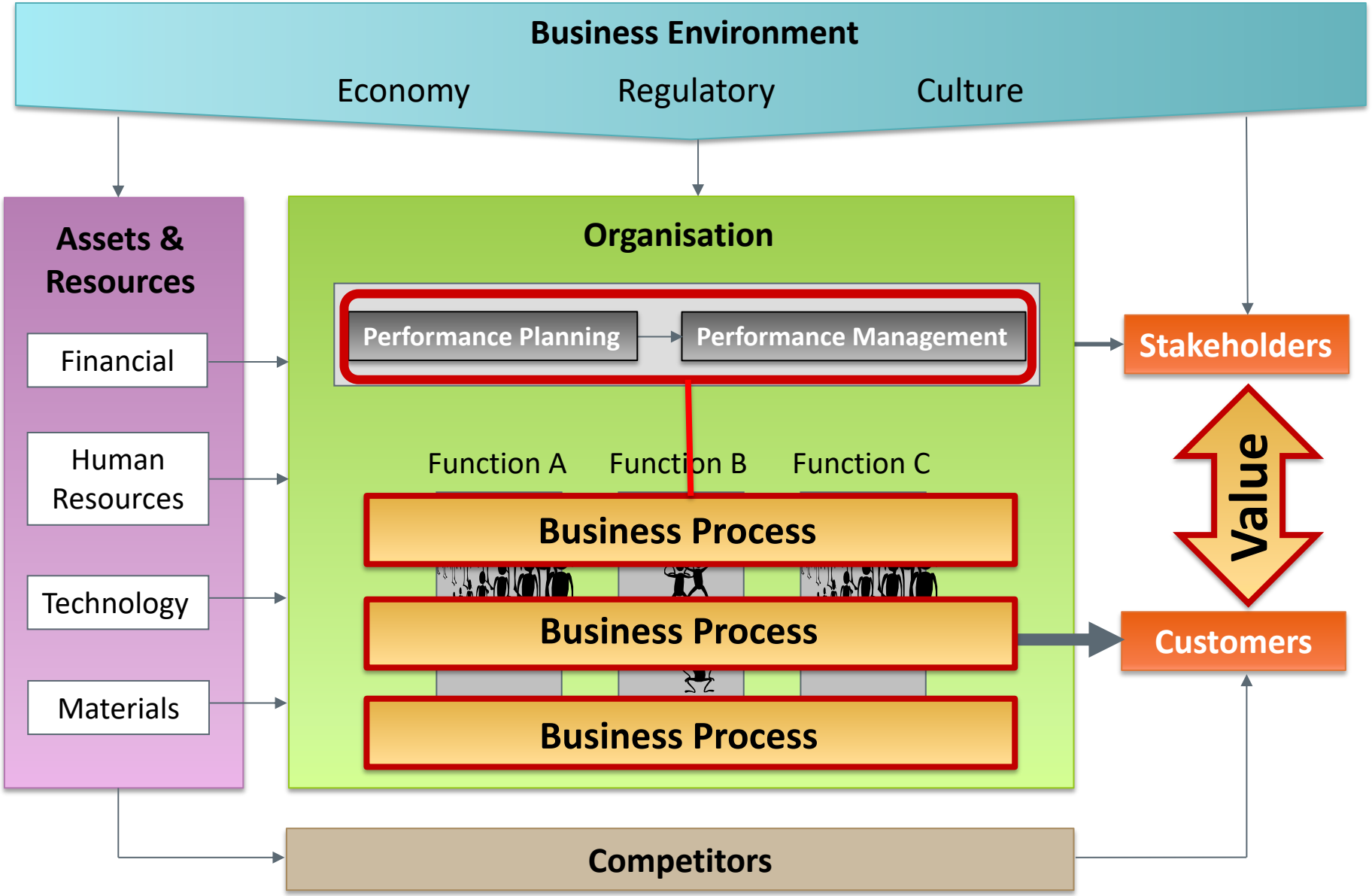
- Think of an organization and a process in this organization:
  - Is it order-to-cash, procure-to-pay, application-to-approval, issue-to-resolution…
  - Who is/are the customer(s)?
  - What value does this process deliver to its customer?
  - Who are the key actors of the process?
  - List at least 3 outcomes of the process.

ULB

What is Business Process Management?

And why should I care about it?

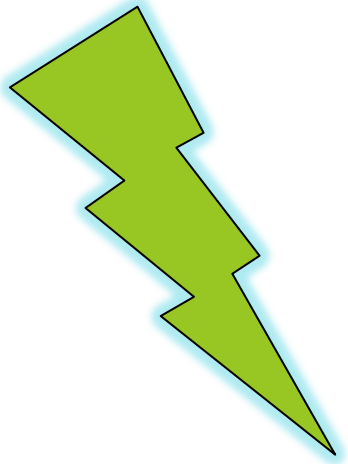
Improving Performance (Rummler's Framework)



Process performance

If you had to choose between two services, you would typically choose the one that is:

- F…
- C…
- B…



Process performance

If you had to choose between two services, you would typically choose the one that is:

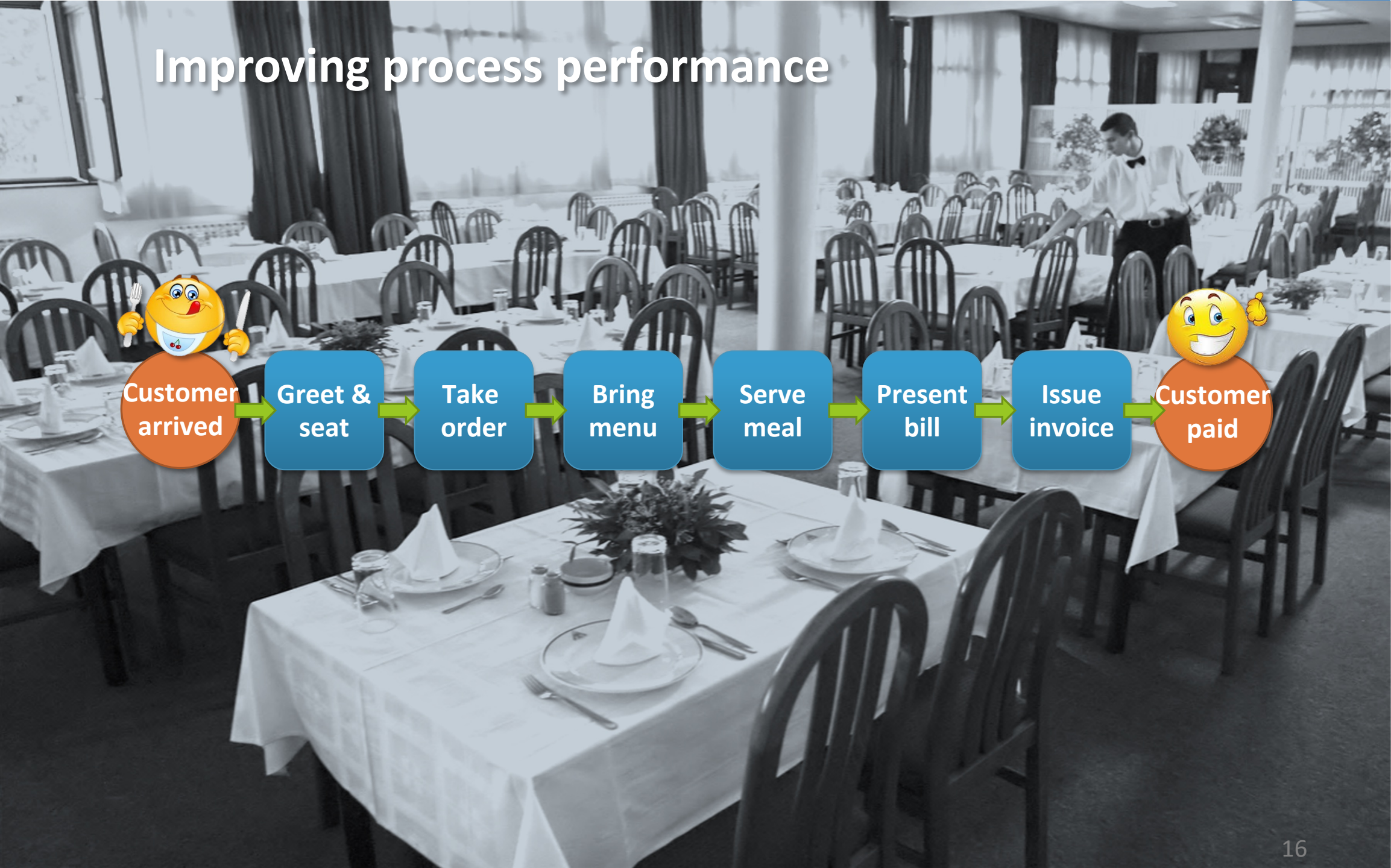
- Faster
- Cheaper
- Better

Process performance

Three dimensions of process performance

- Time
- Cost
- Quality

Improving process performance



How would you improve this process?

Outsource to Customer

Standardize

Eliminate Cooking

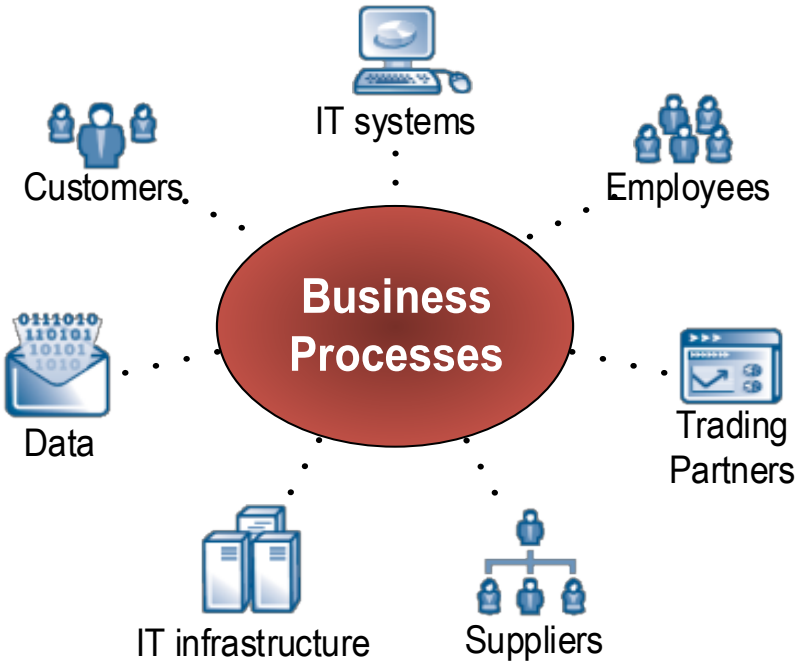
Automate

Invest and Build

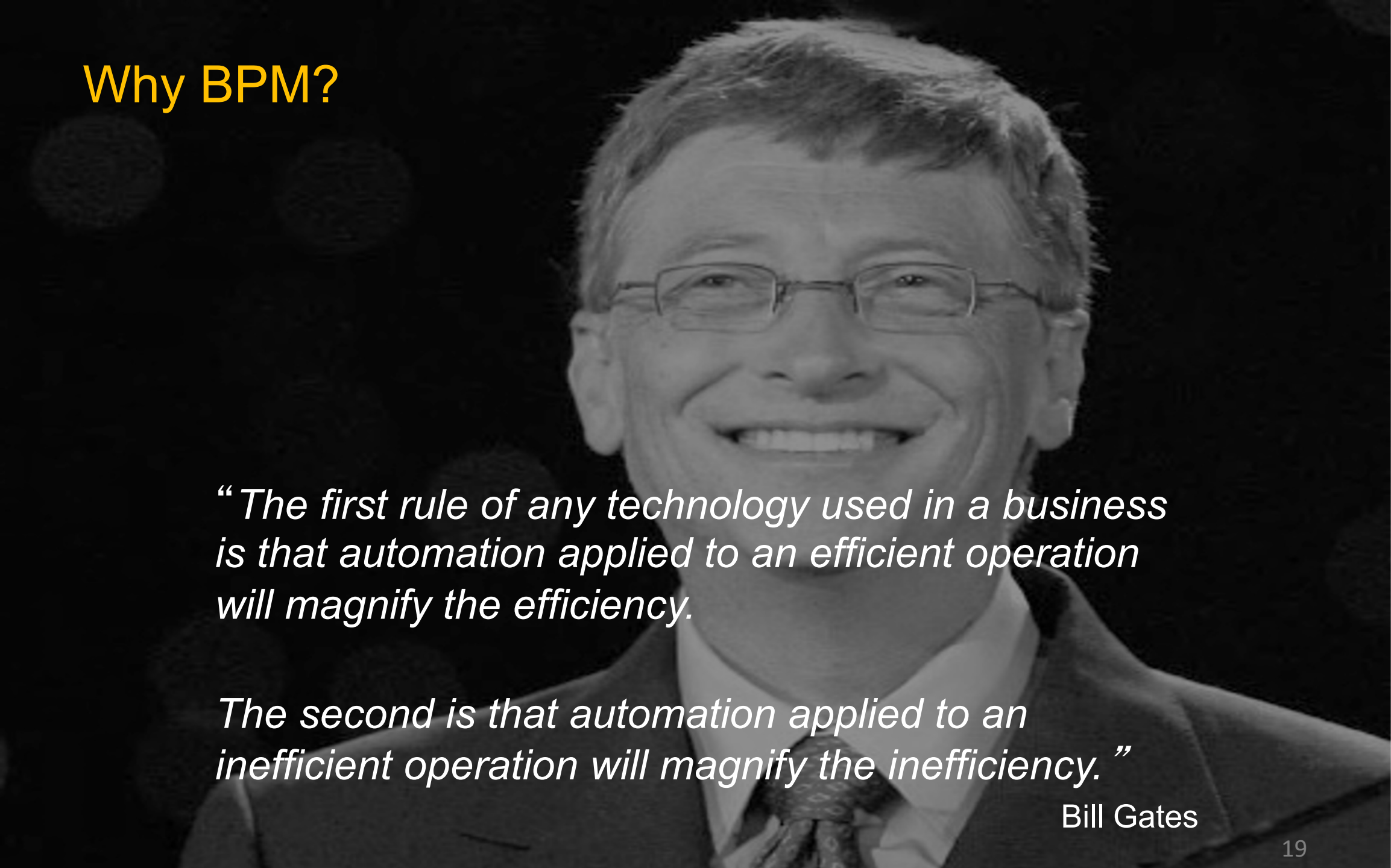
Re-sequence

Business Process Management (BPM)

Body of principles, methods and tools to design, analyze, execute and monitor business processes, with the aim of improving their performance.



Why BPM?

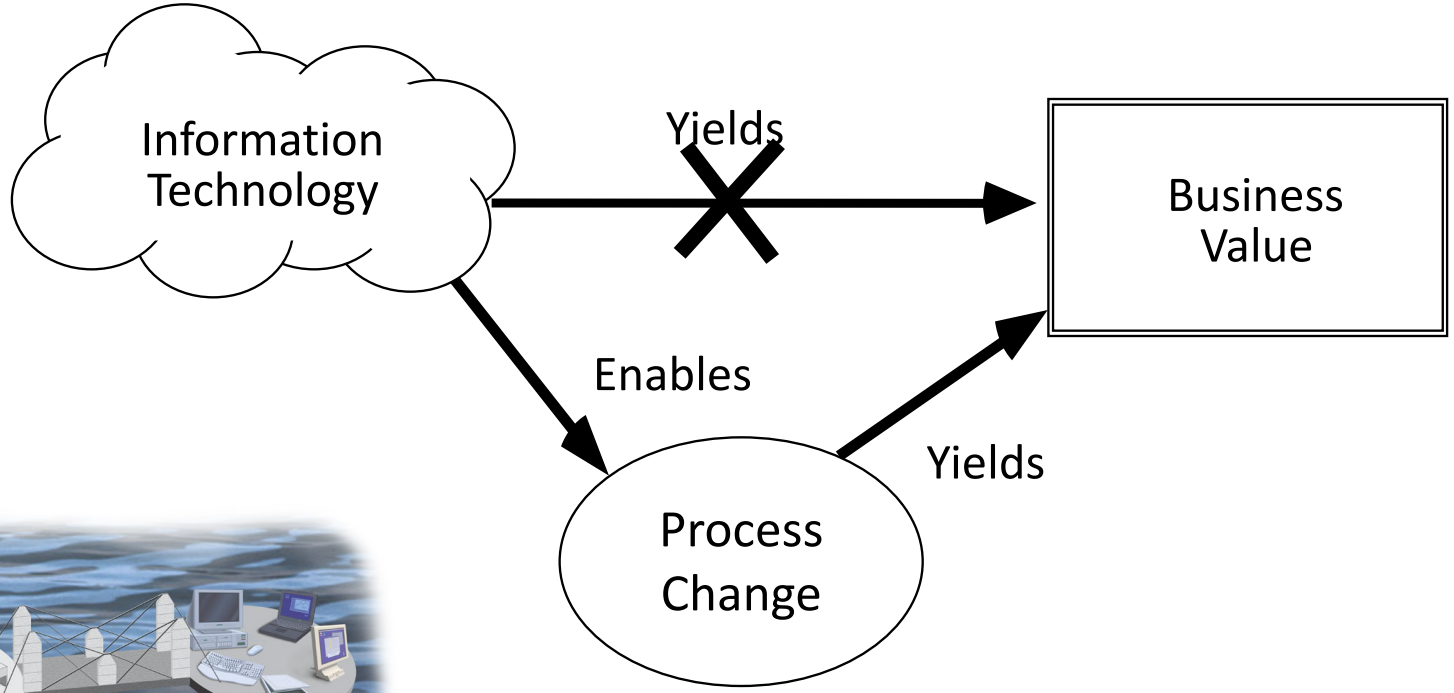


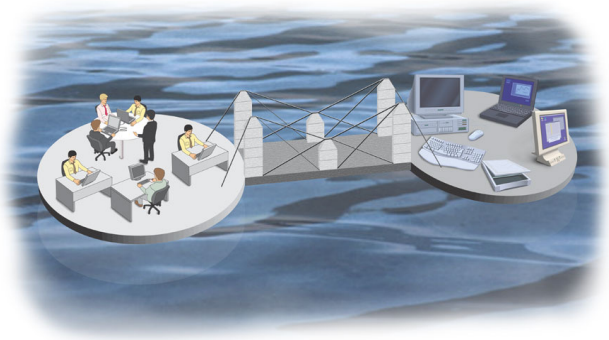
"The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency.

The second is that automation applied to an inefficient operation will magnify the inefficiency."

Bill Gates

In other words…

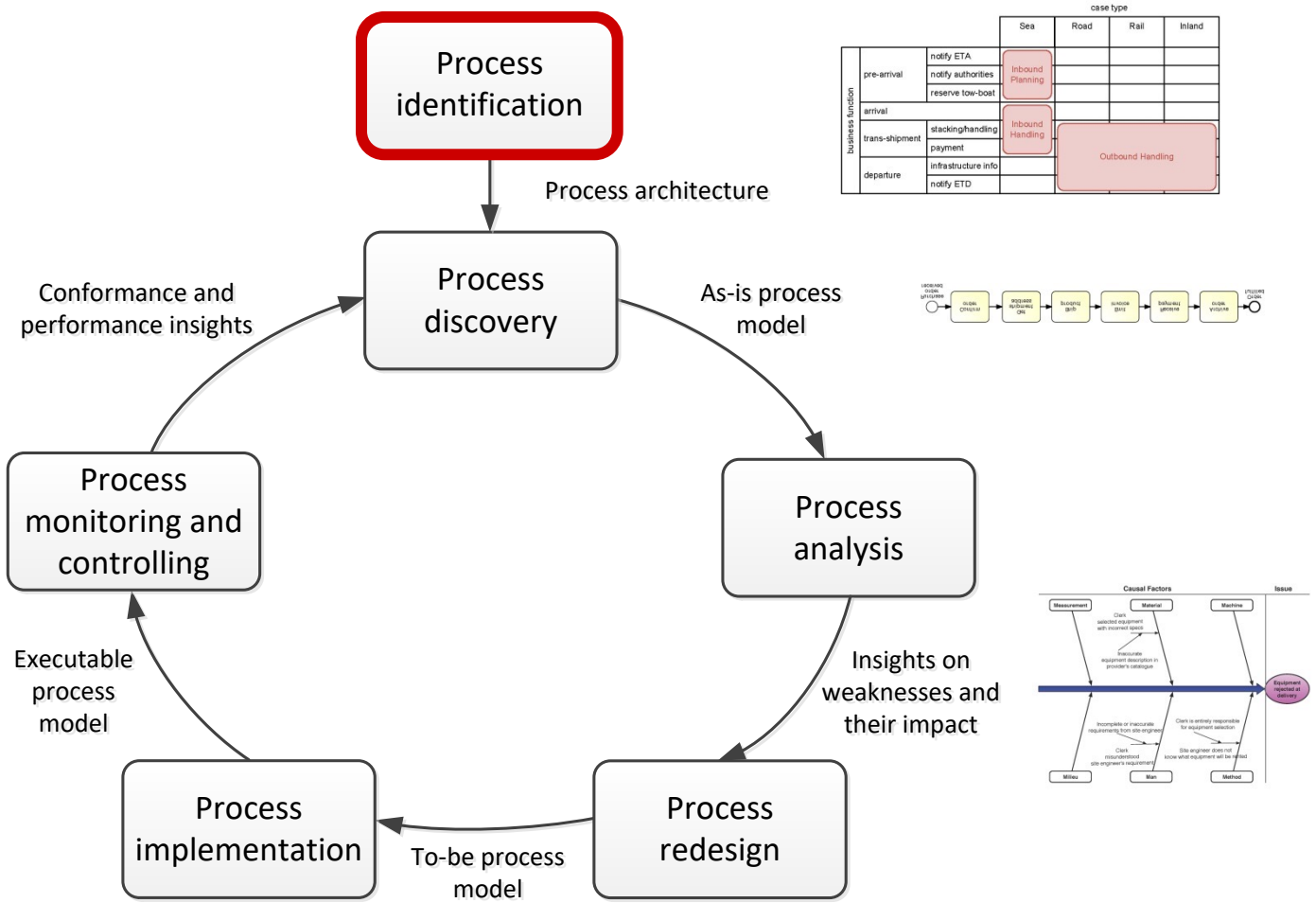




Index Group (1982)

How to go about BPM?

The BPM lifecycle



Process identification steps

1. Designation step
   - Enumerate main processes
   - Determine process scope

→ Process Architecture

2. Prioritization step (aka Process selection)

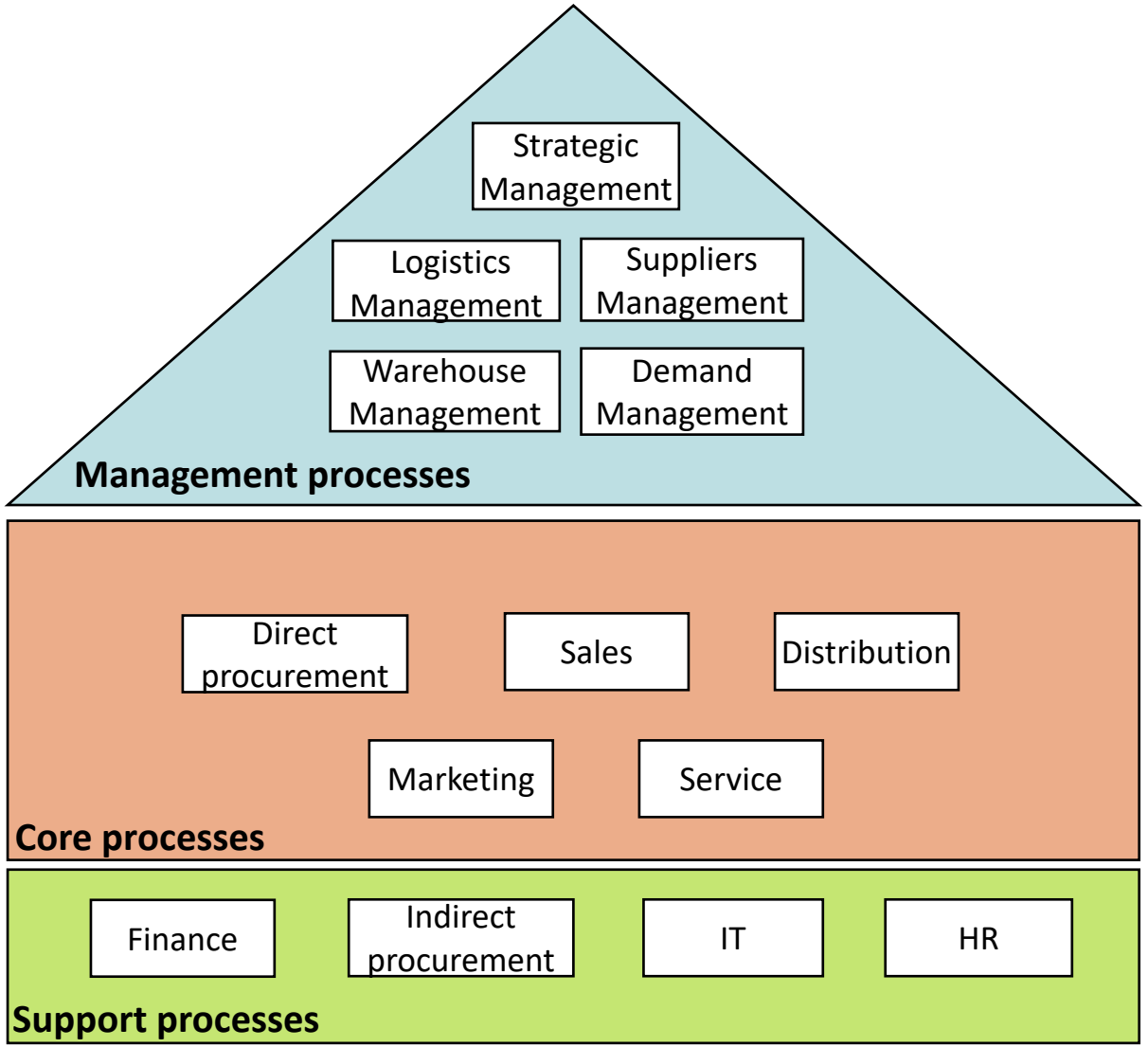
   Prioritize processes based on:
   - Importance
   - Health
   - Feasibility

→ Prioritized Process Portfolio

After Davenport (1993)

Example: process architecture

Wholesaler



Prioritization (aka Process Selection)

1. **Importance**

   Which processes have greatest impact on the organization‘s strategic objectives?

2. **Health (or Dysfunction)**

   Which processes are in deepest trouble?

3. **Feasibility**

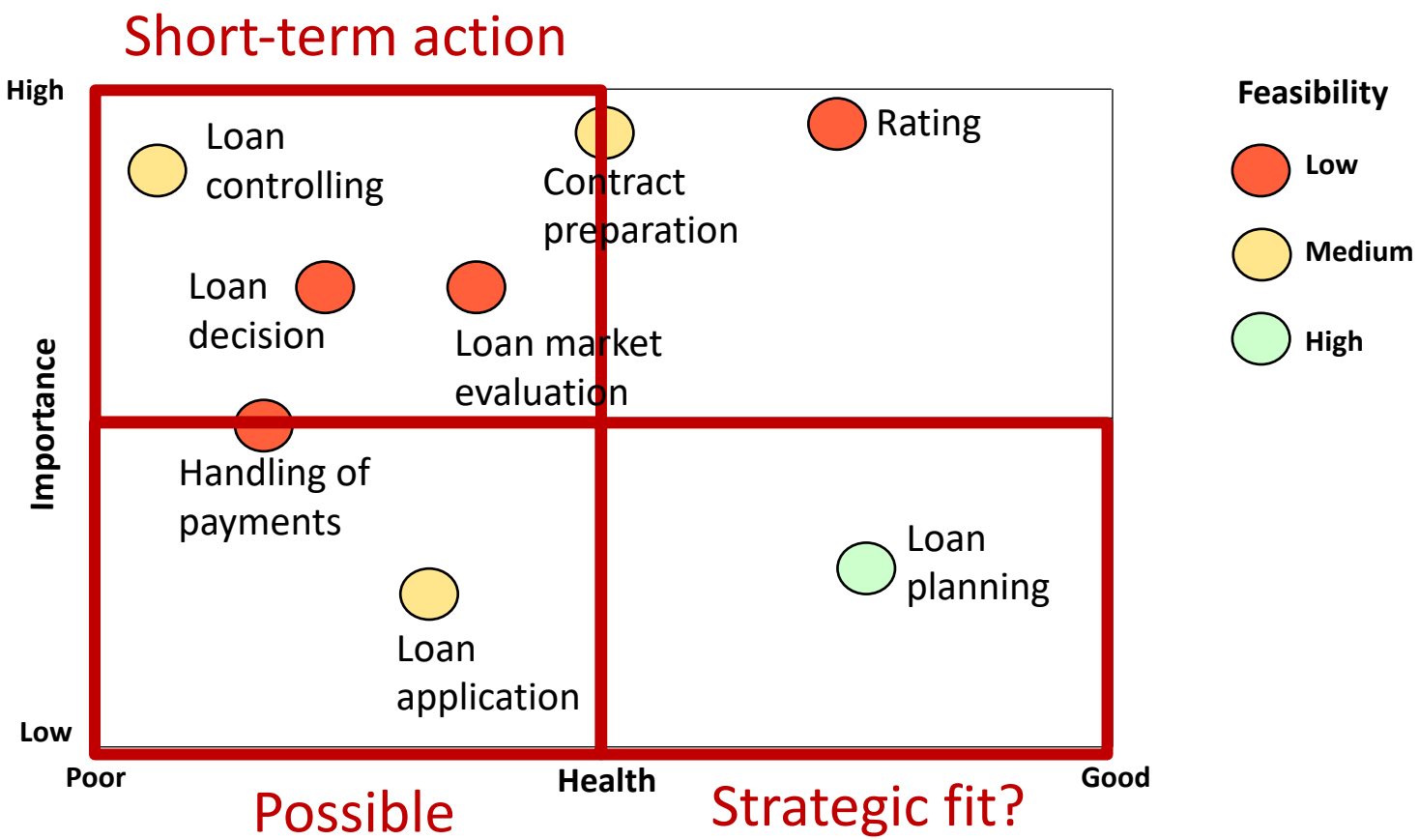
   Which processes are most susceptible to successful process management?

Prioritized process portfolio

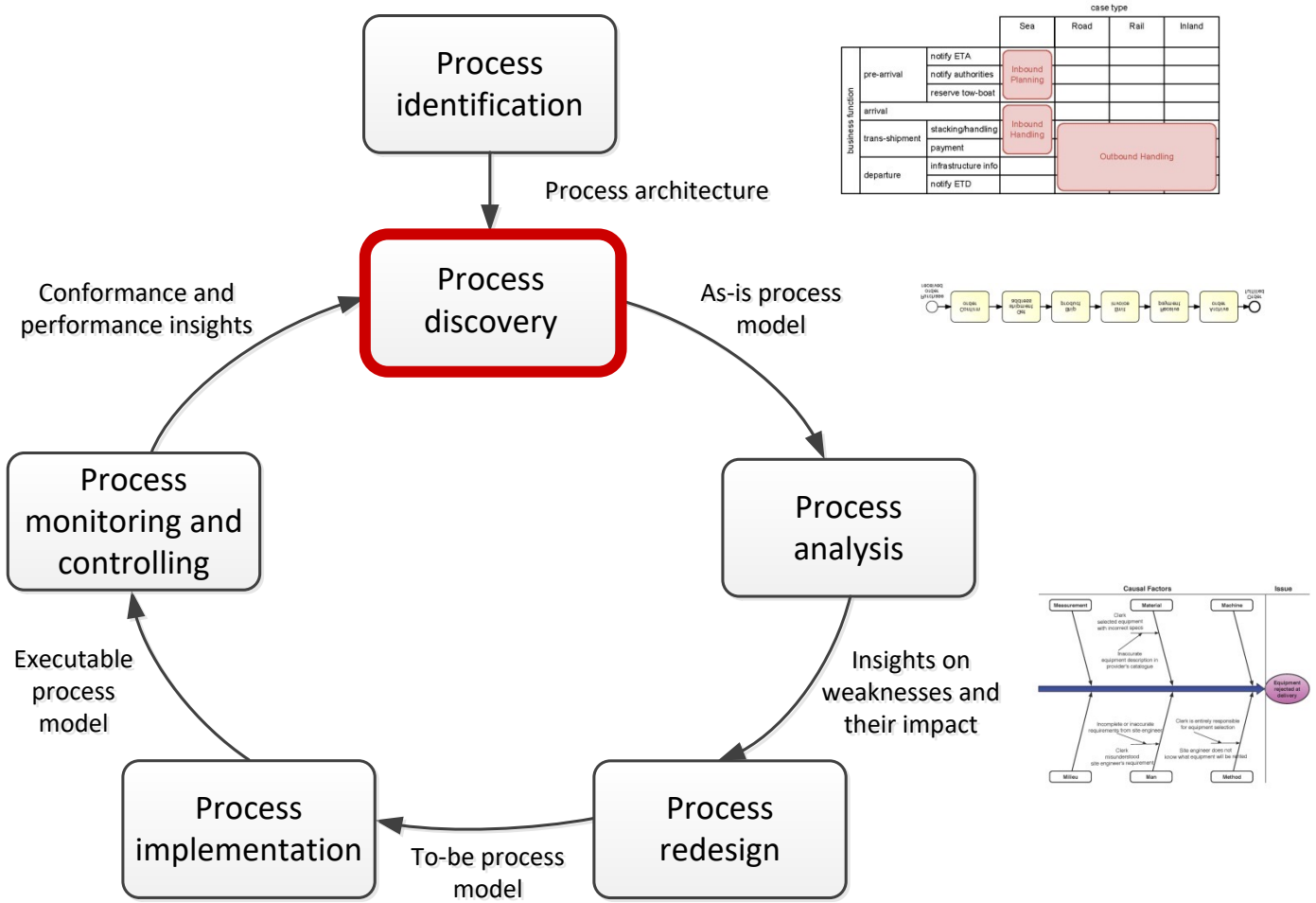
Hammer, Champy (1993)

Example: prioritized process portfolio

Financial institution

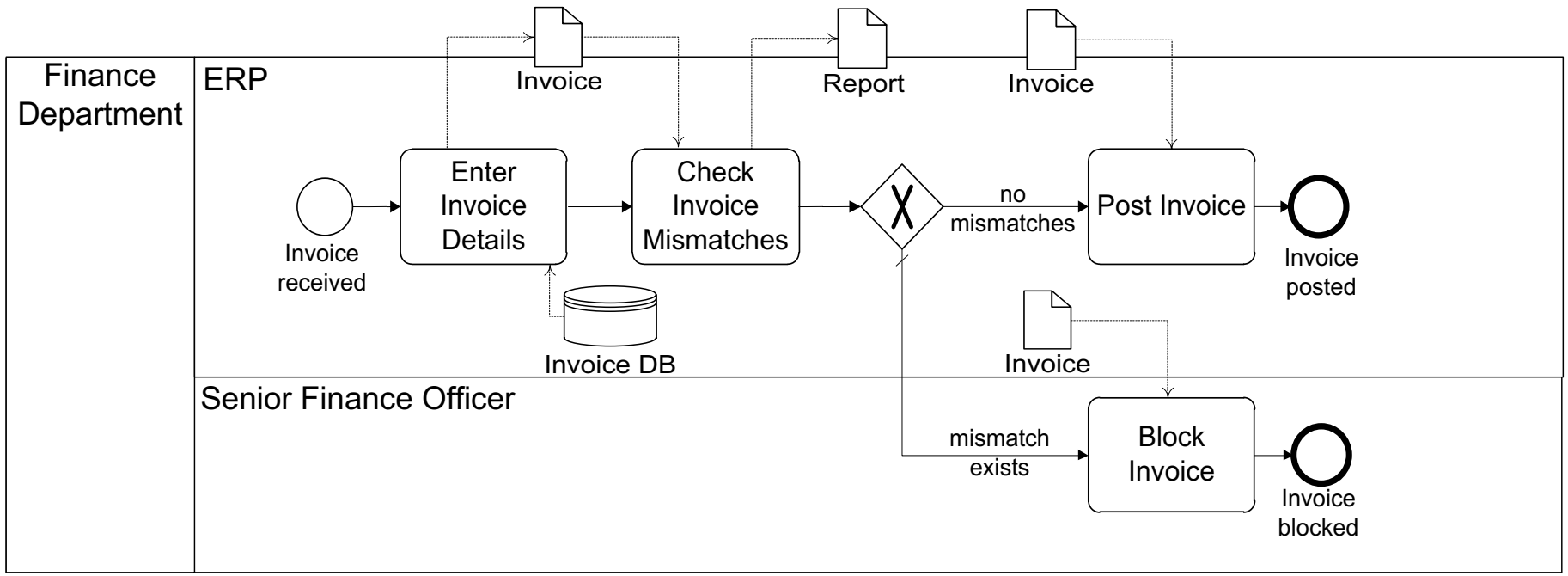


The BPM lifecycle

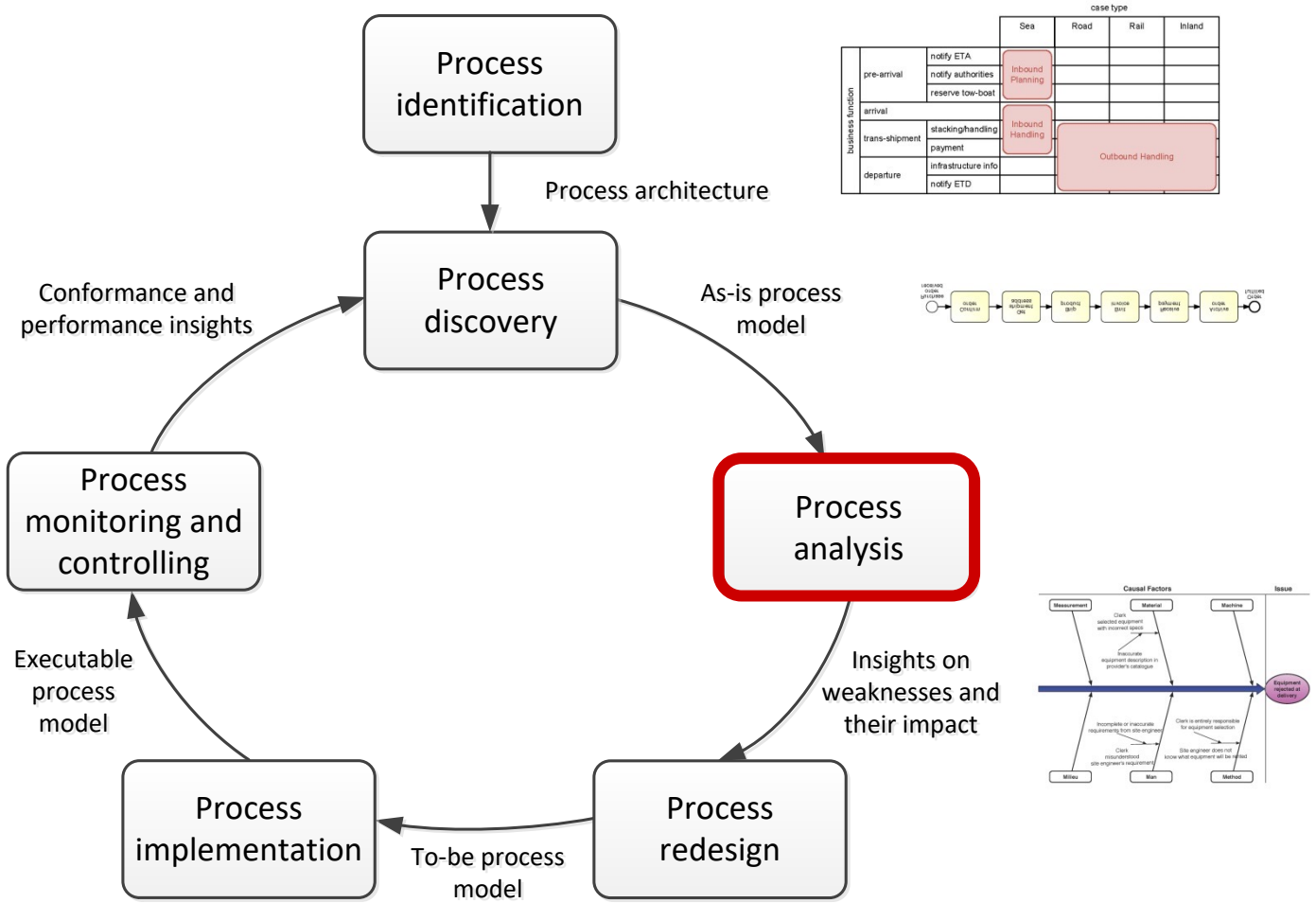


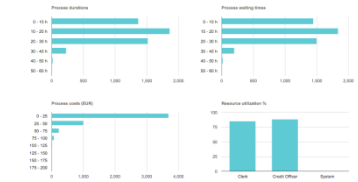
Business process model

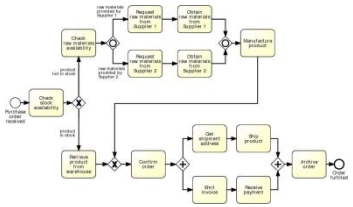
Invoice handling example

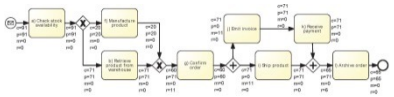


The BPM lifecycle



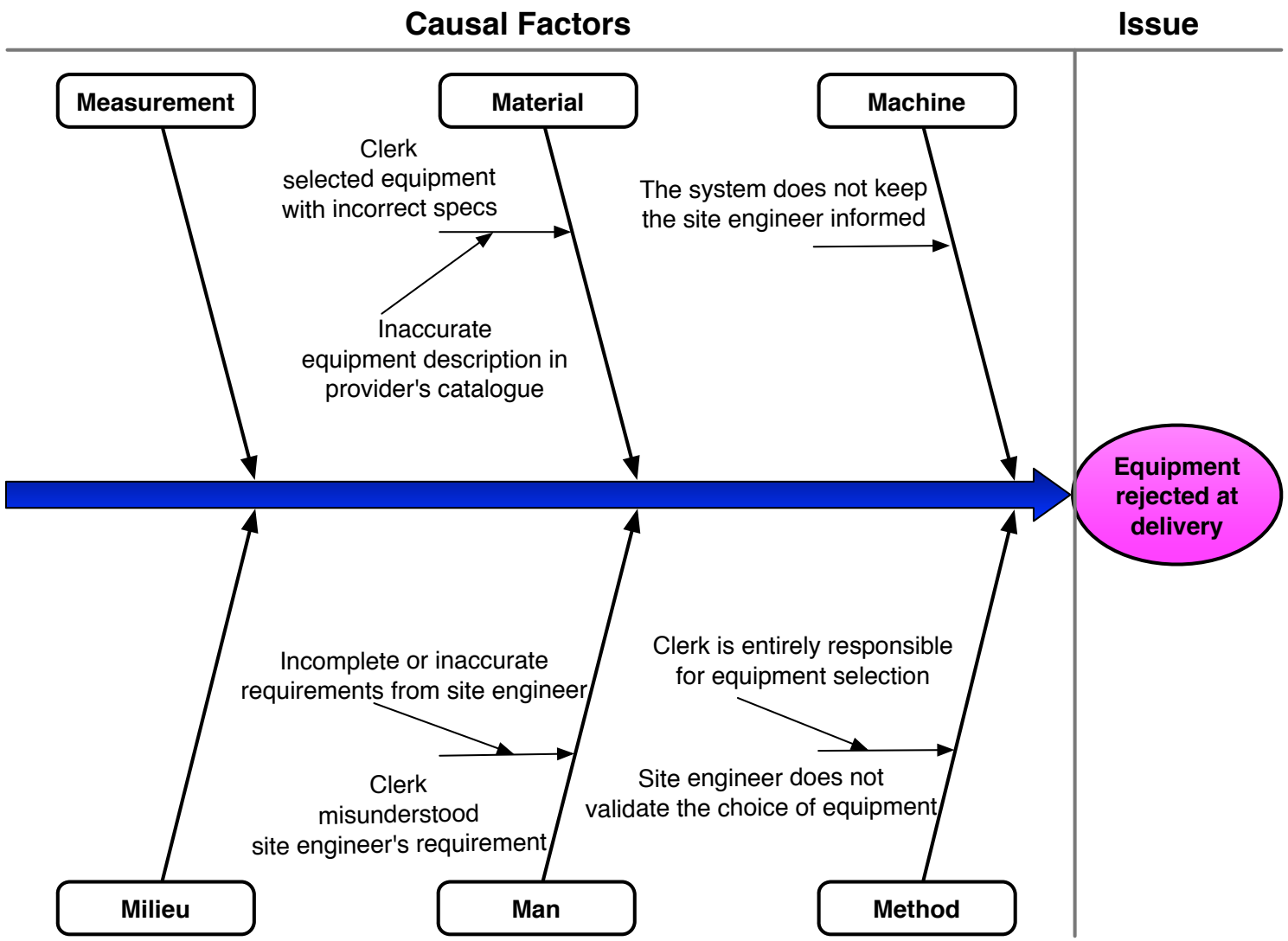






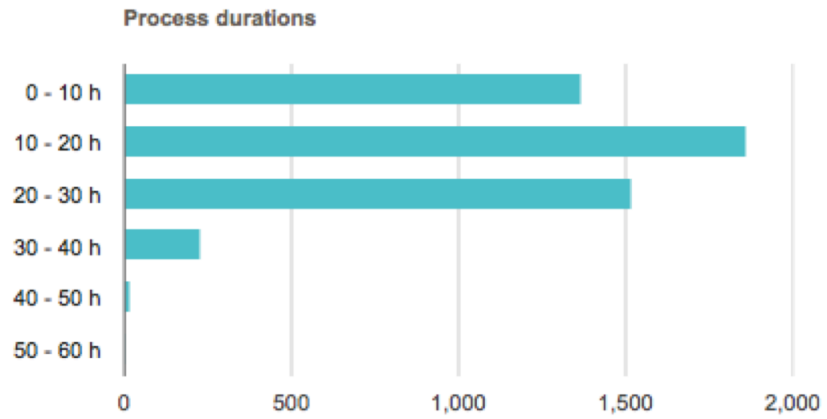
Qualitative process analysis

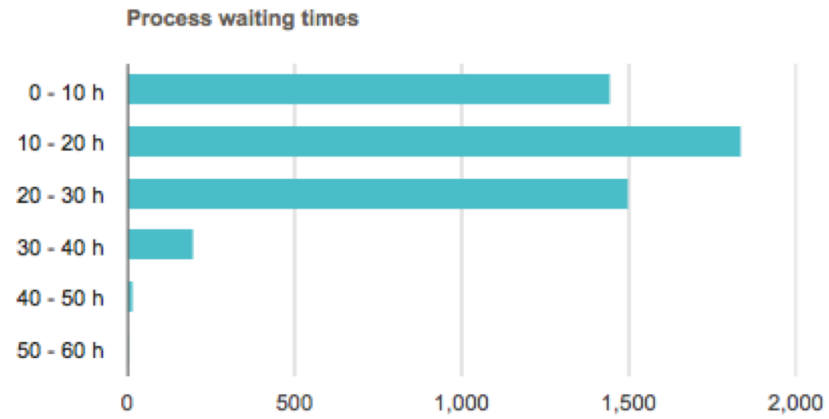
Root-cause analysis example

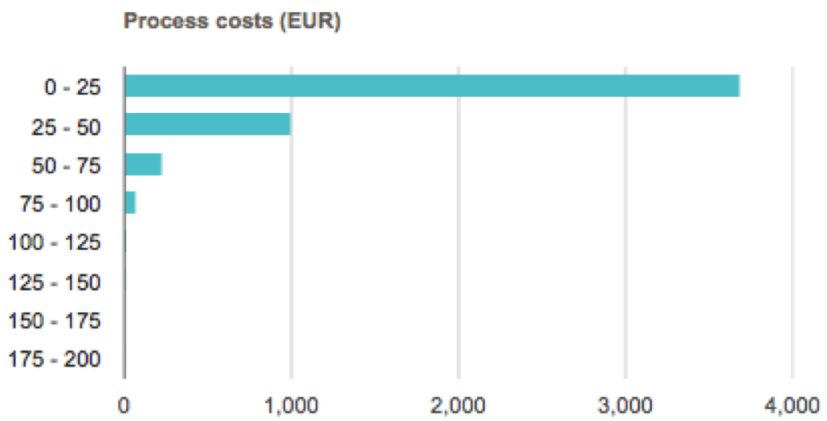


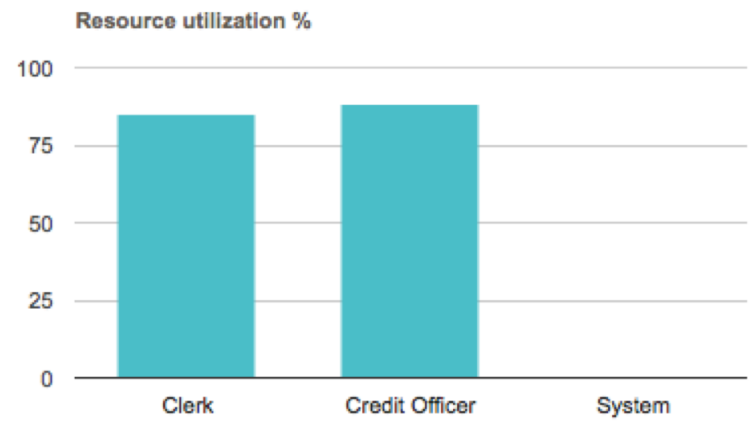
Quantitative process analysis

Process simulation example

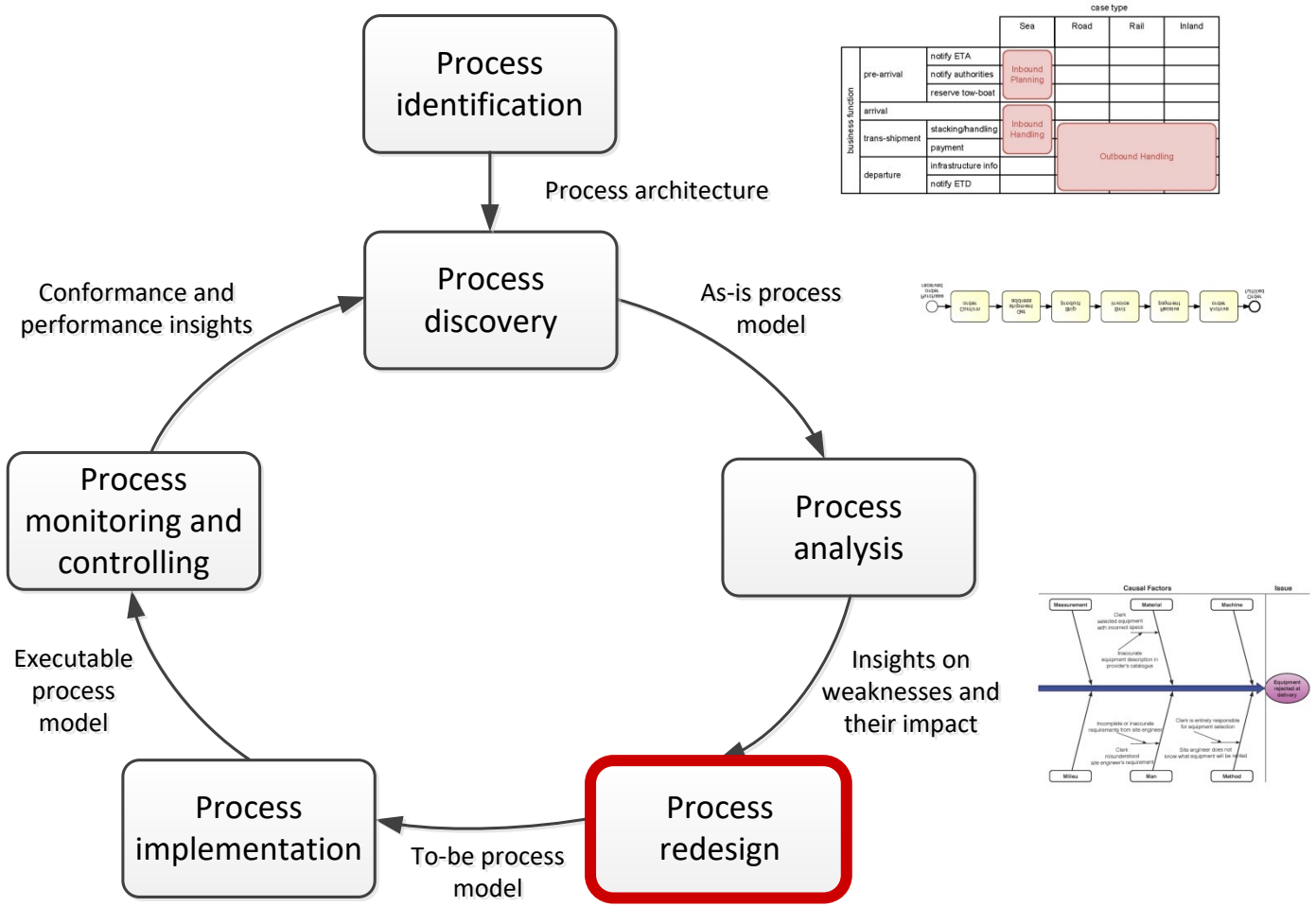


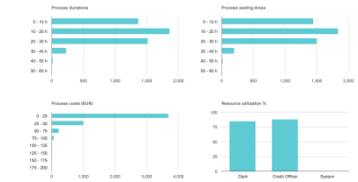


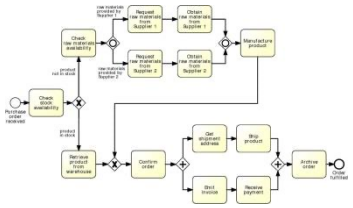


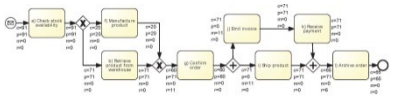


The BPM lifecycle



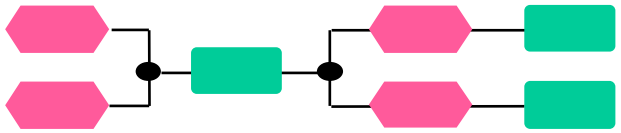




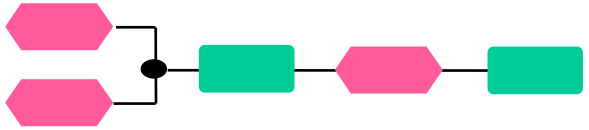


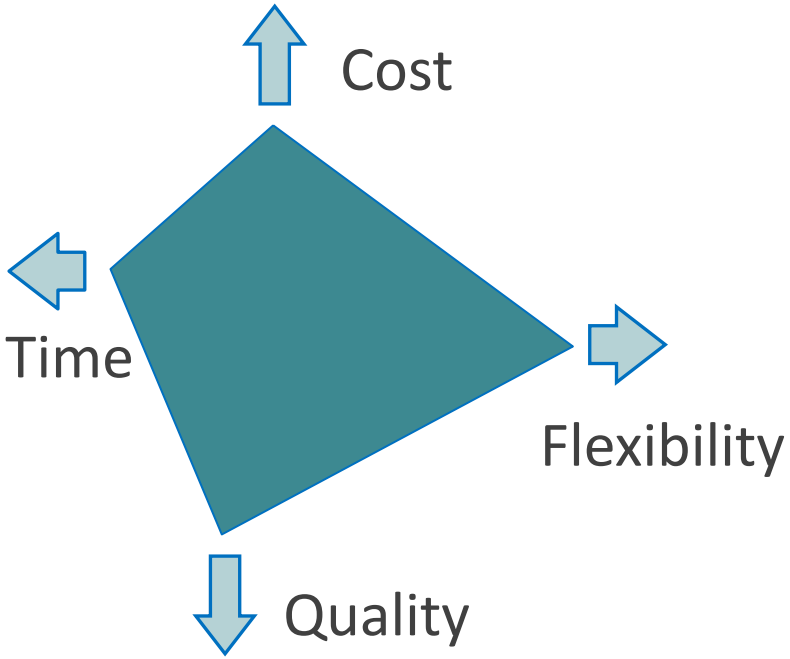
Process redesign

AS-IS process model

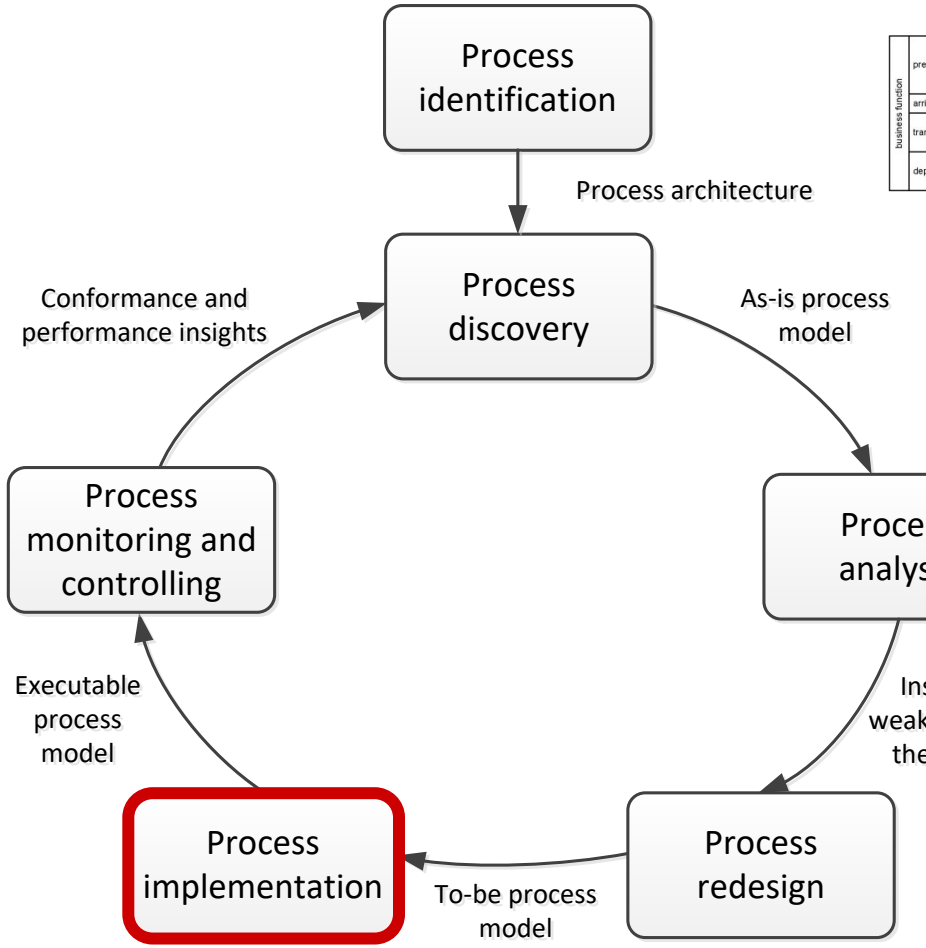


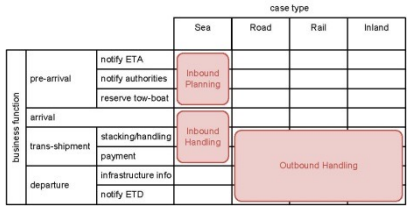
TO-BE process model

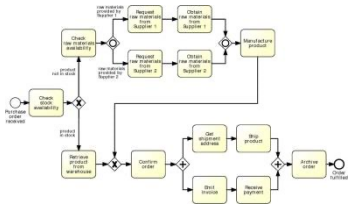


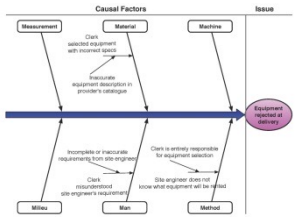


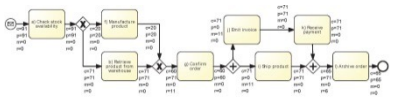
The BPM lifecycle



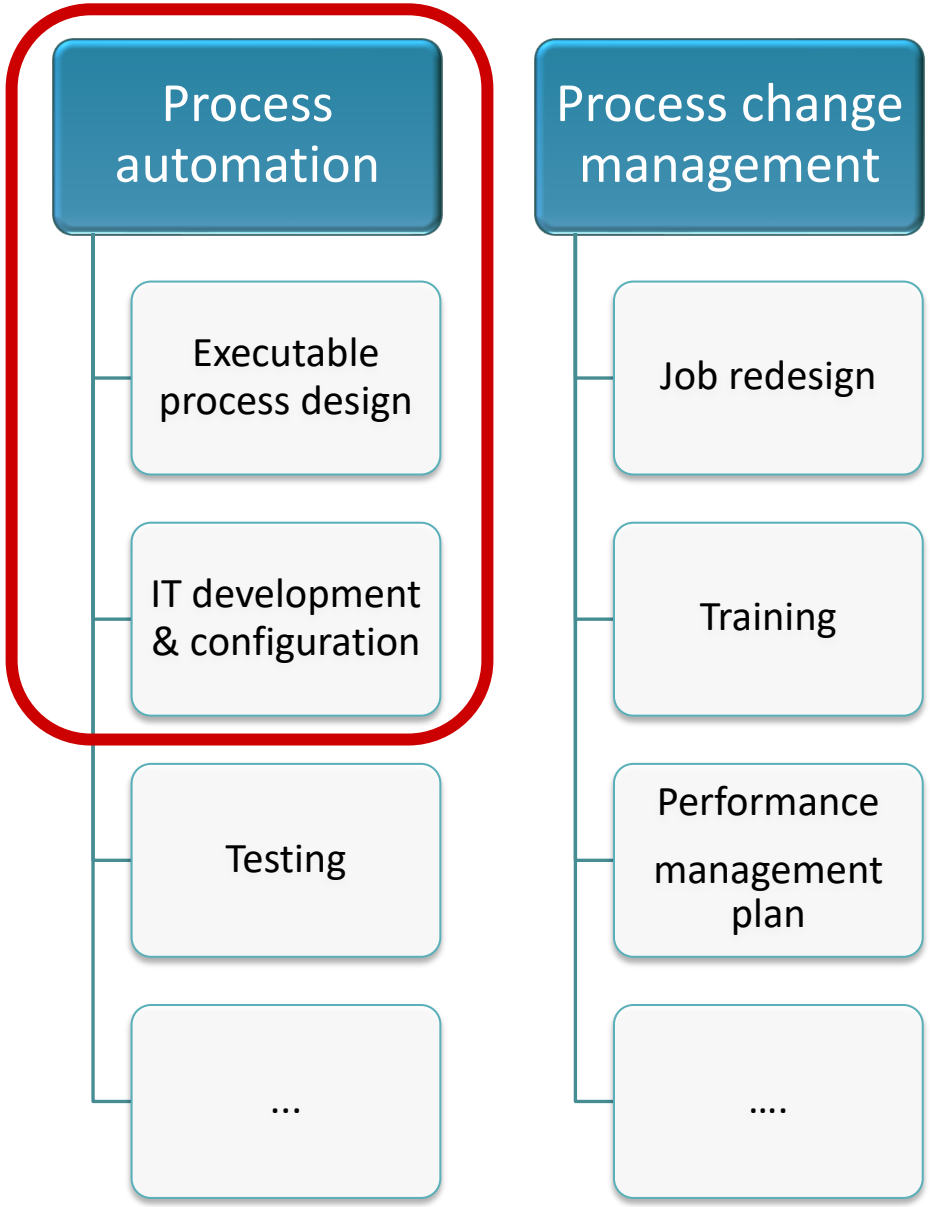




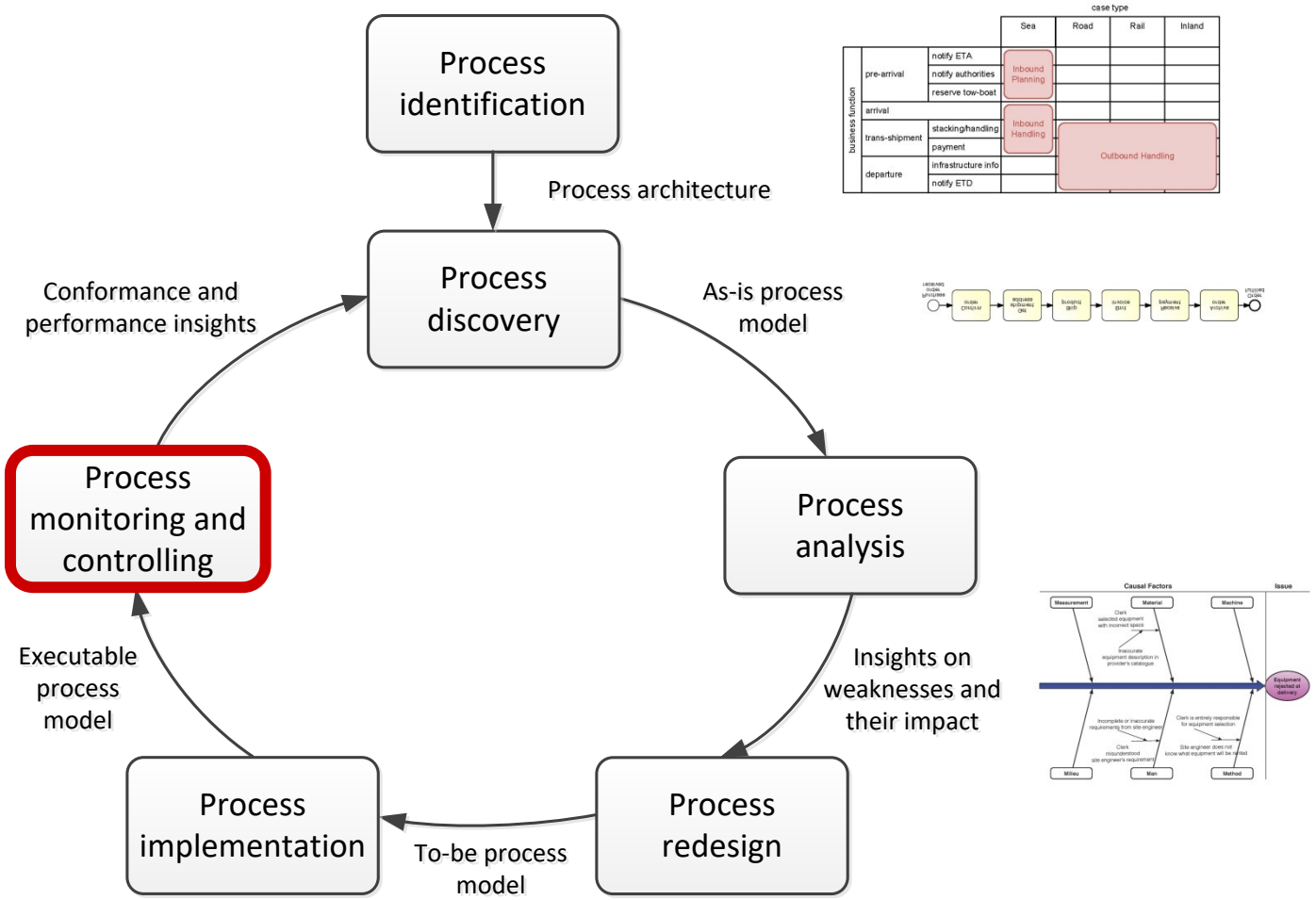




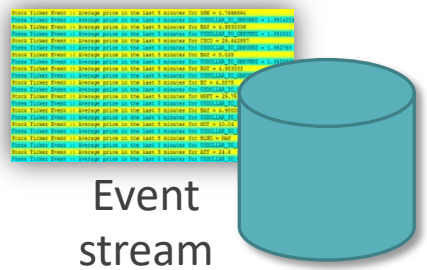
Process implementation



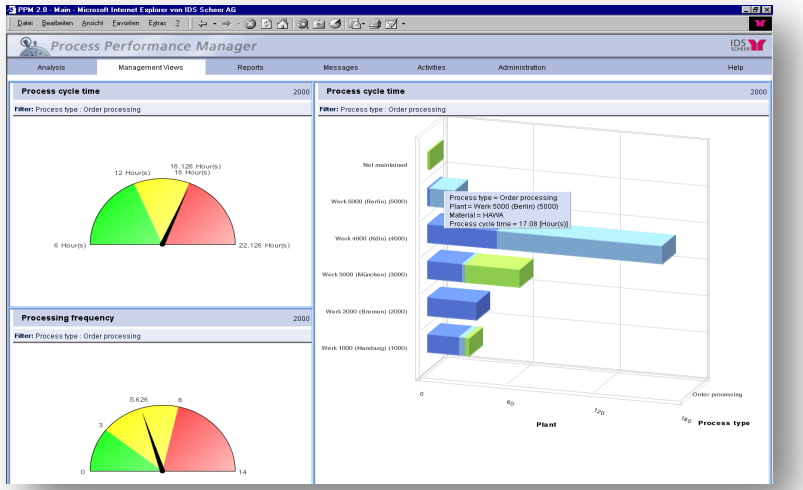
The BPM lifecycle



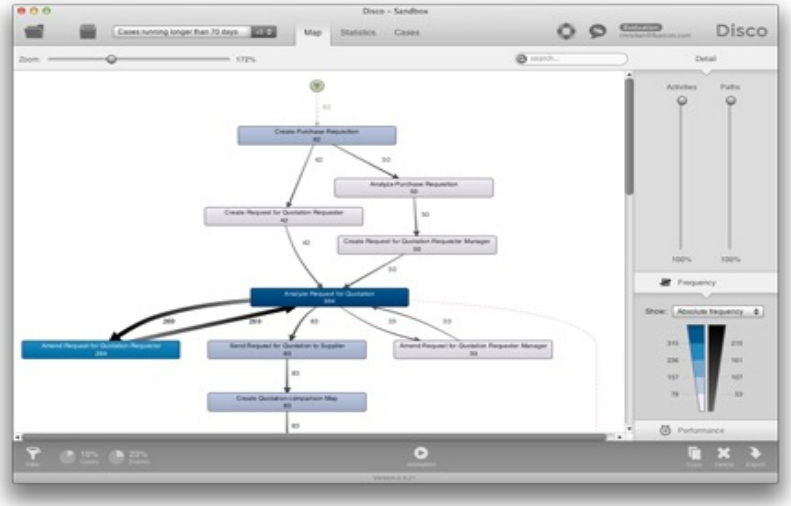
Process monitoring



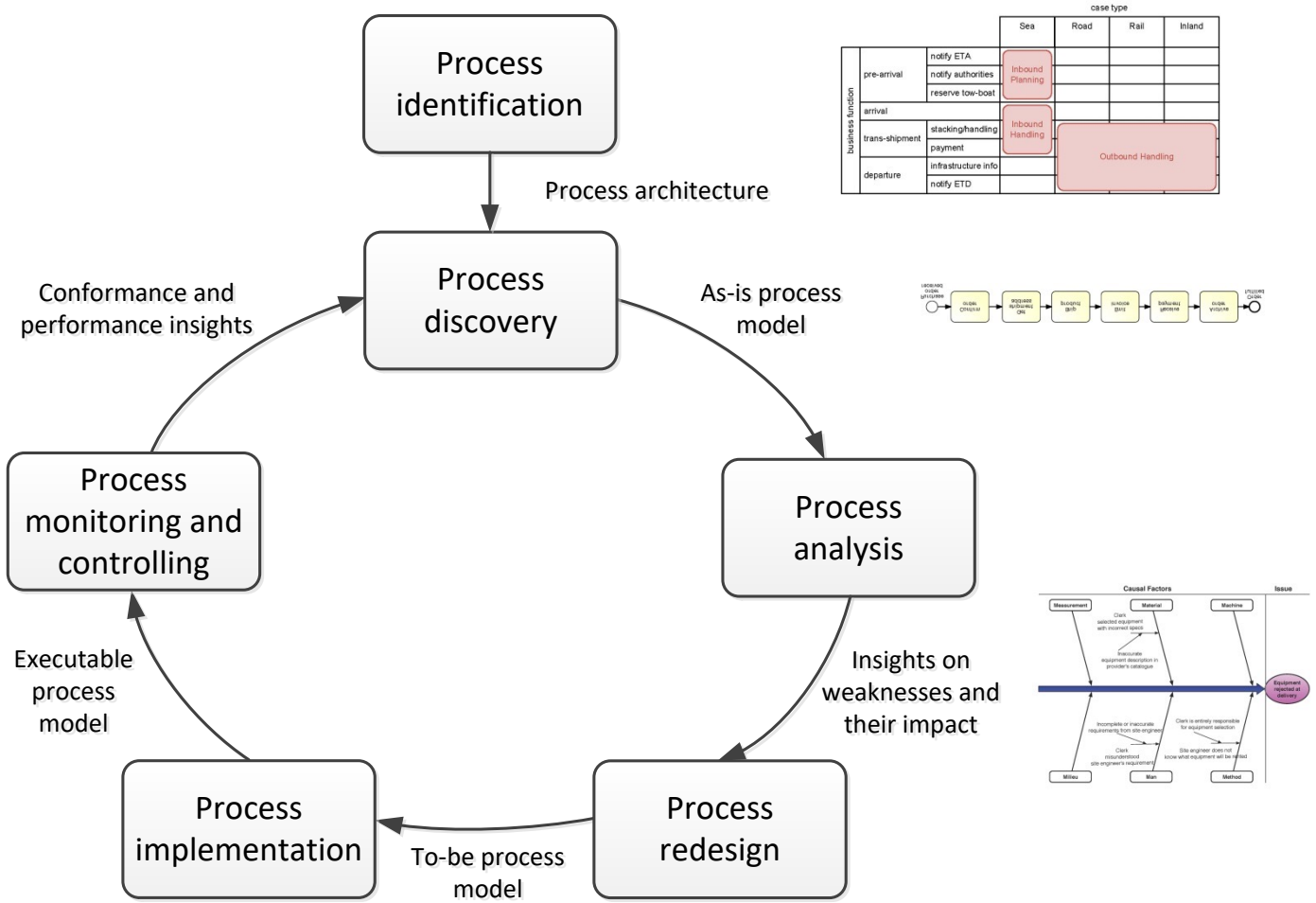
Dashboards, alerts & reports



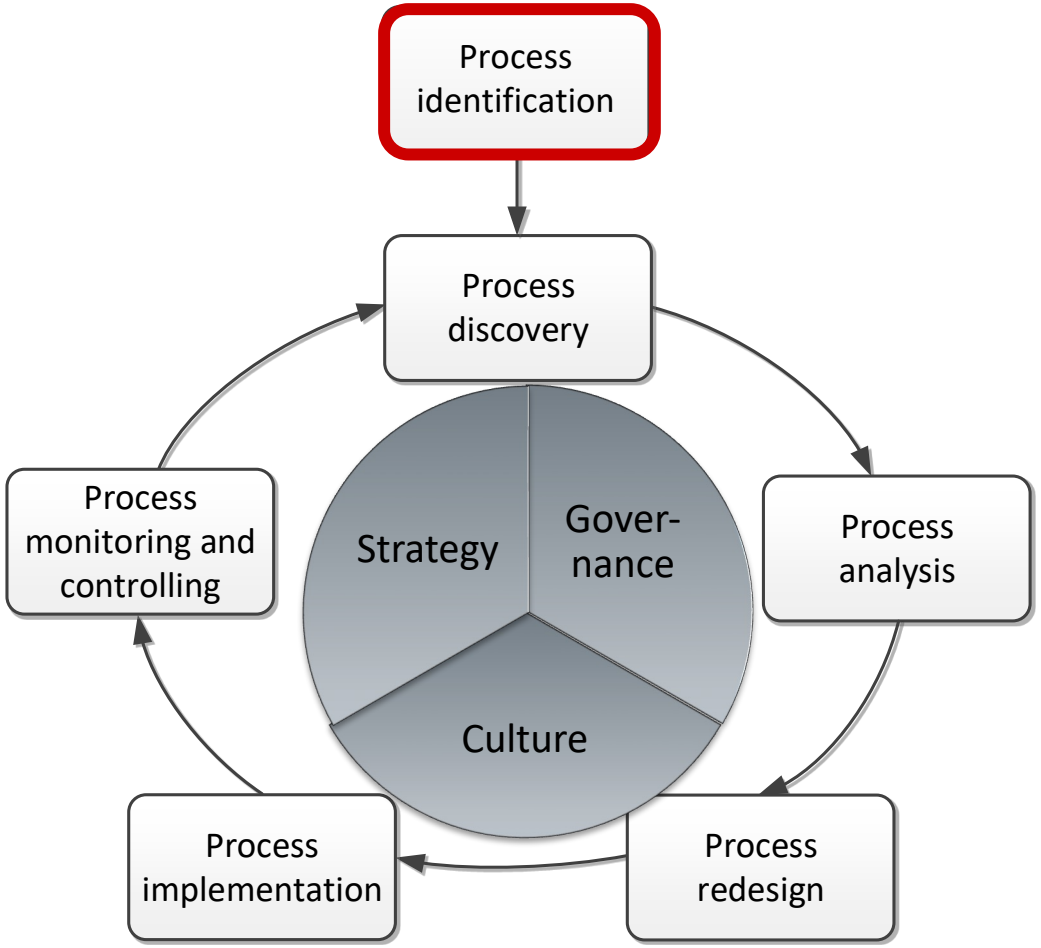
Model-based analytics (p. mining)



The BPM lifecycle



The BPM lifecycle



Process identification

What?

1. Identify an organization's business processes
2. Prioritize their management based on certain criteria

Why?

1. Understand the organization
2. Maximize value of BPM projects

Process identification steps

1. Designation step
   - Enumerate main processes
   - Determine process scope

→ Process Architecture

2. Prioritization step (aka Process selection)

   Prioritize processes based on:
   - Importance
   - Health
   - Feasibility

→ Prioritized Process Portfolio

After Davenport (1993)

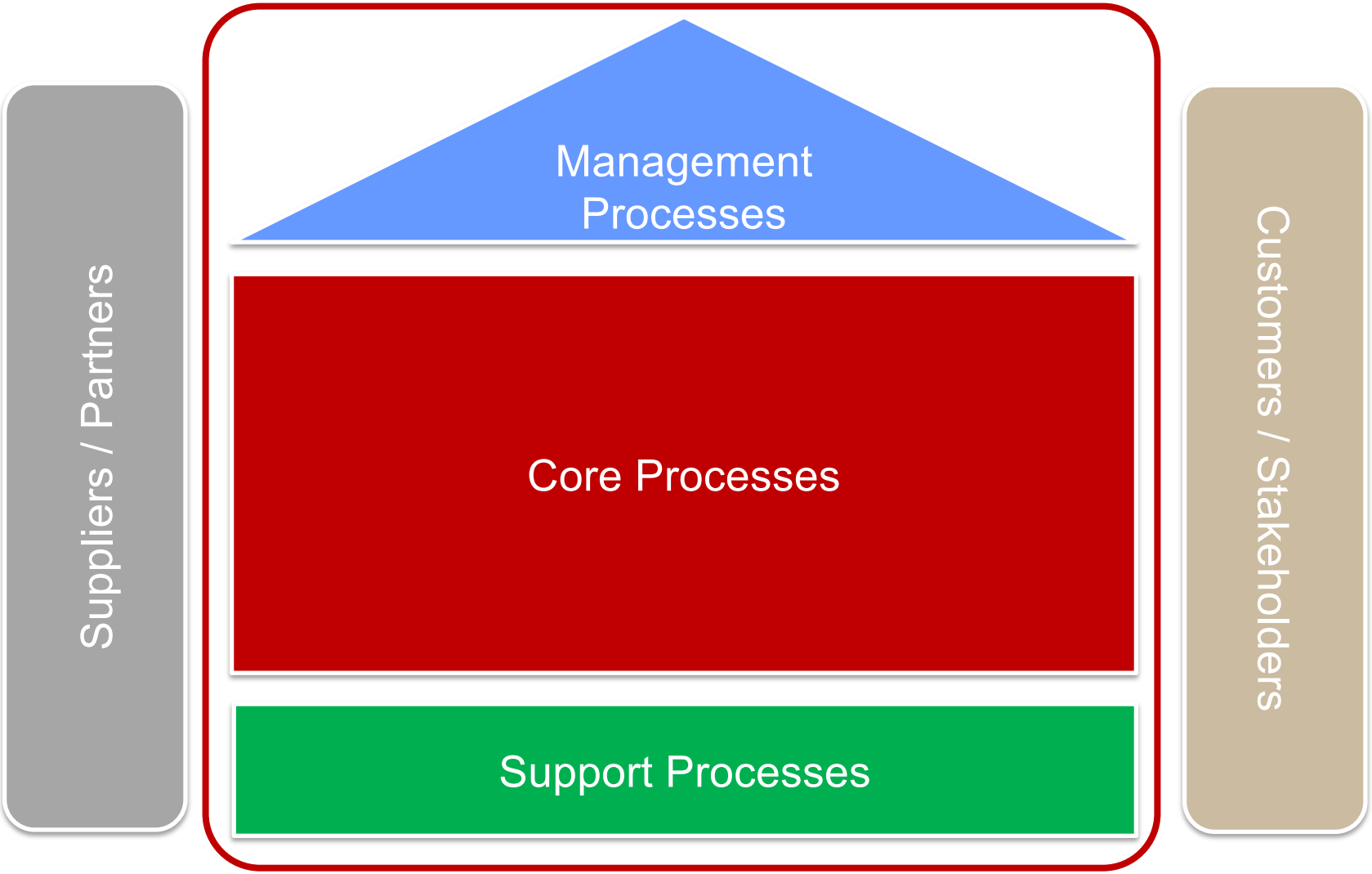
Process Enumeration

"Most businesses have just three core processes:

1. Sell stuff
2. Deliver stuff
3. Making sure you have stuff to sell and deliver"

Geary Rummler

Porter: Types of processes



After Michael Porter (1985)

Example: core, support and management processes

Wholesaler

Core processes

- Sales (lead-to-quote, quote-to-order, order-to-cash)
- Purchase-to-Pay (direct procurement, e.g. supplies replenishment)
- …

Support processes

- Purchase-to-pay (indirect procurement, e.g. parts replenishment, operational resources replenishment…)
- HR (policies update, recruitment, induction, probation…)
- …

Management processes

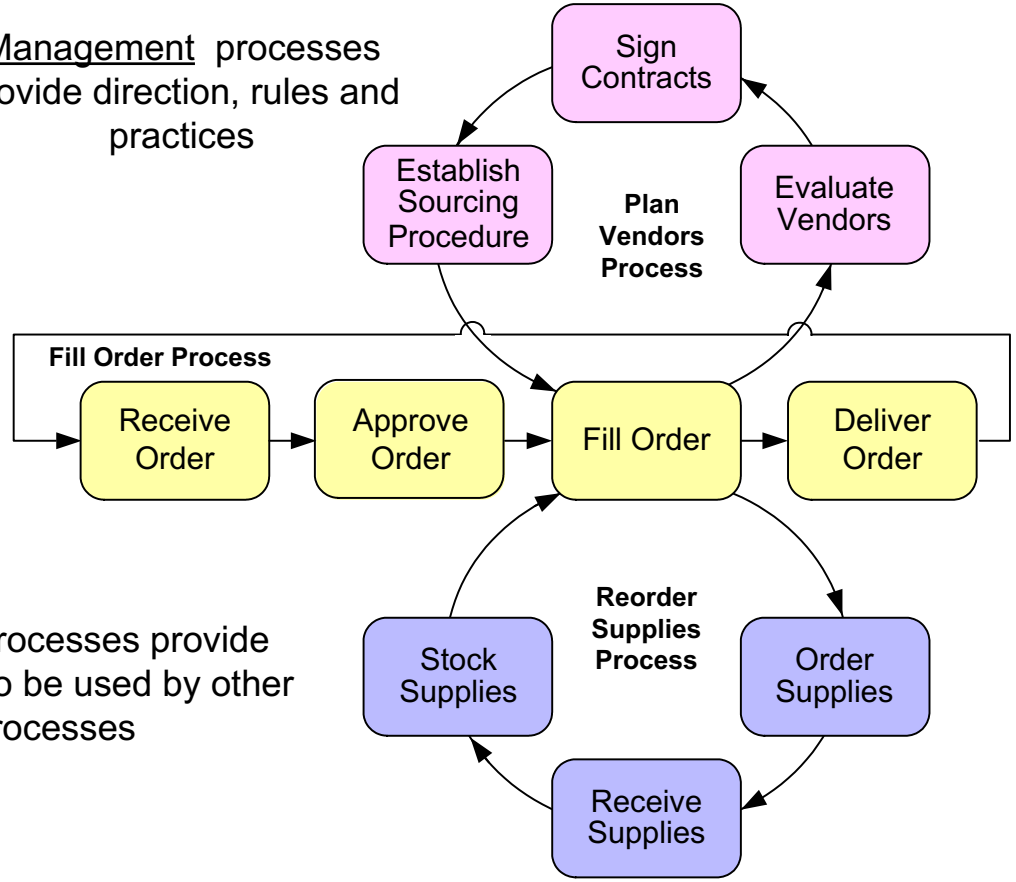
- Suppliers management (suppliers planning, suppliers acquisition…)
- Logistics management (logistics planning, logistics controlling…)
- …

Relations between core, support, mgt processes

<u>Management</u> processes provide direction, rules and practices

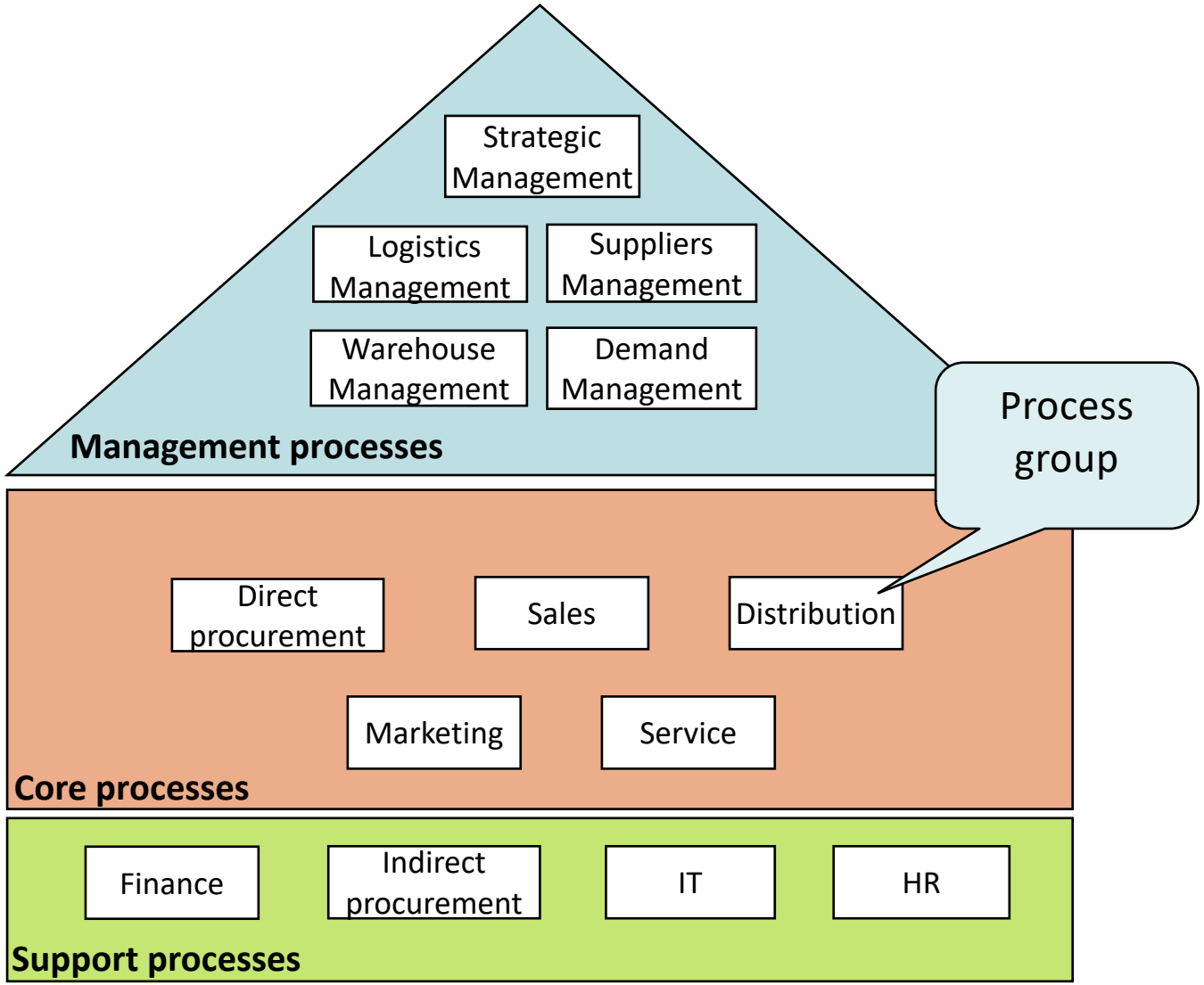
<u>Core</u> processes generate value as they are directly linked to external customers

<u>Support</u> processes provide resources to be used by other processes



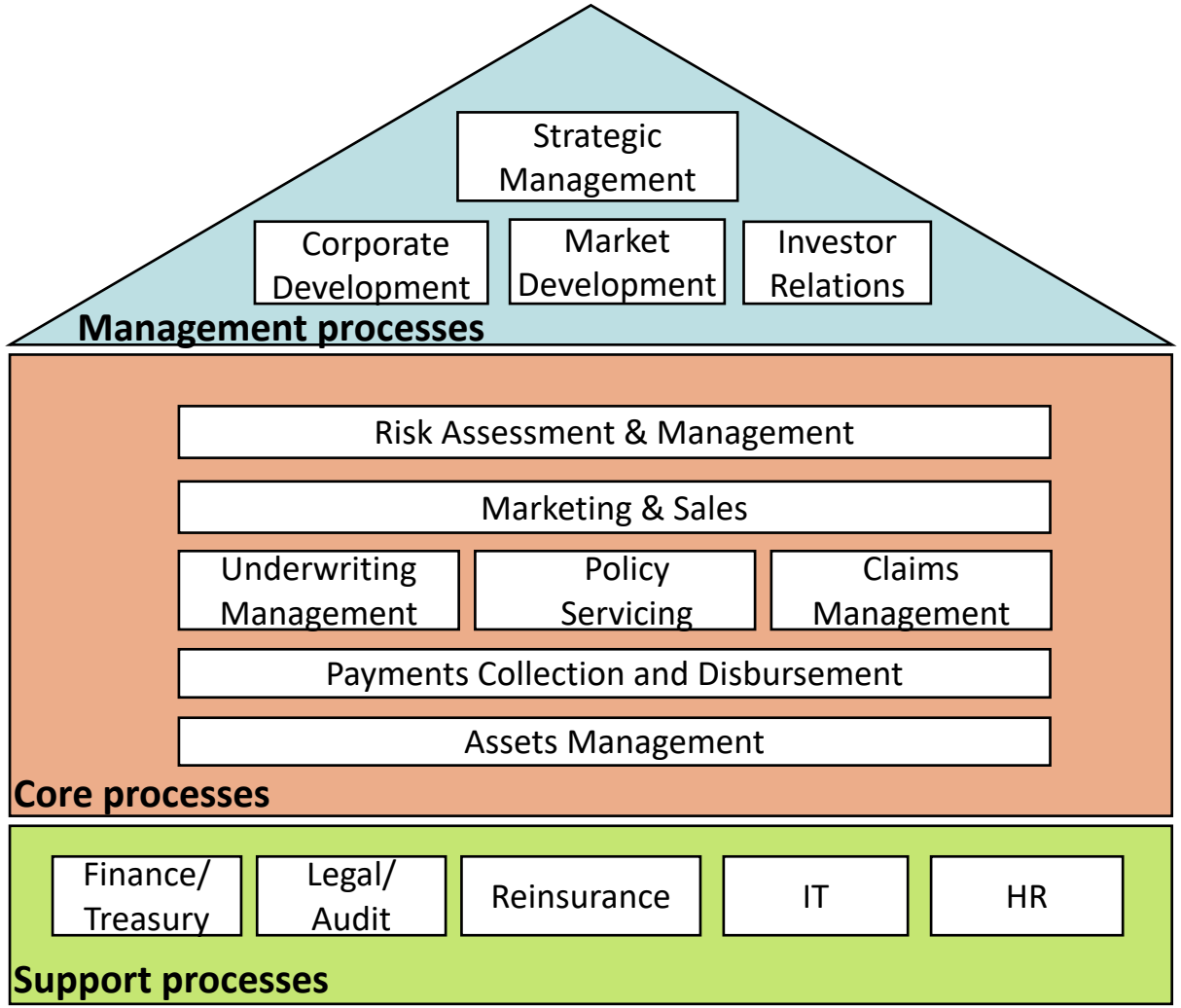
Example: process architecture

Wholesaler



Example: process architecture

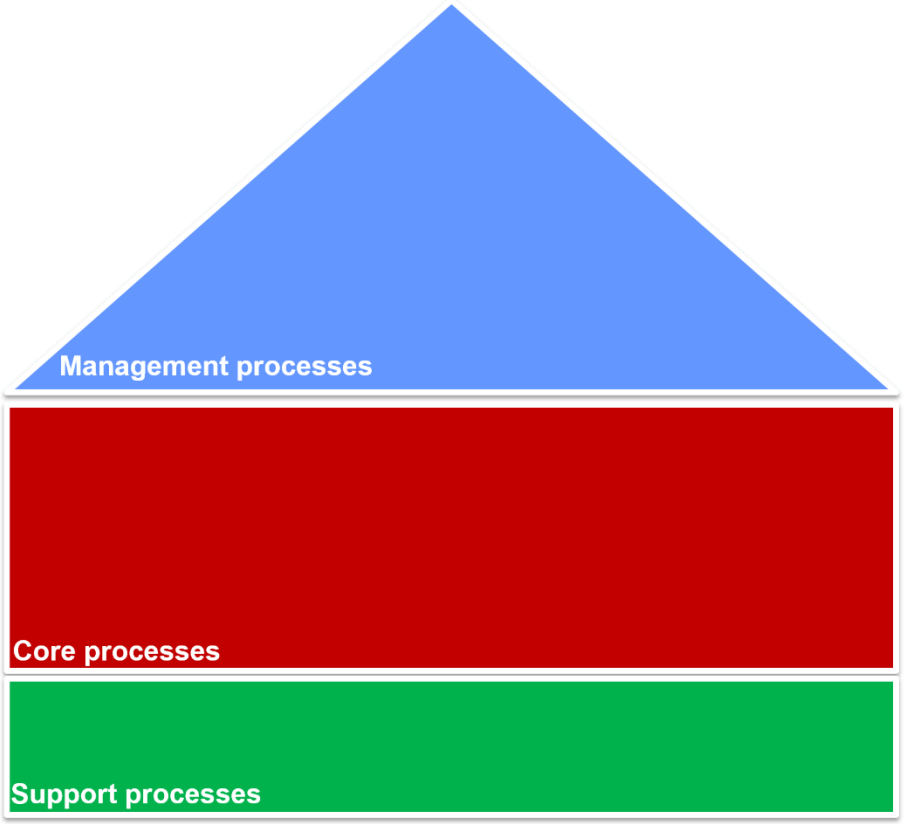
Insurance company



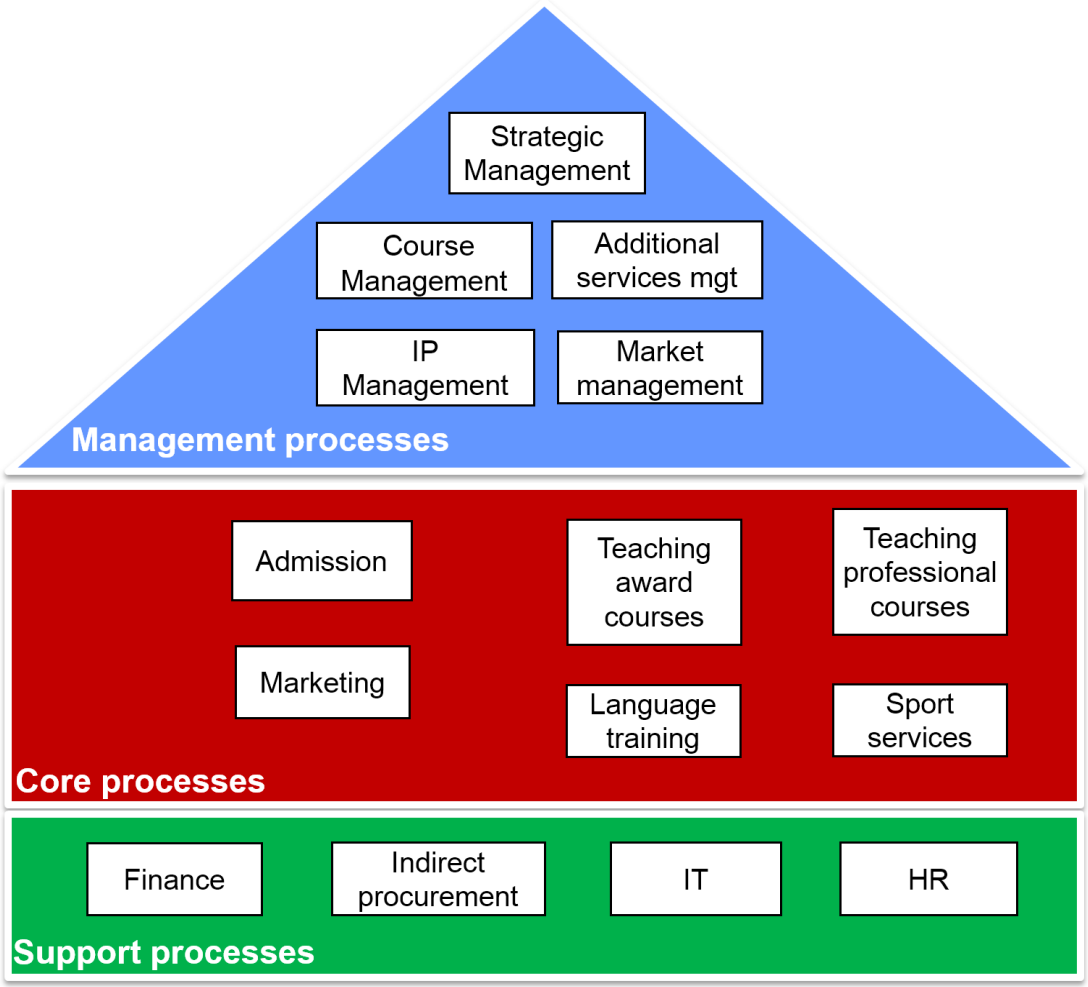
Exercise: classify by process type

These groups of processes are typically performed at a university. Categorize each process group as core, support or management

- Sport services
- HR
- Indirect procurement
- Strategic Management
- Teaching award courses
- IP Management
- Marketing
- Course Management
- Additional services mgt
- Language training
- Admission
- IT
- Market management
- Teaching professional courses



Solution: identify process types



Process scoping

Processes are interdependent → insights into interrelations required

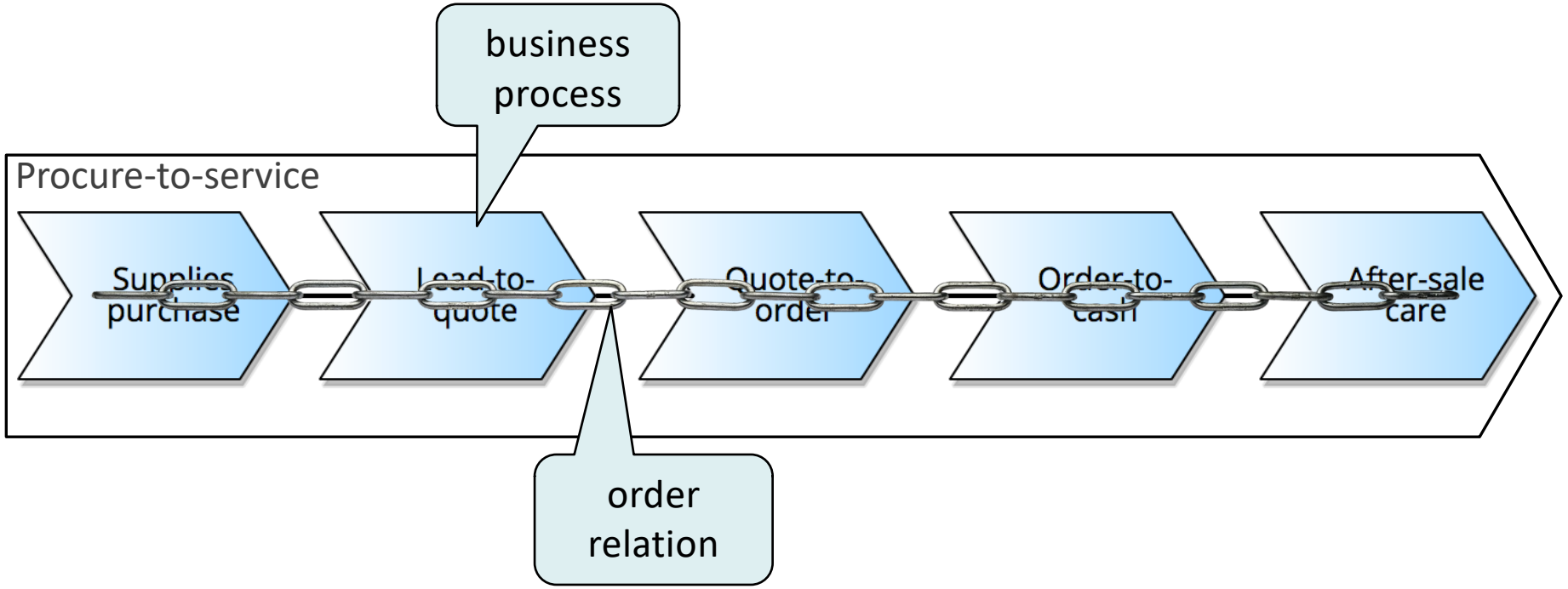
- Specialization: general – special product/service
- Horizontal: upstream – downstream processes and their value chains
- Vertical: main processes – sub-processes

↓

Process architecture

Value chain modeling

- Chain of *processes* an organization performs to deliver value to customers and stakeholders
- More generally, a mechanism to group high-level business processes according to an order relation (can be applied to core, support and management processes)



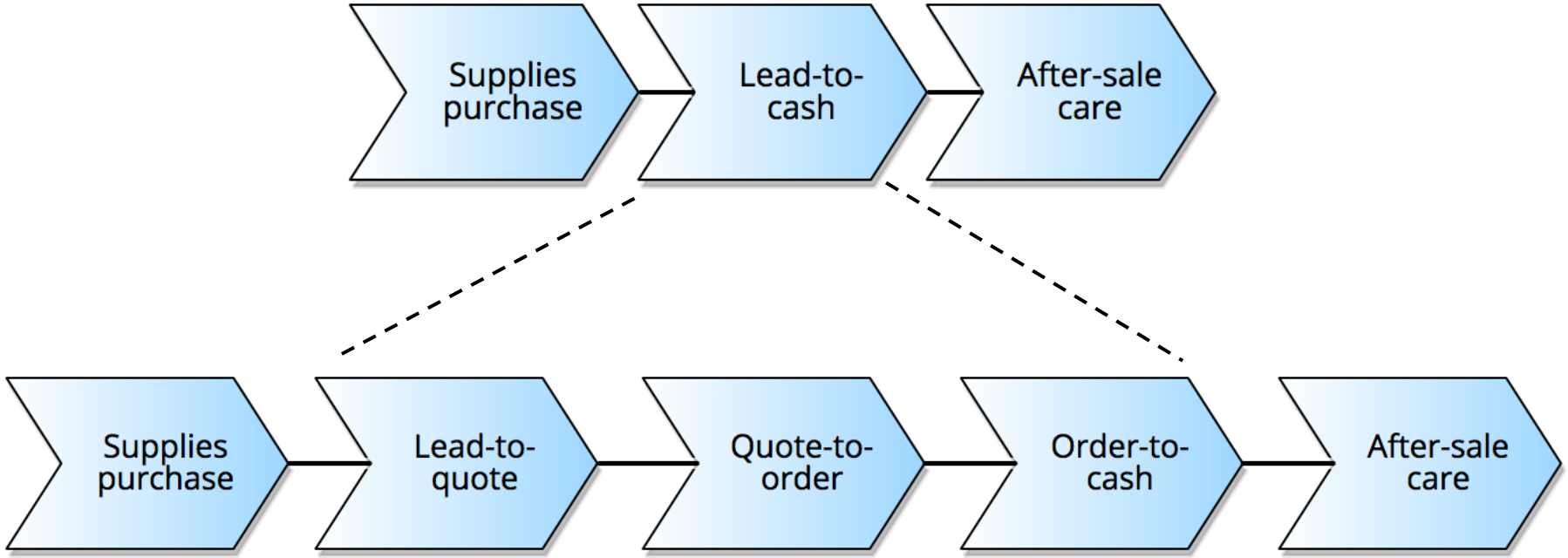
Guidelines to identify horizontal boundaries in value chains

1. Change of key business object in the process
2. Change of granularity of main business object
3. Change in frequency/time
4. Change in intermediate outcome/resolution/objective

Example: value chain

Wholesaler

Core processes

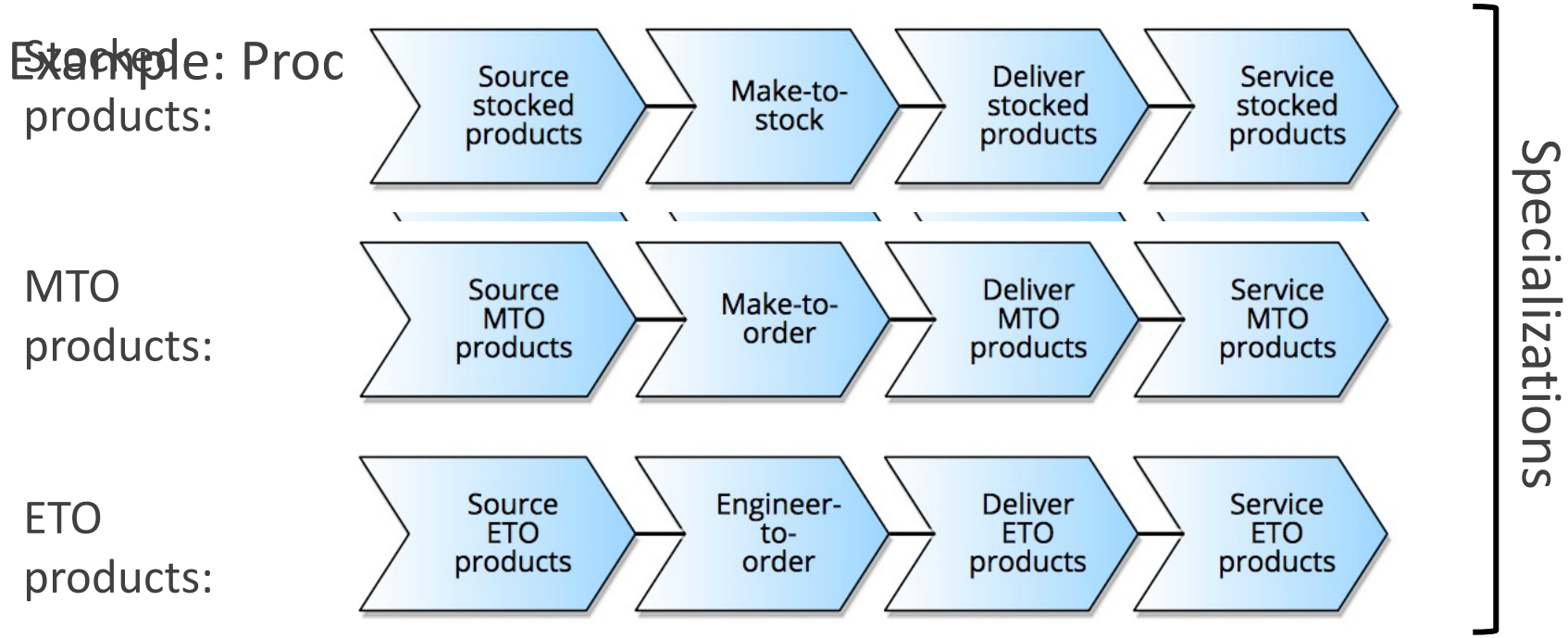


Typical value chains for core processes

Think around three main steps:

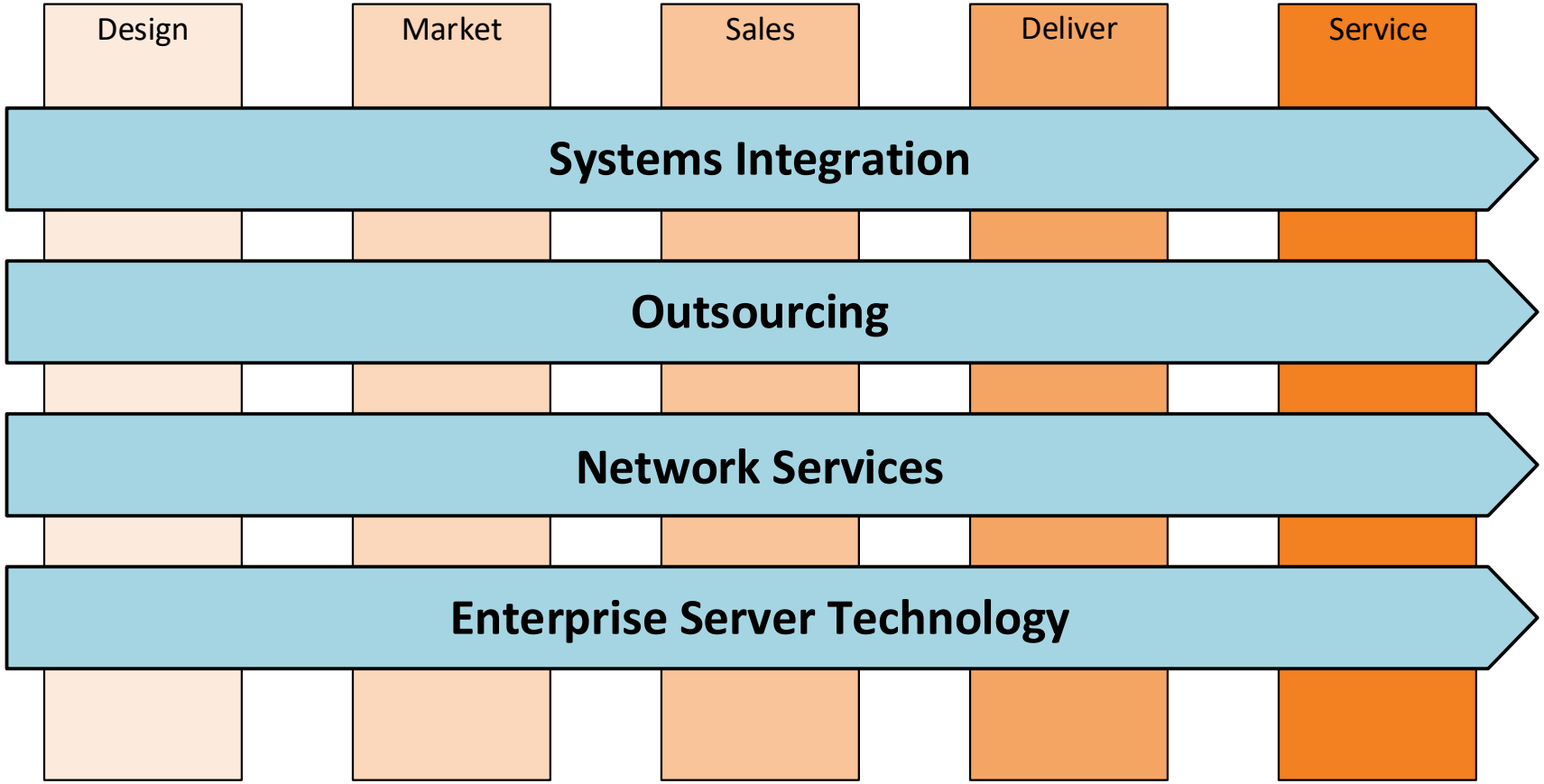
- Imagine it (design new product/service)
- Build it (source, assemble, deliver product/service)
- Sell it (market, sell, service product/service)

Example: Proc



Example: value chains for service provider

IT service provider

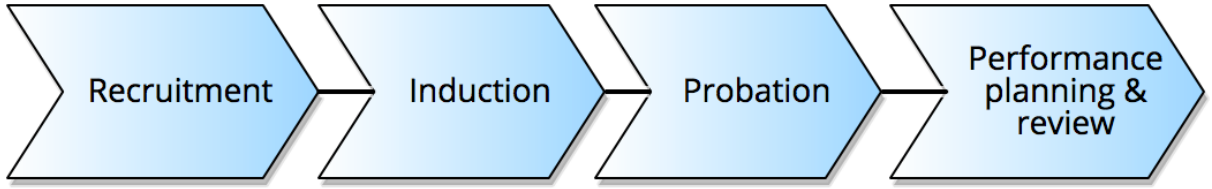


Example: value chain of non-core processes

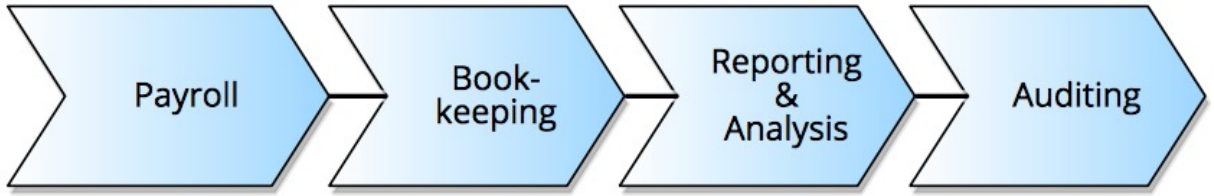
ULB

Support processes

HR:



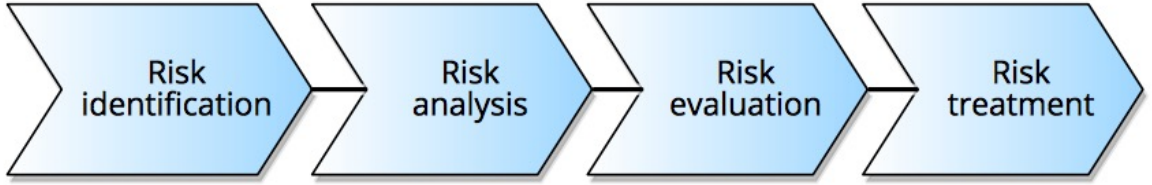
Accounting:



Management processes

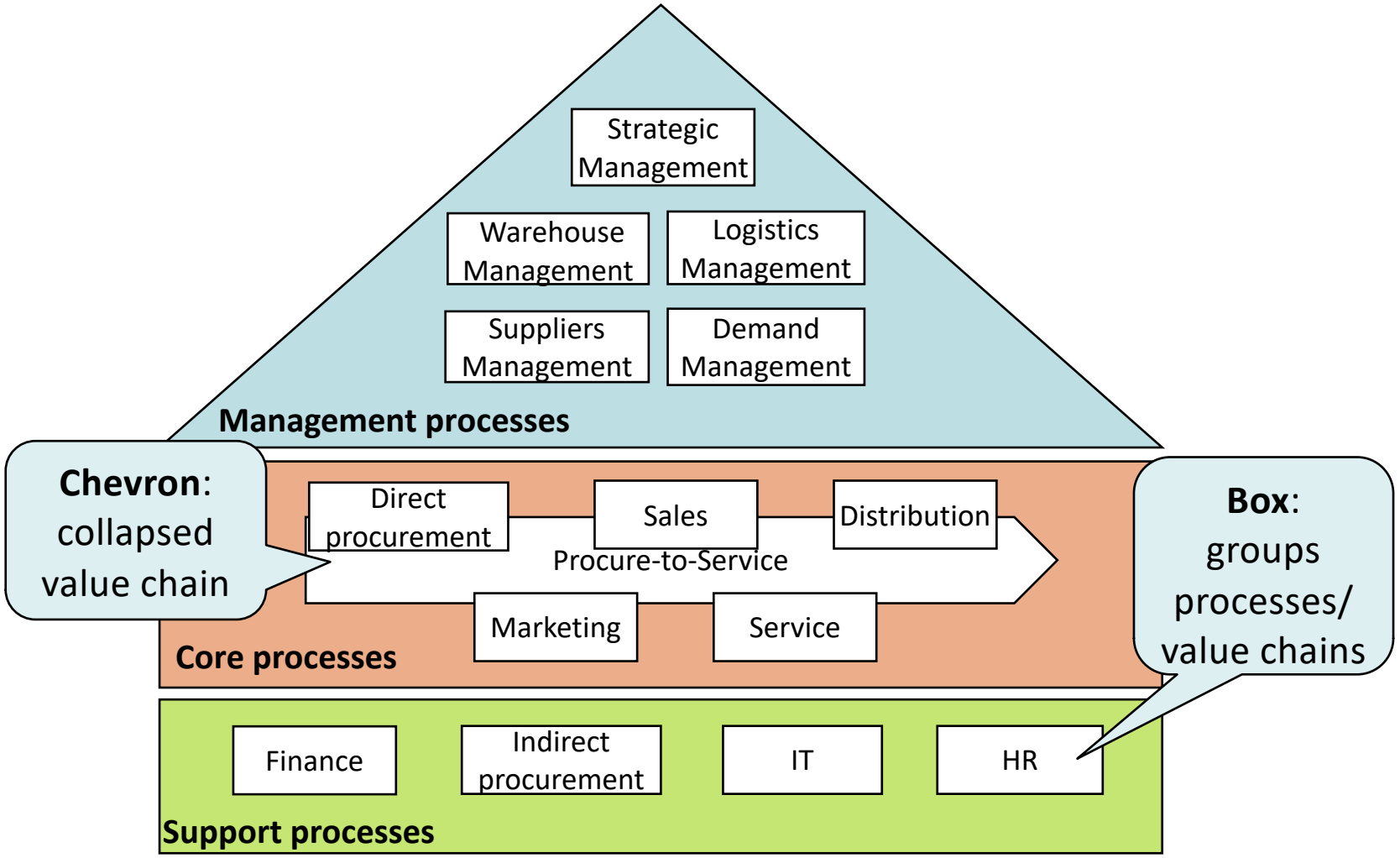
Suppliers management:

Risk management:



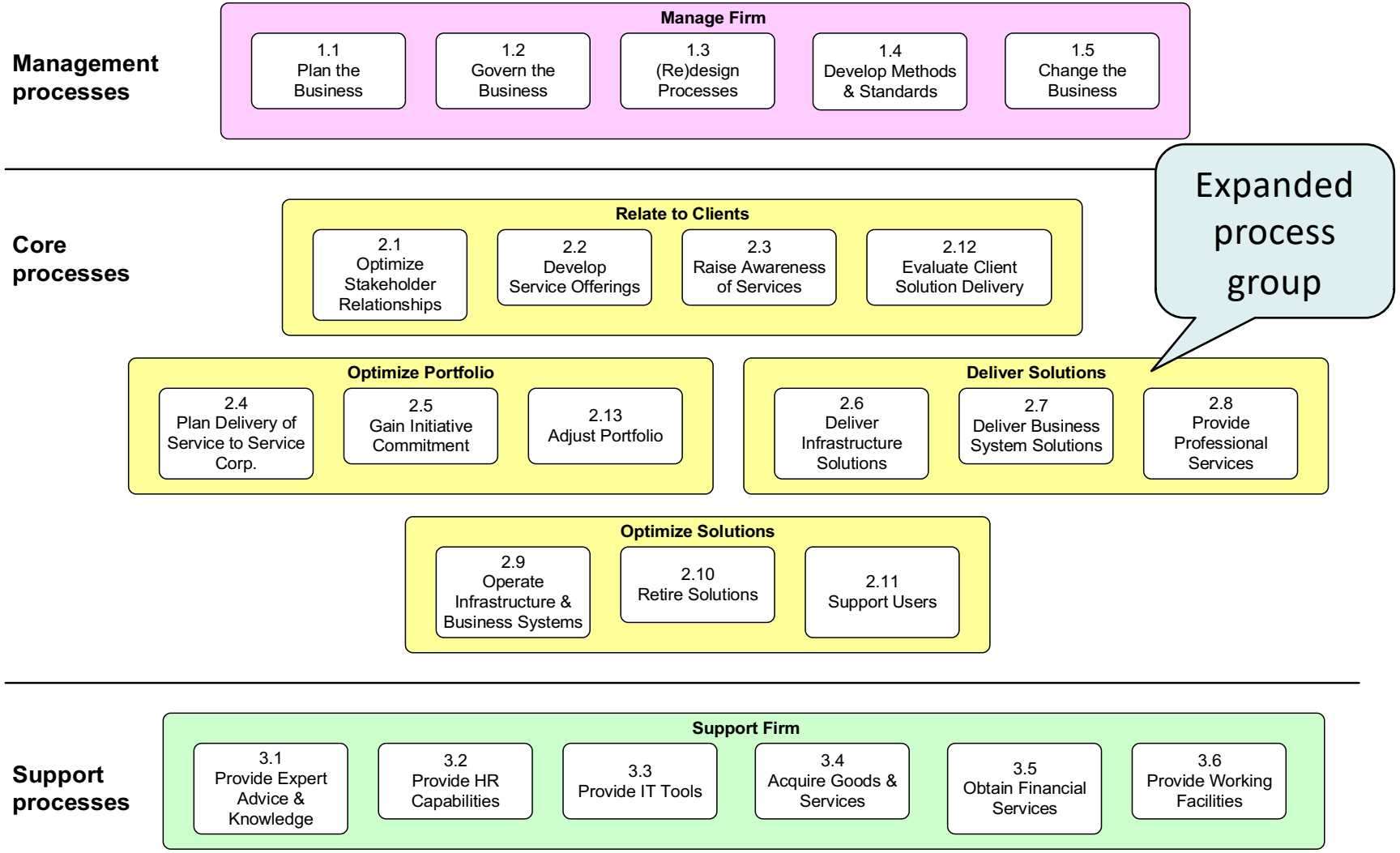
Example: process architecture & value chains

Wholesaler



Alternative: process architecture – groups

Consultancy Firm



Typical artifacts for vertical scoping

Value chains

Chains of processes. Stay at a high level. Rule of thumb: 3-7 processes

- *Procure-to-service, Risk management*

(Root/Main) Processes

Build up value chains and affect each other. They are abstract

- *Lead-to-quote, Quote-to-order, Order-to-cash*

Typical focus of Process enumeration

Subprocesses

Build up processes. They are detailed, involve multiple activities and can be layered on different levels of abstraction (i.e. sub-subprocesses)

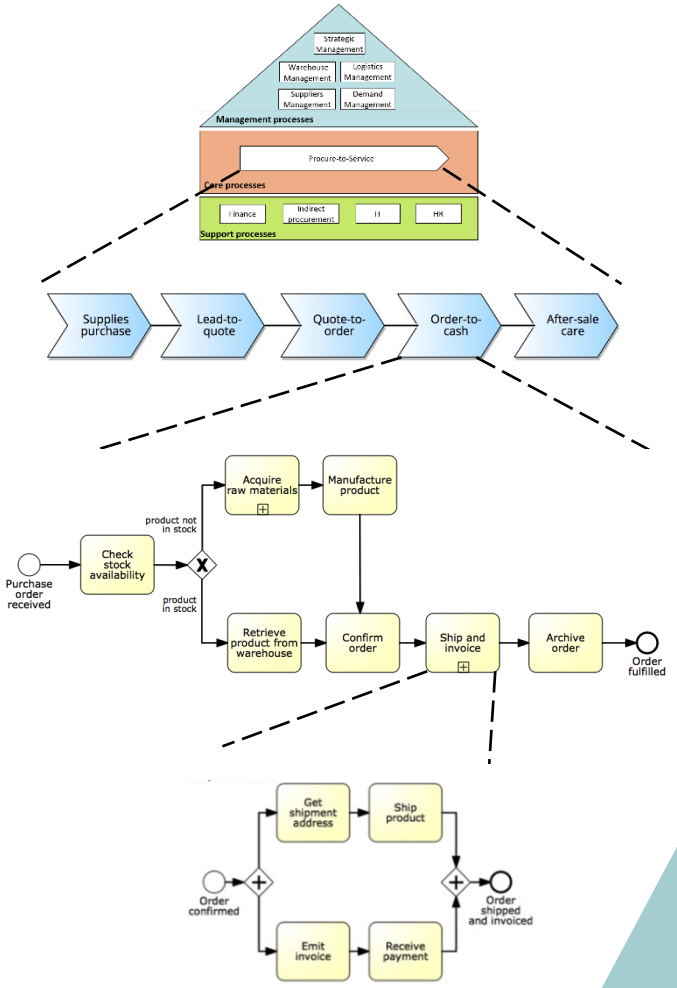
- *Order shipment, invoicing*

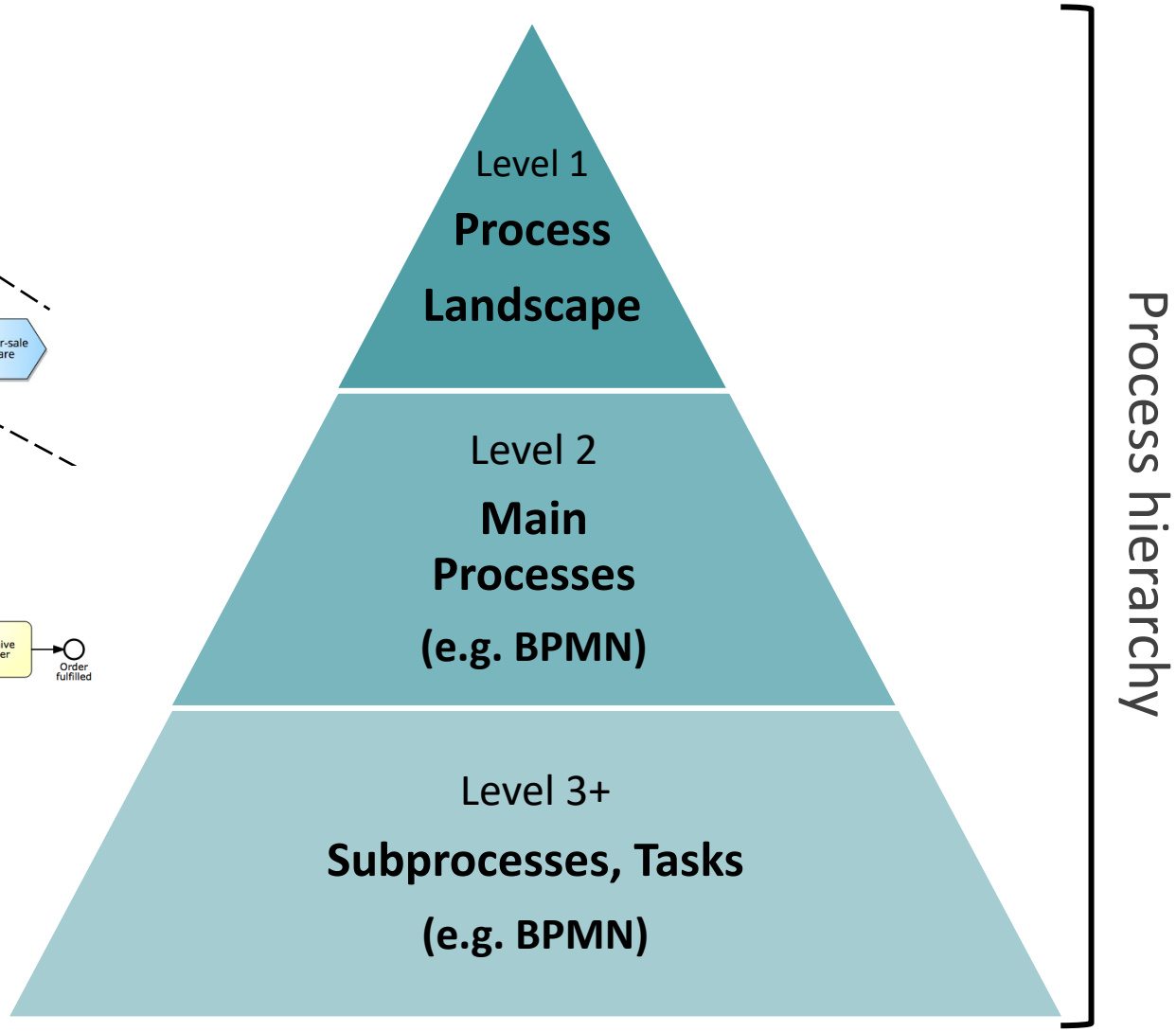
Process tasks

Build up processes and sub-processes. They are atomic and performed by human beings, IT systems or equipment

- *Approve invoice*

Process architecture: hierarchical view



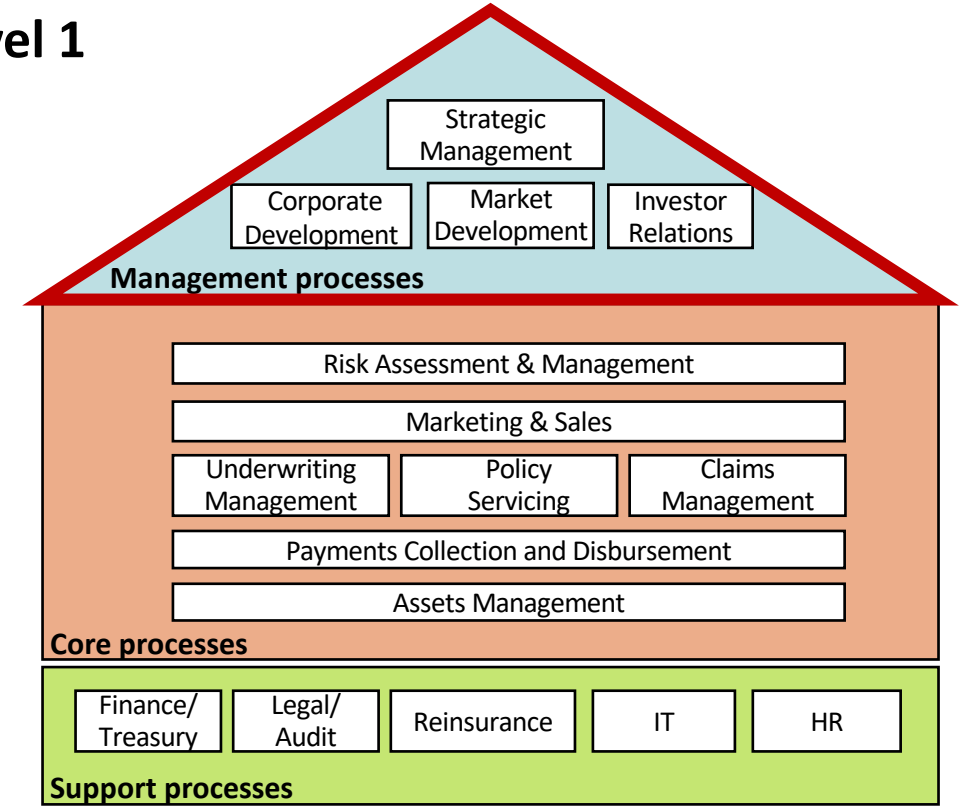


How many levels in the process architecture?

Example: hierarchical process architecture

Insurance company

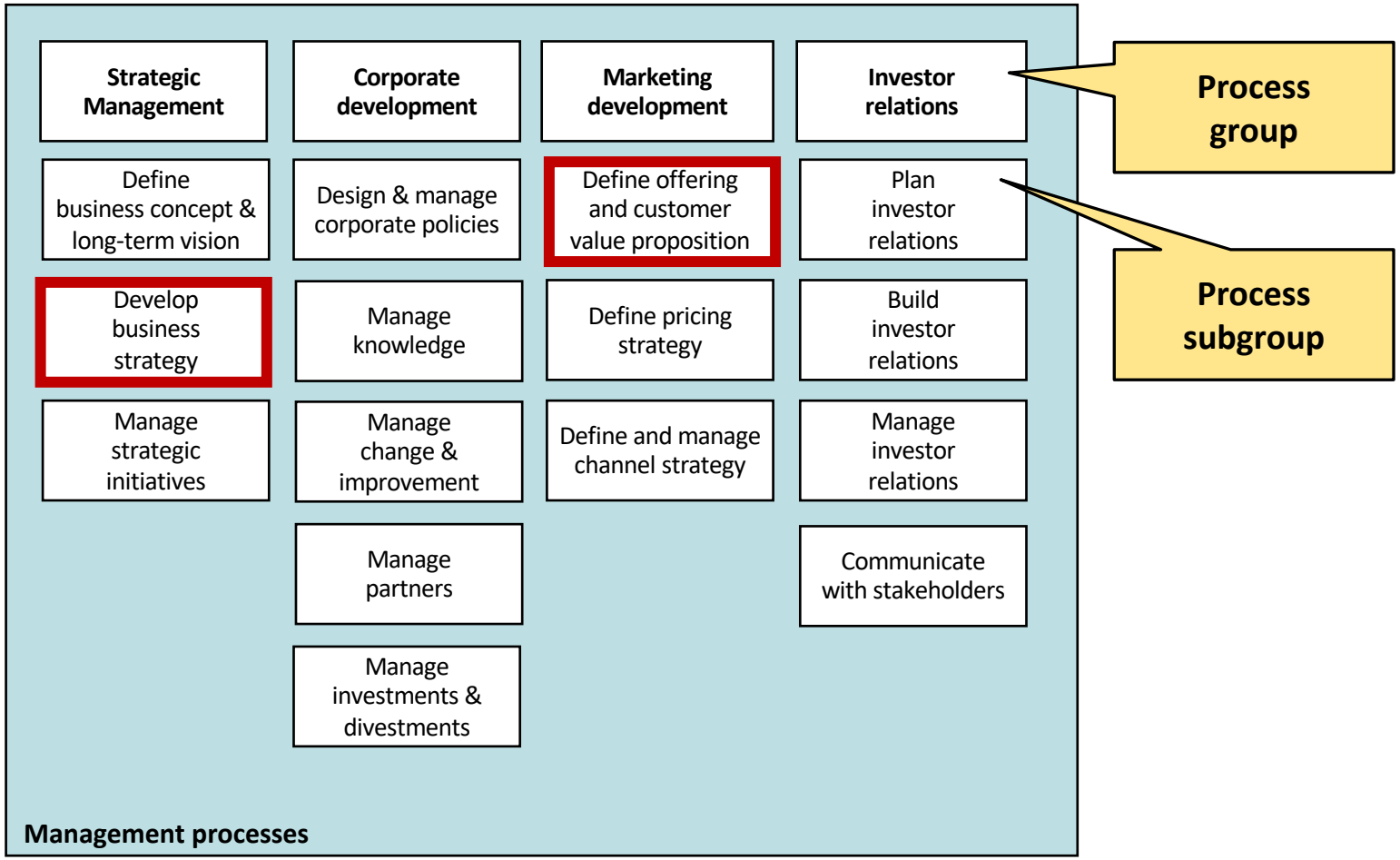
Level 1



Example: hierarchical process architecture

Insurance company

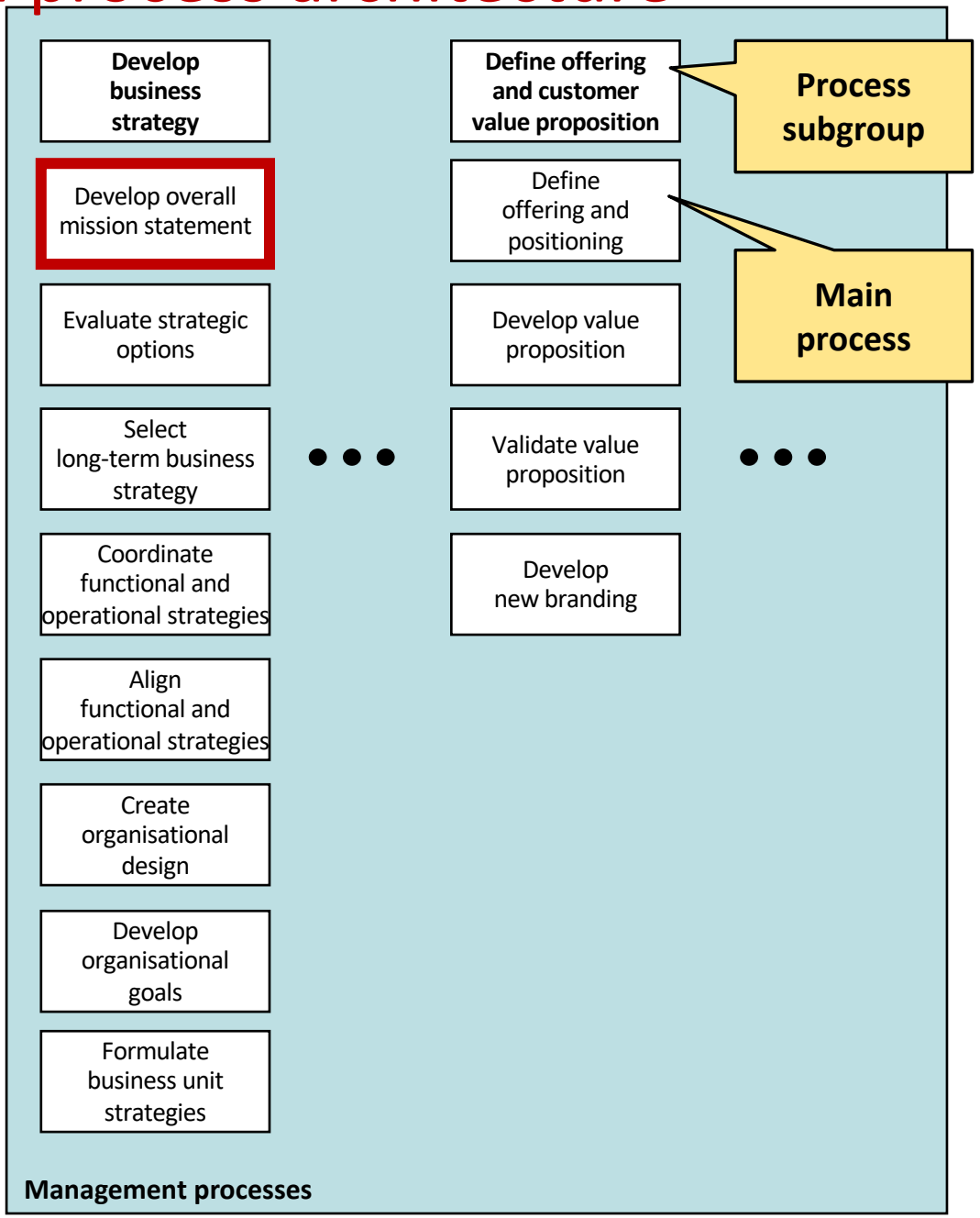
Level 2



Example: hierarchical process architecture

Insurance company

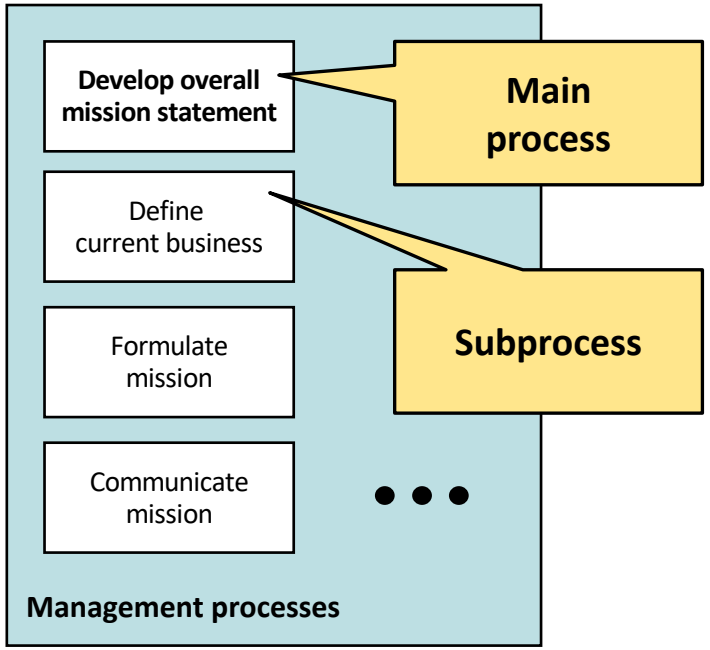
Level 3



Example: hierarchical process architecture

Insurance company

Level 4



Designation via reference models

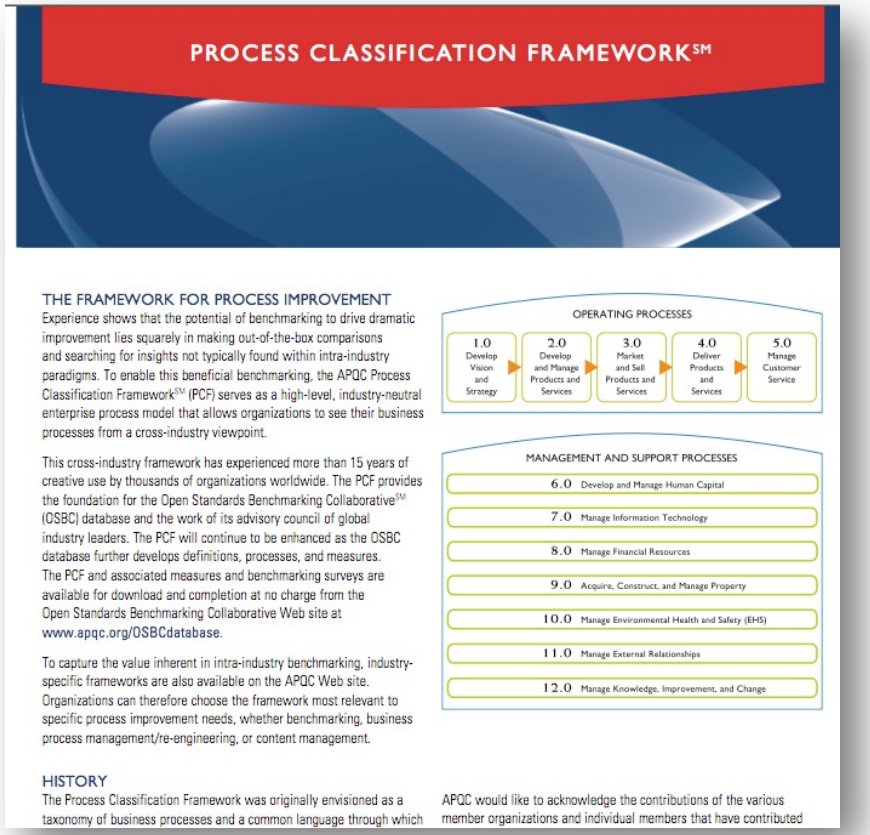
A reference model is used as a template to design the process architecture

Examples:

- Information Technology Infrastructure Library (ITIL)
- Supply Chain Operations Reference Model (SCOR)
- Process Classification Framework (PCF)
- Control Objectives for Information Technology (COBIT)
- Value Reference Model (VRM)
- Voluntary Interindustry Commerce Solutions (VICS)
- eTOM Business Process Framework
- Performance Framework

Example: APQC Process Classification Framework (PCF)

- Industry-neutral enterprise model
- Open standard for benchmarking
- Four levels
  - Categories
  - Process group
  - Process
  - Activity



PROCESS CLASSIFICATION FRAMEWORK℠

THE FRAMEWORK FOR PROCESS IMPROVEMENT

Experience shows that the potential of benchmarking to drive dramatic improvement lies squarely in making out-of-the-box comparisons and searching for insights not typically found within intra-industry paradigms. To enable this beneficial benchmarking, the APQC Process Classification Framework℠ (PCF) serves as a high-level, industry-neutral enterprise process model that allows organizations to see their business processes from a cross-industry viewpoint.

This cross-industry framework has experienced more than 15 years of creative use by thousands of organizations worldwide. The PCF provides the foundation for the Open Standards Benchmarking Collaborative℠ (OSBC) database and the work of its advisory council of global industry leaders. The PCF will continue to be enhanced as the OSBC database further develops definitions, processes, and measures. The PCF and associated measures and benchmarking surveys are available for download and completion at no charge from the Open Standards Benchmarking Collaborative Web site at www.apqc.org/OSBCdatabase.

To capture the value inherent in intra-industry benchmarking, industry-specific frameworks are also available on the APQC Web site. Organizations can therefore choose the framework most relevant to specific process improvement needs, whether benchmarking, business process management/re-engineering, or content management.

OPERATING PROCESSES

1.0 Develop Vision and Strategy
2.0 Develop and Manage Products and Services
3.0 Market and Sell Products and Services
4.0 Deliver Products and Services
5.0 Manage Customer Service

MANAGEMENT AND SUPPORT PROCESSES

6.0 Develop and Manage Human Capital
7.0 Manage Information Technology
8.0 Manage Financial Resources
9.0 Acquire, Construct, and Manage Property
10.0 Manage Environmental Health and Safety (EHS)
11.0 Manage External Relationships
12.0 Manage Knowledge, Improvement, and Change

HISTORY

The Process Classification Framework was originally envisioned as a taxonomy of business processes and a common language through which

APQC would like to acknowledge the contributions of the various member organizations and individual members that have contributed

APQC PCF Overview

Category

APQC Classification Framework

ULB

4.1.8.4 Identify performance trends (10273)
4.1.8.5 Analyze performance benchmark gaps (10274)
4.1.8.6 Prepare appropriate reports (10275)
4.1.8.7 Develop performance improvement plan (10276)
4.1.9 Develop quality standards and procedures (10368)
4.1.9.1 Establish quality targets (10371)
4.1.9.2 Develop standard testing procedures (10372)
4.1.9.3 Communicate quality specifications (10373)

4.2 Procure materials and services (10216) Group

4.2.1 Develop sourcing strategies (10277)
4.2.1.1 Develop procurement plan (10281)
4.2.1.2 Clarify purchasing requirements (10282)
4.2.1.3 Develop inventory strategy (10283)
Activity 4.2.1.4 Match needs to supply capabilities (10284)
4.2.1.5 Analyze company's spend profile (10285)
4.2.1.6 Seek opportunities to improve efficiency and value (10286)
4.2.1.7 Collaborate with suppliers to identify sourcing opportunities (10287)

4.2.2 Select suppliers and develop/maintain contracts

Process

4.3.1.4 Release production orders and create lots (10309)
4.3.2 Produce product (10304)
4.3.2.1 Manage raw material inventory (10310)
4.3.2.2 Execute detailed line schedule (10311)
4.3.2.3 Rerun defective items (10313)
4.3.2.4 Assess production performance (10314)
4.3.3 Schedule and perform maintenance (10305)
4.3.3.1 Determine process for preventive (planned) maintenance (Preventive Maintenance Orders) (10315)
4.3.3.2 Determine process for requested (unplanned) maintenance (Work Order Cycle) (10316)
4.3.3.3 Execute maintenance (10317)
4.3.3.4 Calibrate test equipment (10318)
4.3.3.5 Report maintenance issues (10319)
4.3.4 Perform quality testing (10369)
4.3.4.1 Perform testing using the standard testing procedure (10374)
4.3.4.2 Record test results (10375)
4.3.5 Maintain production records and manage lot traceability (10370)
4.3.5.1 Determine lot numbering system (10376)

Prioritization (aka Process Selection)

1. **Importance**

   Which processes have greatest impact on the organization‘s strategic objectives?

2. **Health (or Dysfunction)**

   Which processes are in deepest trouble?

3. **Feasibility**

   Which processes are most susceptible to successful process management?

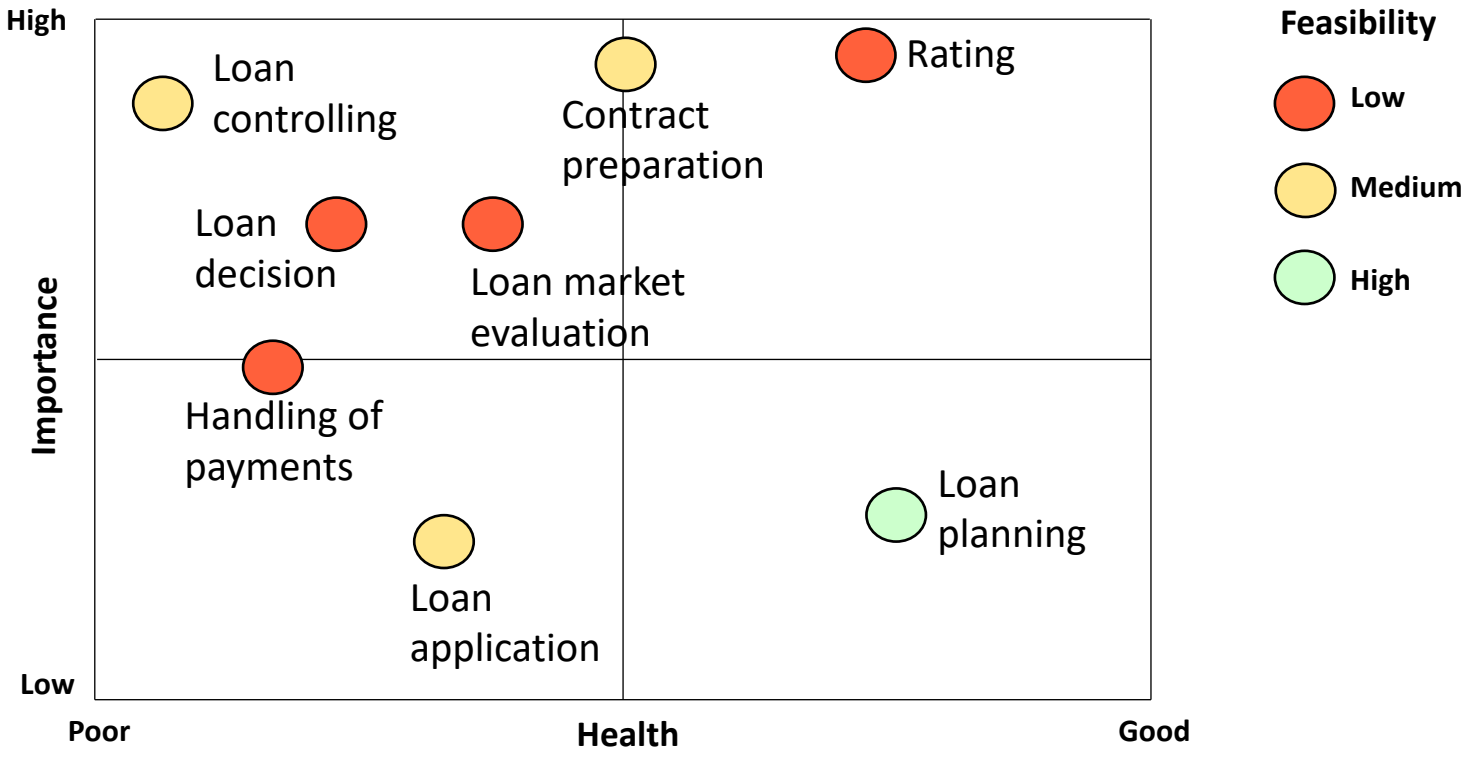
↓

Prioritized process portfolio

Hammer, Champy (1993)

Example: prioritized process portfolio

Financial institution



Acknowledgements

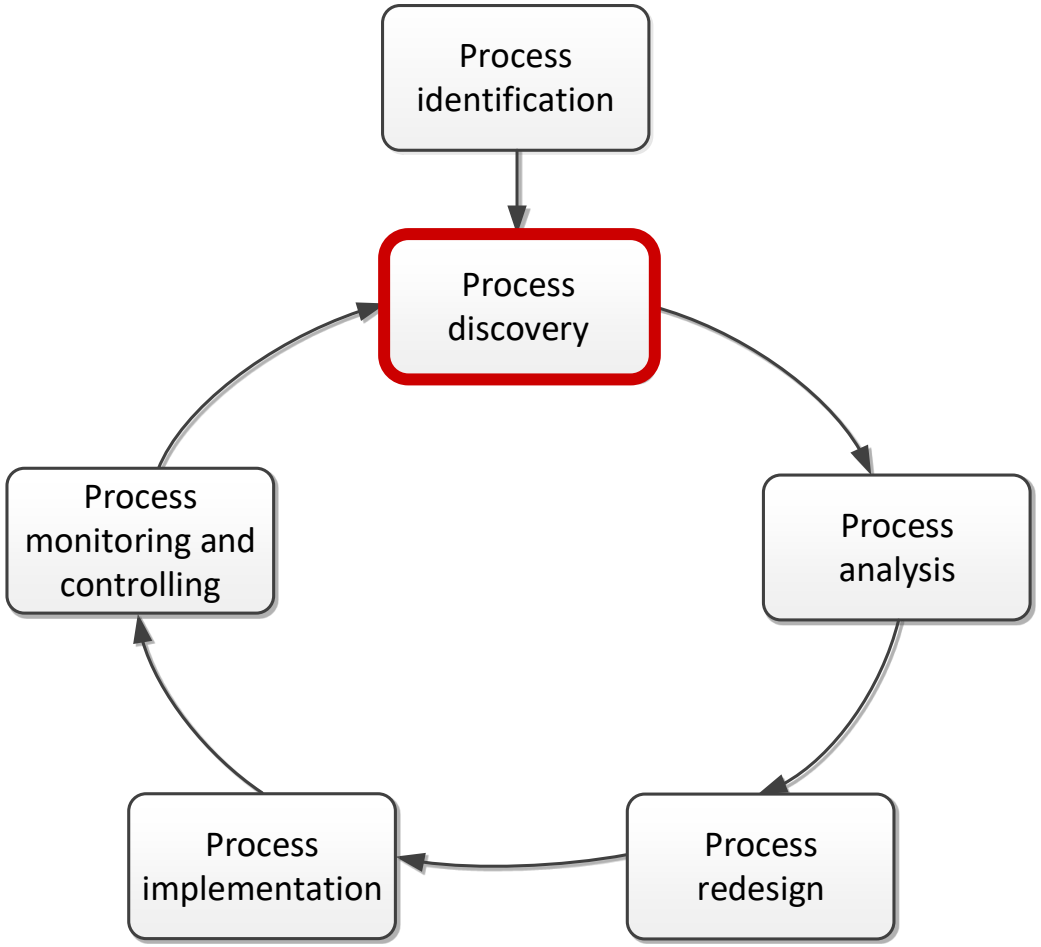
- All material comes from Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A. Reijers, authors of the “Fundamentals of Business Process Management” book.

INFO-H420
Management of Data Science and Business Workflows
Part I
2. Process Modeling

Dimitris SACHARIDIS

2023-2024

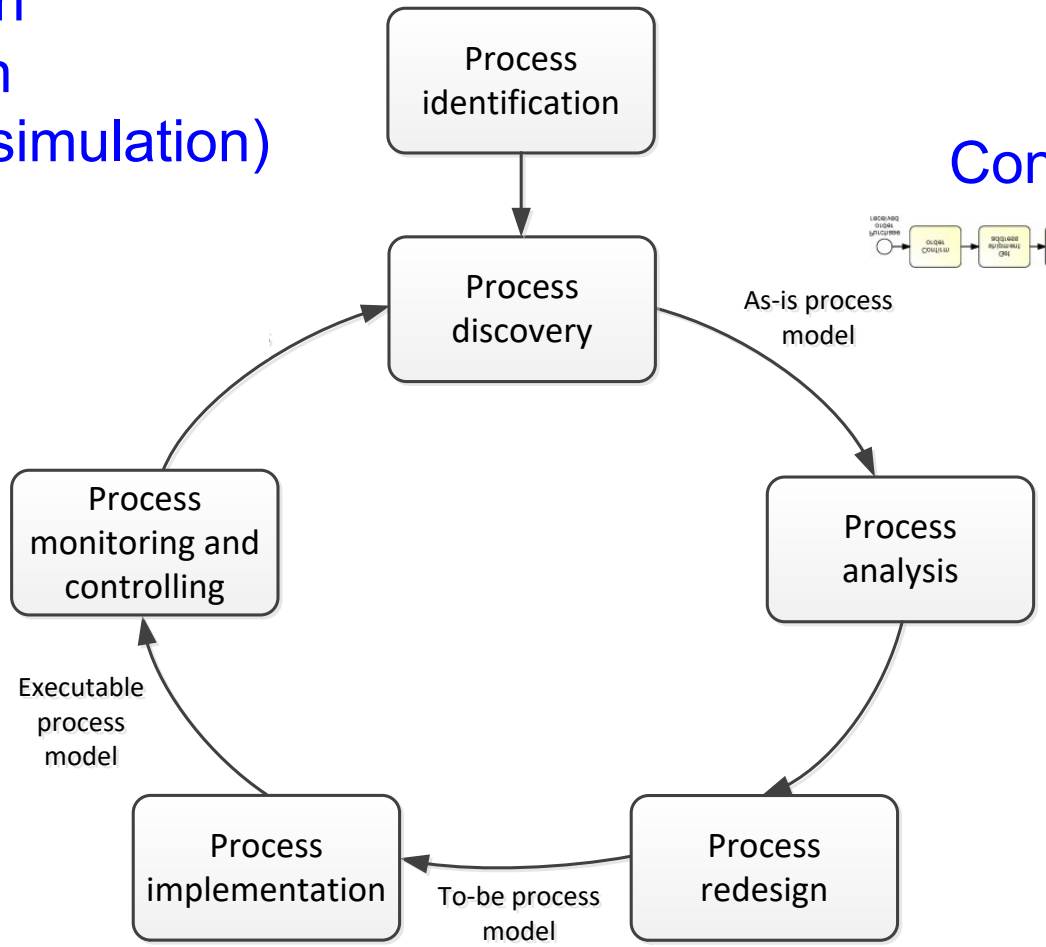
BPM lifecycle



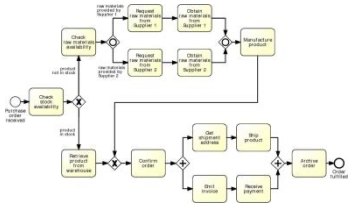
Purposes of process modeling

- Communication
- Documentation
- Analysis (e.g. simulation)

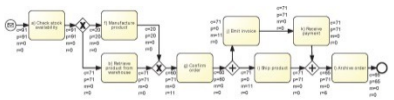
Conceptual



Executable



- Automation
- Testing



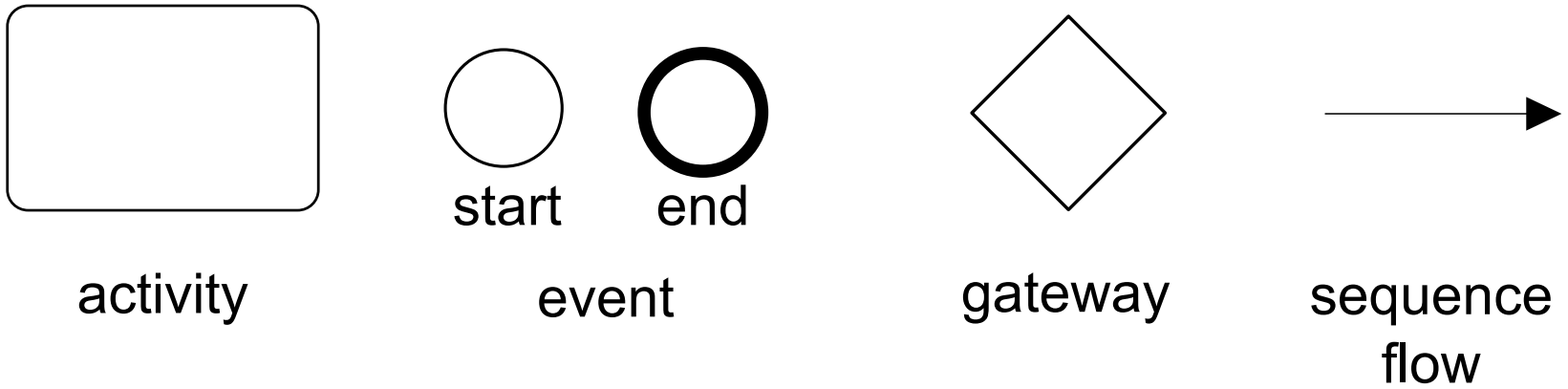
Conceptual

Business Process Model and Notation (BPMN)

- OMG standard (nowadays BPMN 2.0)
- Both for conceptual and executable models
- Supported by numerous tools: bpmn.org lists over 70 tools, incl.
  - **Camunda**
  - **Apromore**
  - Signavio
  - Bizagi Process Modeler
  - Cameo Business Analyst

BPMN from 10,000 miles…

A BPMN process model is a graph consisting of four types of **core elements**:



Let’s start modeling

Order-to-cash

An order-to-cash process is triggered by the receipt of a purchase order from a customer. Upon receipt, the purchase order has to be checked against the stock to determine if the the requested item(s) are available. Depending on stock availability the purchase order may be confirmed or rejected.

If the purchase order is confirmed, an invoice is emitted, and the goods requested are shipped. The process completes by archiving the order.

Let’s start modeling – break it down

Order-to-cash

- An order-to-cash process is triggered by the receipt of a purchase order from a customer.
- Upon receipt, the purchase order has to be checked against the stock to determine if the requested item(s) are available.
- Depending on stock availability the purchase order may be confirmed or rejected.
- If the purchase order is confirmed, an invoice is emitted and the goods requested are shipped. The process completes by archiving the order.

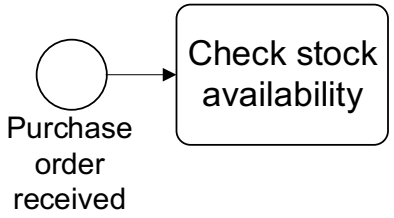
Let’s start modeling – break it down

Order-to-cash

- **An order-to-cash process is triggered by the receipt of a purchase order from a customer.**
- **Upon receipt, the purchase order has to be checked against the stock to determine if the requested item(s) are available.**

BPMN Model

Order-to-cash



Naming conventions

- Event: noun + past-participle verb (e.g., insurance claim lodged)
- Activity: verb + noun (e.g., assess credit risk)

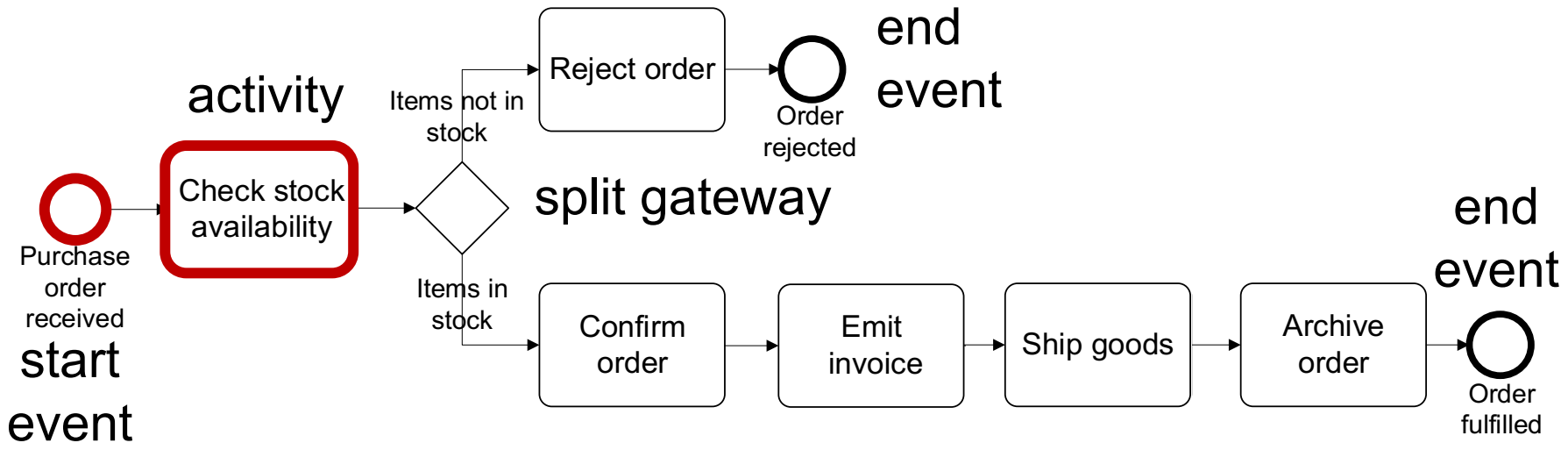
Let’s start modeling – break it down

Order-to-cash

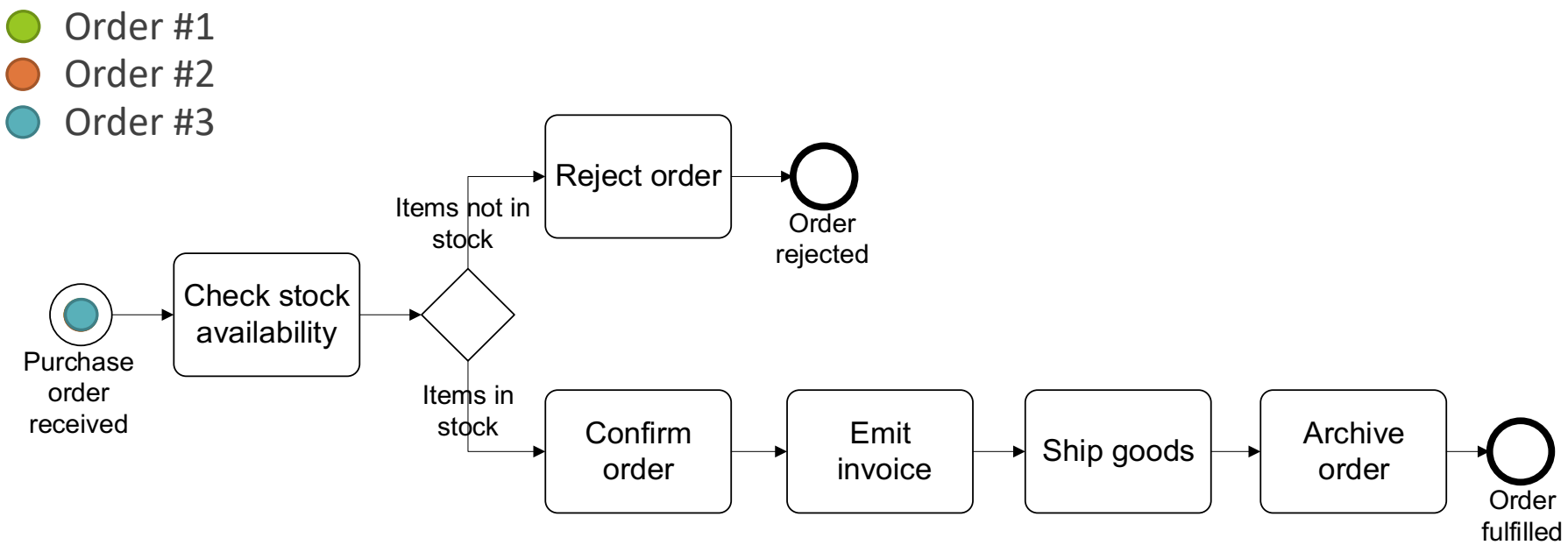
- An order-to-cash process is triggered by the receipt of a purchase order from a customer.
- Upon receipt, the purchase order has to be checked against the stock to determine if the requested item(s) are available.
- **Depending on stock availability the purchase order may be confirmed or rejected.**
- **If the purchase order is confirmed, an invoice is emitted and the goods requested are shipped. The process completes by archiving the order.**

BPMN Model

Order-to-cash

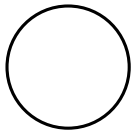


Execution of a process model
“Game of Tokens”



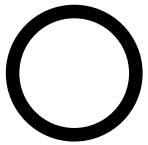
A little bit more on events…

A *start event* triggers a new process instance by generating a token that traverses the sequence flow ("tokens source")



start event

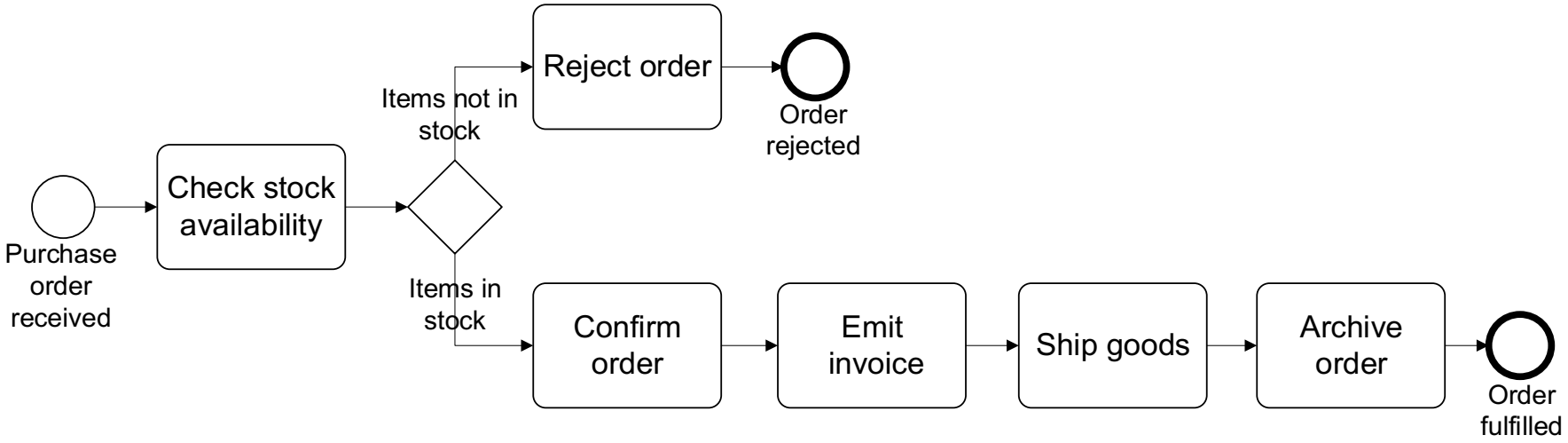
An *end event* signals that a process instance has completed with a given outcome by consuming a token ("tokens sink")



end event

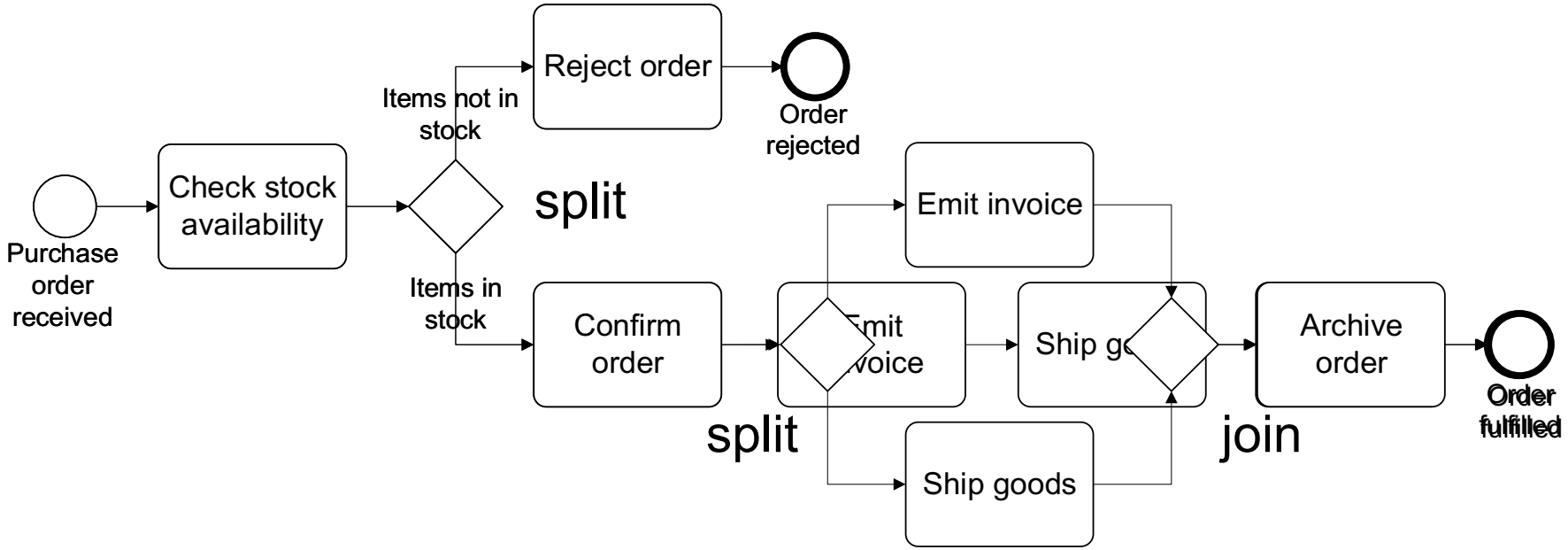
Order-to-cash example revisited…

[…] If the purchase order is confirmed, **an invoice is emitted and the goods requested are shipped (in any order)**. The process completes by archiving the order. […]

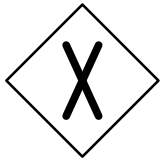


First try

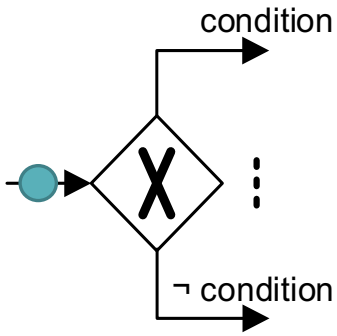
Order-to-cash



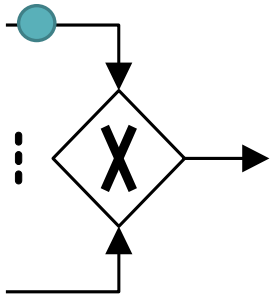
A little more on gateways: XOR Gateway



An *XOR Gateway* captures decision points (XOR-split) and points where alternative flows are merged (XOR-join)



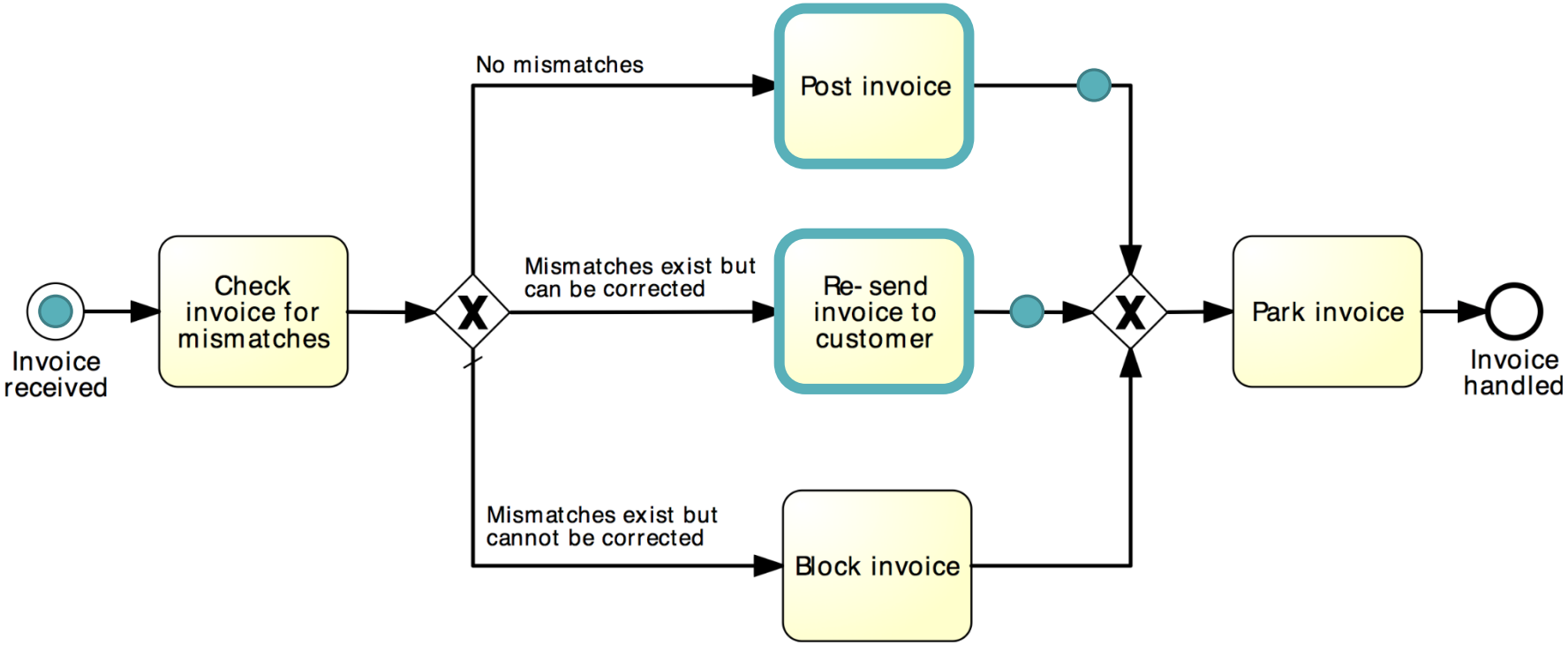
XOR-split ➔ takes **one** outgoing branch



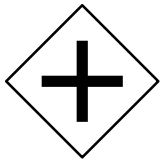
XOR-join ➔ proceeds when **one** incoming branch has completed

Example: XOR Gateway

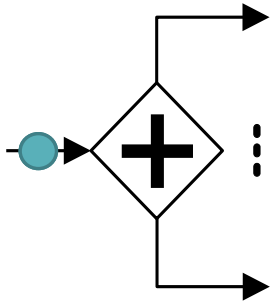
Invoice checking process



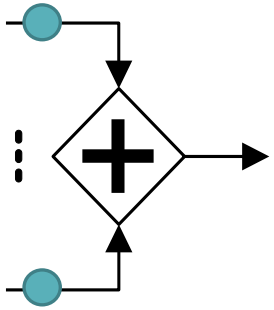
A little more on gateways: AND Gateway



An *AND Gateway* provides a mechanism to create and synchronize “parallel” flows.



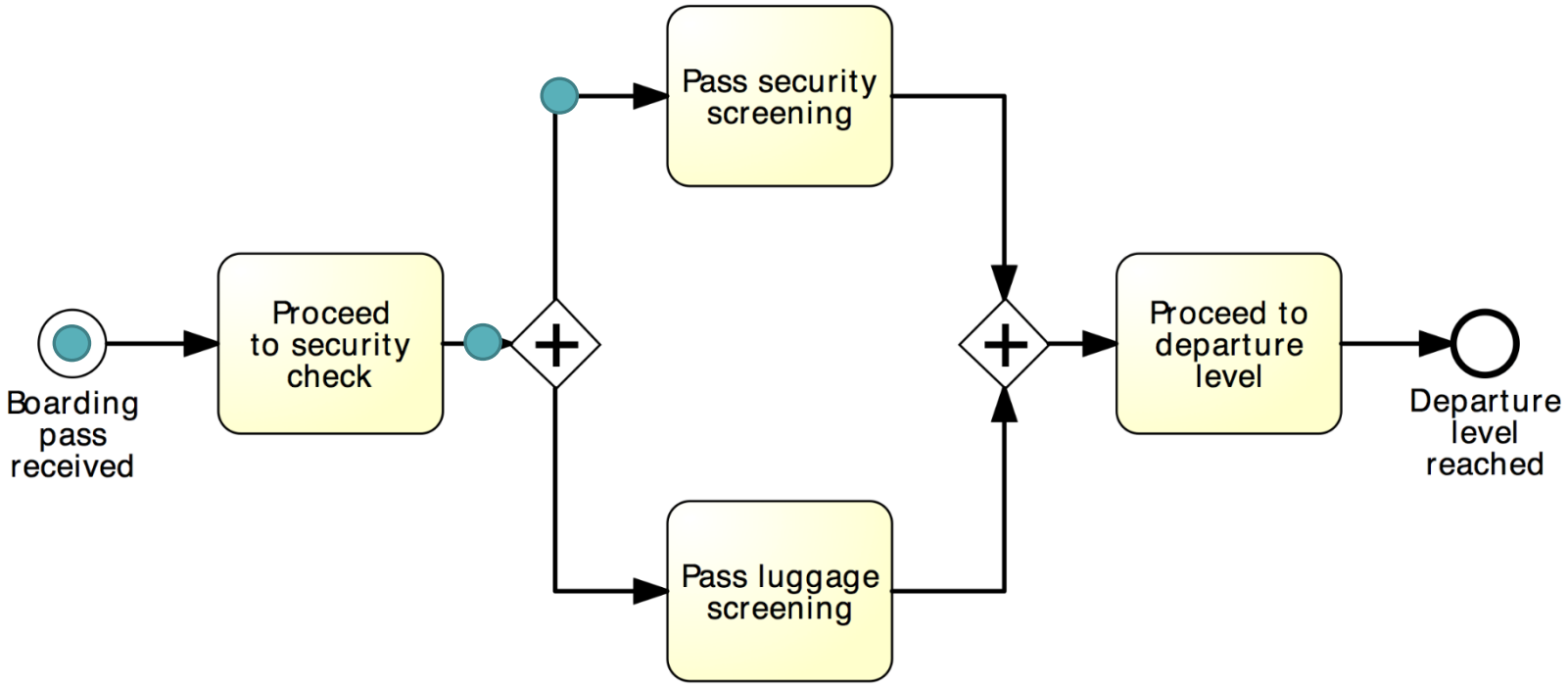
AND-split ➔ takes **all** outgoing branches



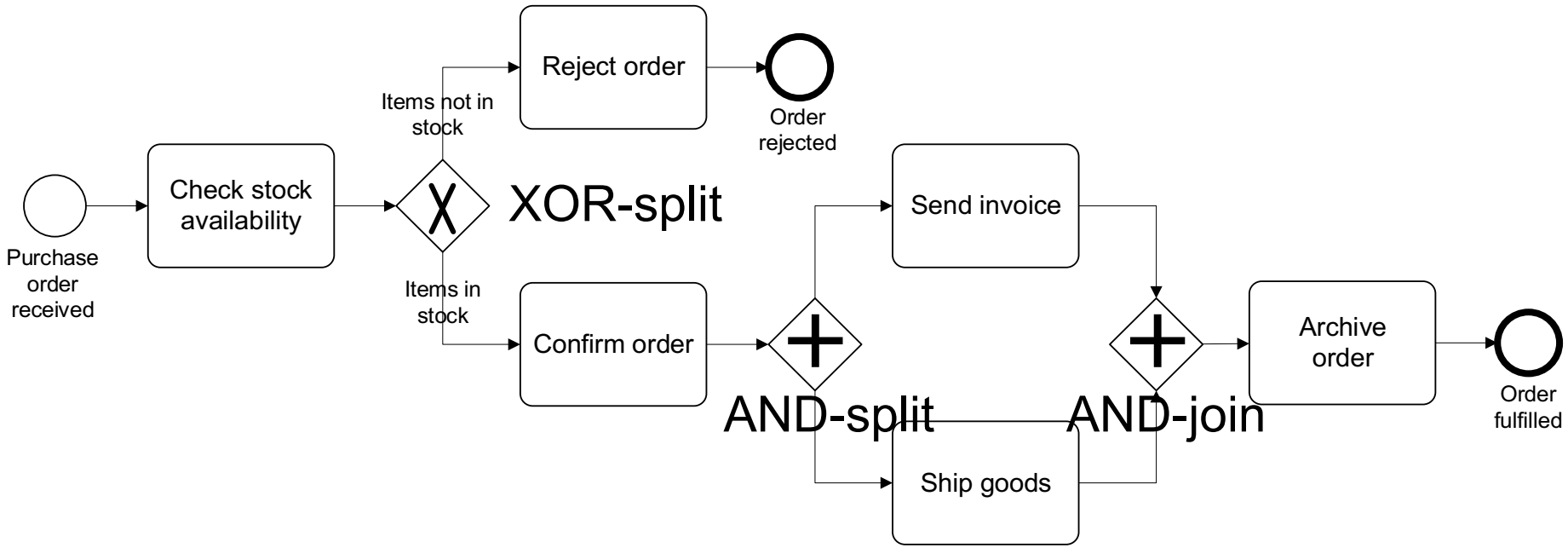
AND-join ➔ proceeds when **all** incoming branches have completed

Example: AND Gateway

Airport security check



Revised order-to-cash process model



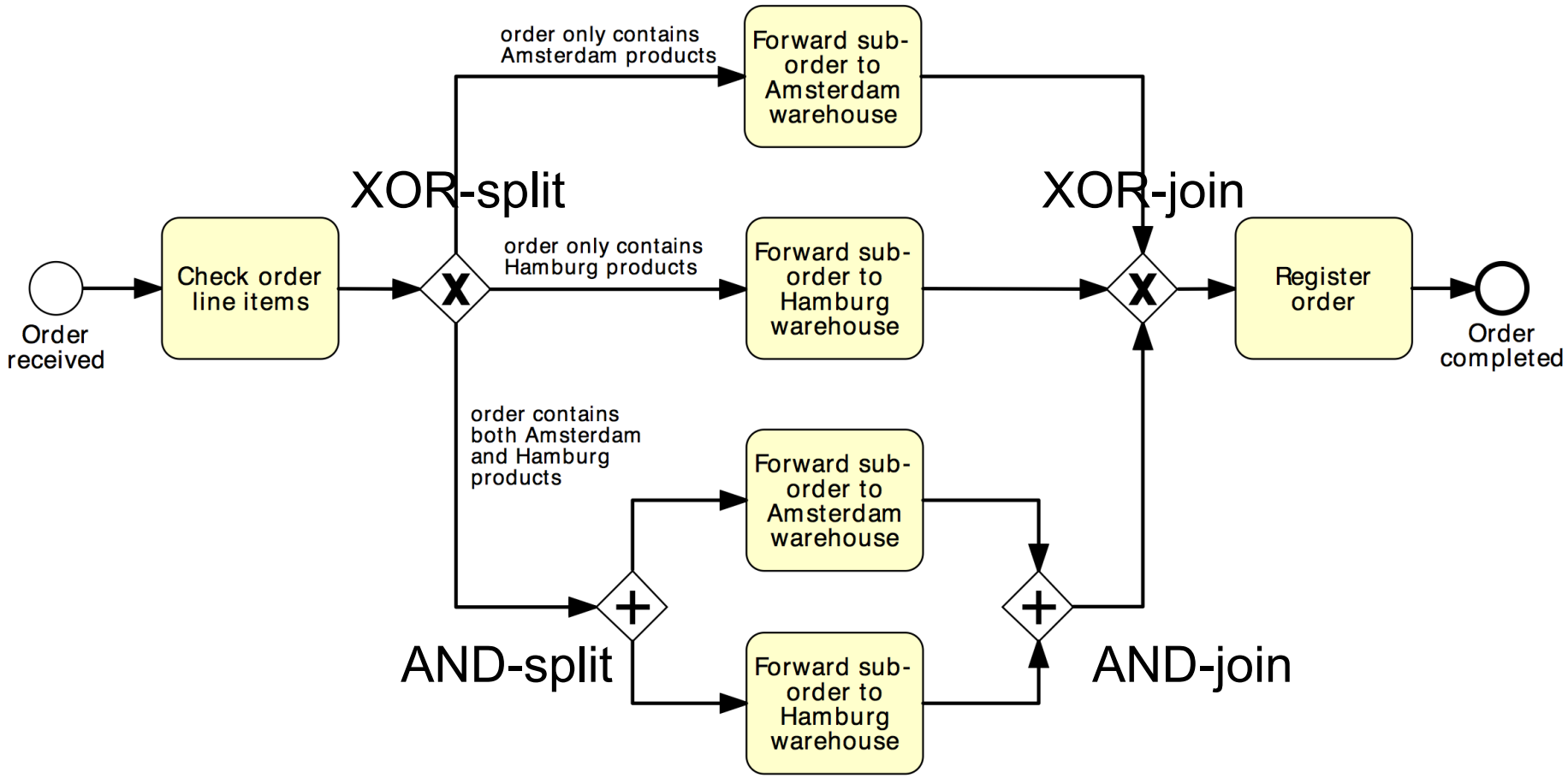
Between XOR and AND

Order distribution process

A company has two warehouses that store different products: Amsterdam and Hamburg. When an order is received, it is distributed across these warehouses: if some of the relevant products are maintained in Amsterdam, a sub-order is sent there; likewise, if some relevant products are maintained in Hamburg, a sub-order is sent there. Afterwards, the order is registered and the process completes.

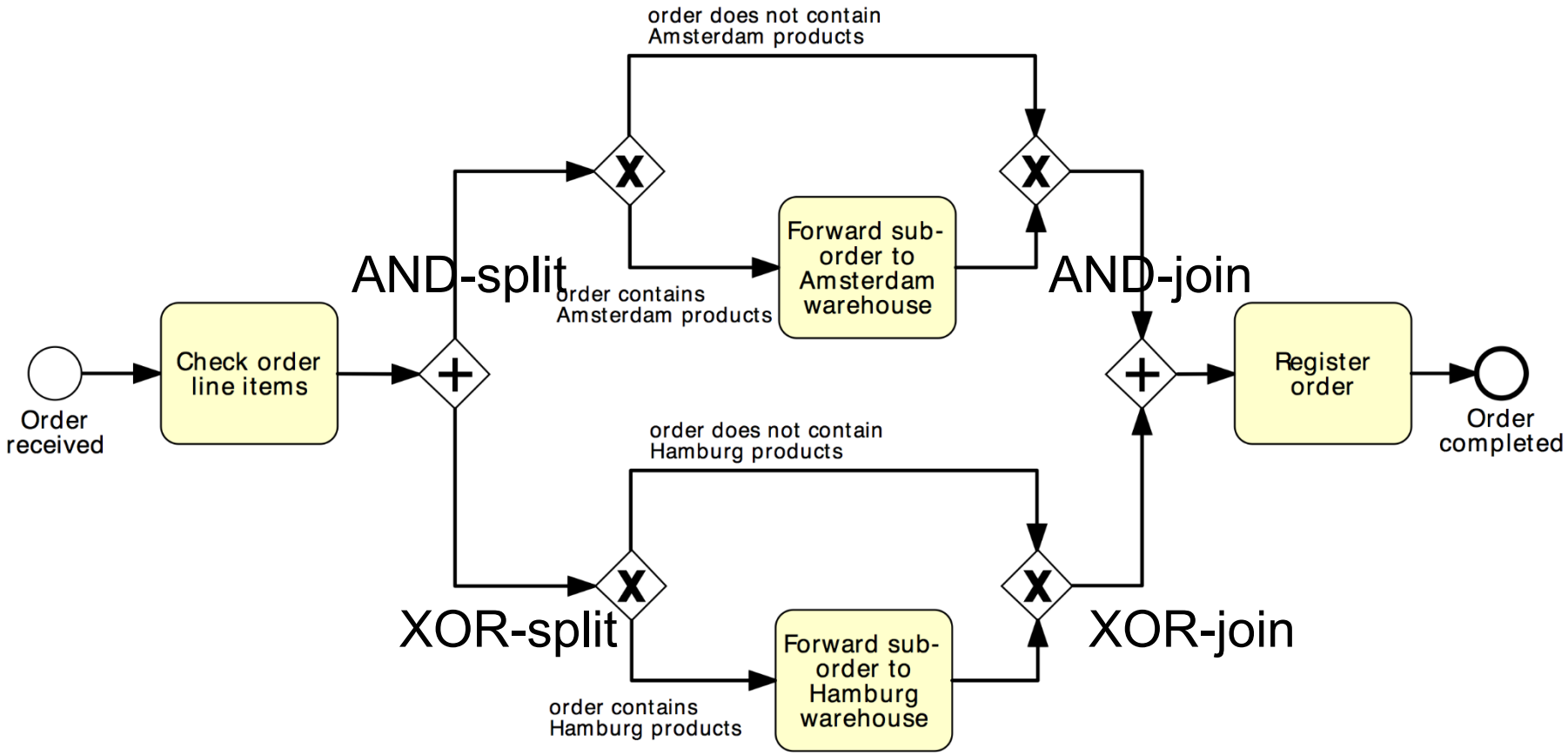
Solution 1

Order distribution process

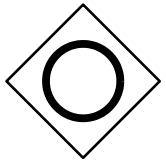


Solution 2

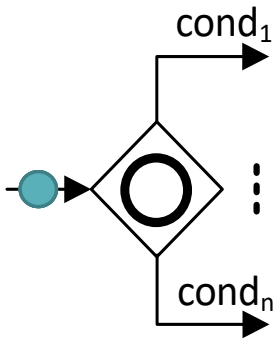
Order distribution process



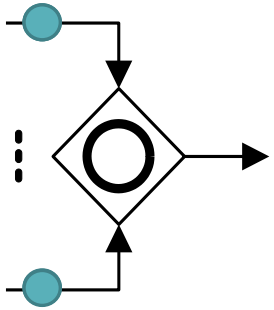
OR Gateway



An *OR Gateway* provides a mechanism to create and synchronize n out of m parallel flows.



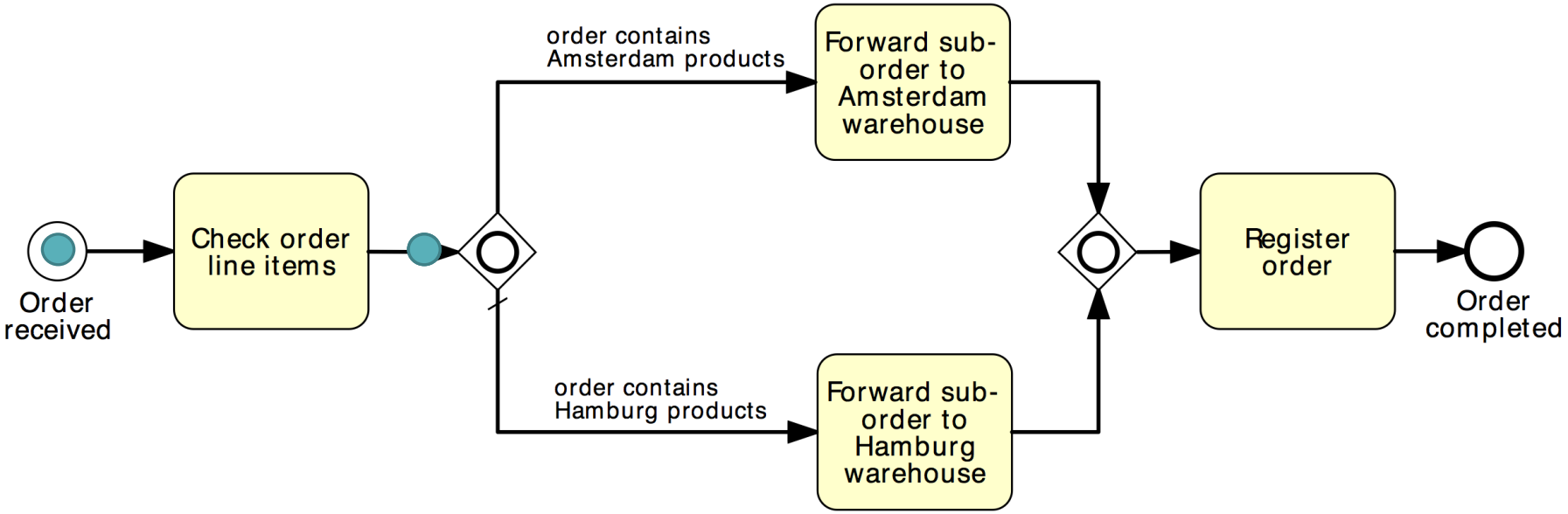
OR-split ➔ takes one or more branches depending on conditions



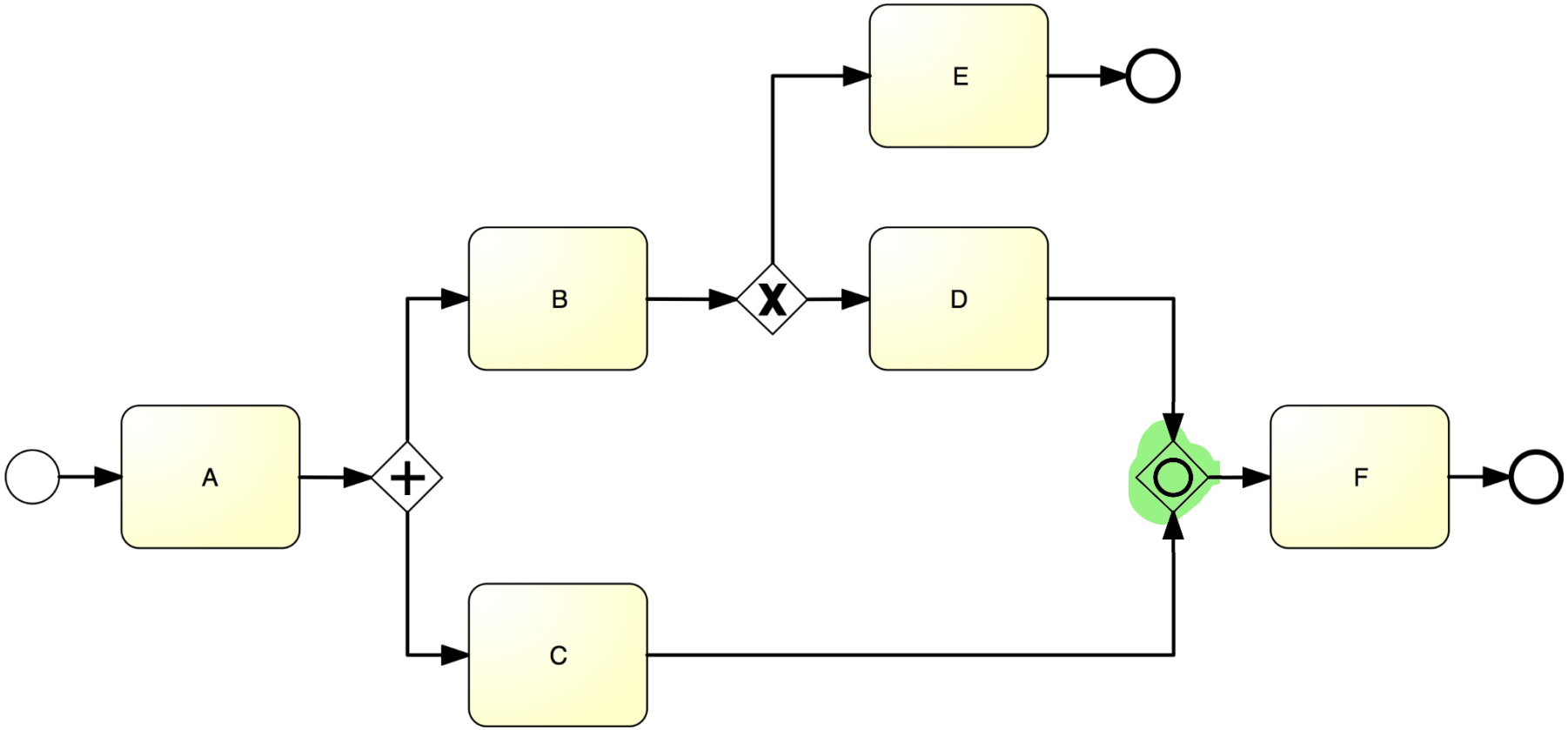
OR-join ➔ proceeds when all **active** incoming branches have completed

Solution using OR Gateway

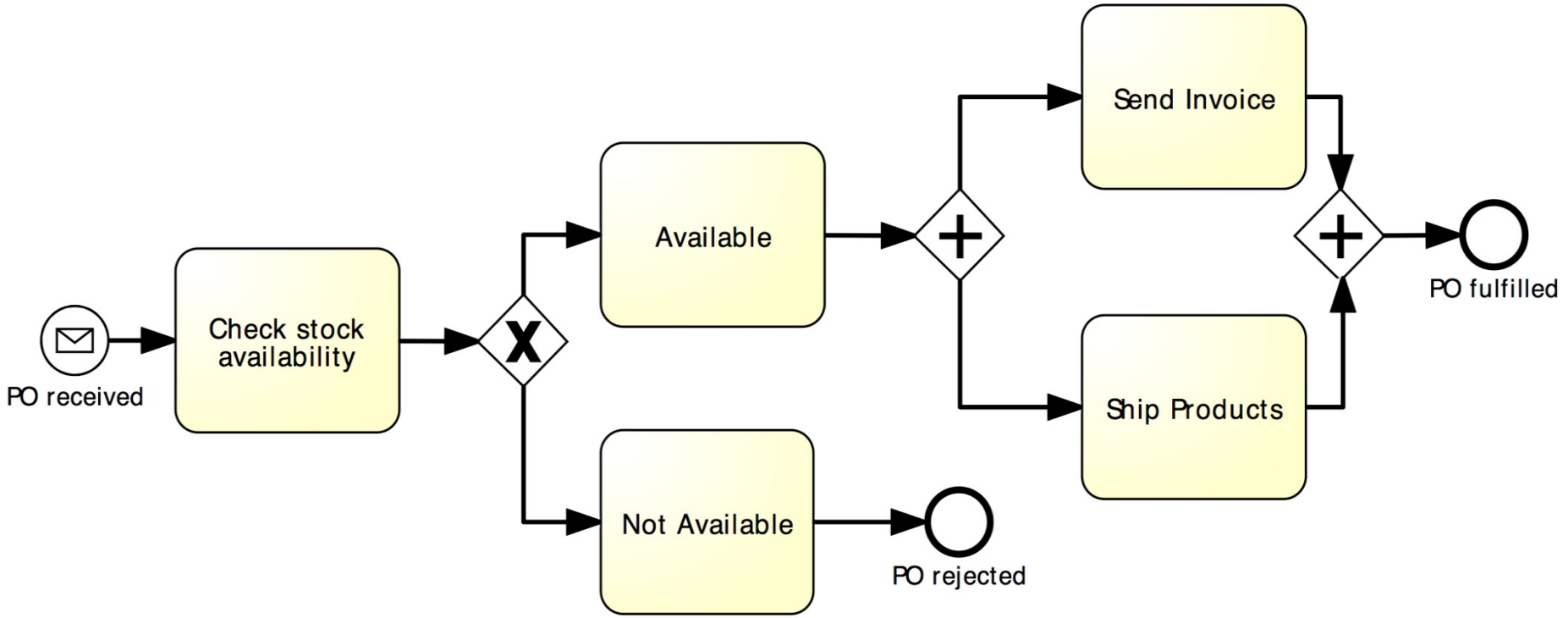
Order distribution process



What join type do we need here?



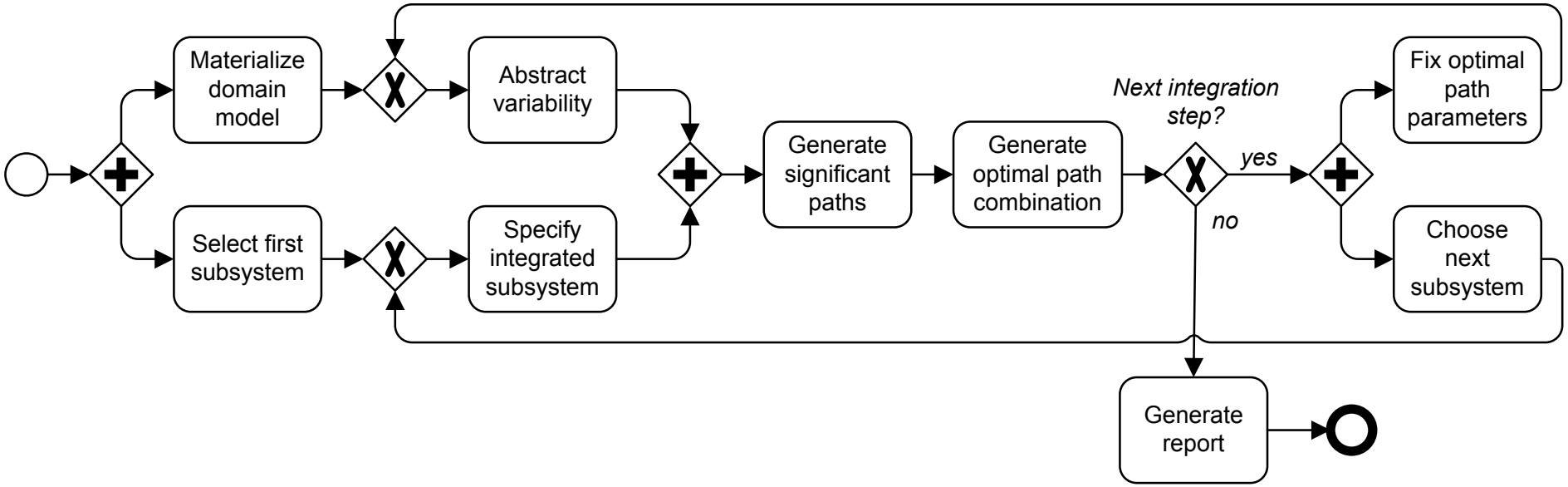
Beware: Beginner's Mistake…

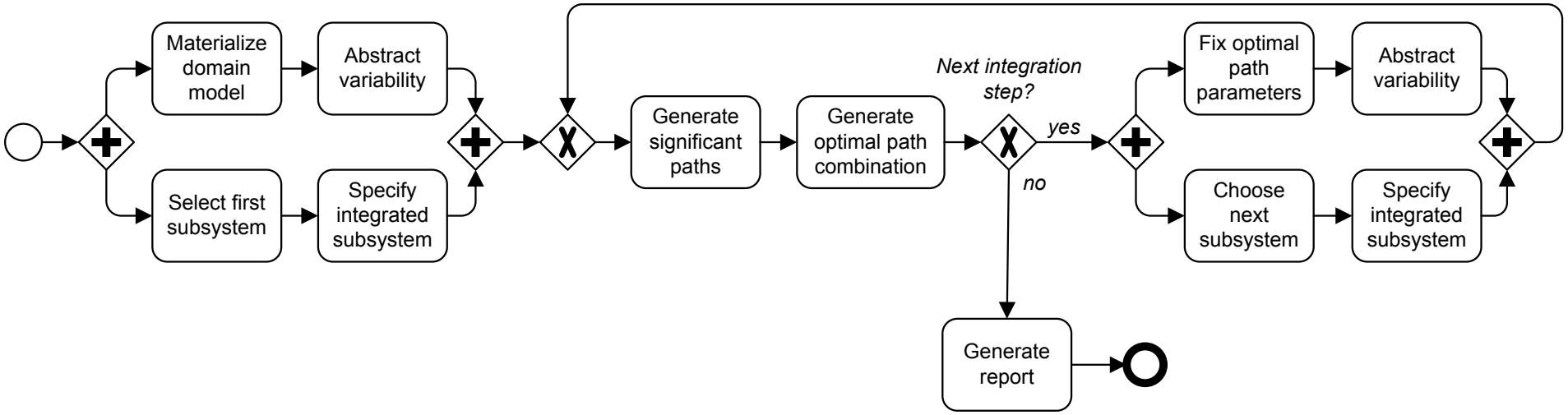


Guidelines: Naming Conventions

1. Give a name to every event and task
2. For tasks: verb followed by business object name and possibly complement
   - Issue Driver Licence, Renew Licence via Agency
3. For message events: object + past participle
   - Invoice received, Claim settled
4. Avoid generic verbs such as Handle, Record…
5. Label each XOR-split with a condition
   - Policy is invalid, Claim is inadmissible

Poll: Which model do you prefer?





One more guideline…

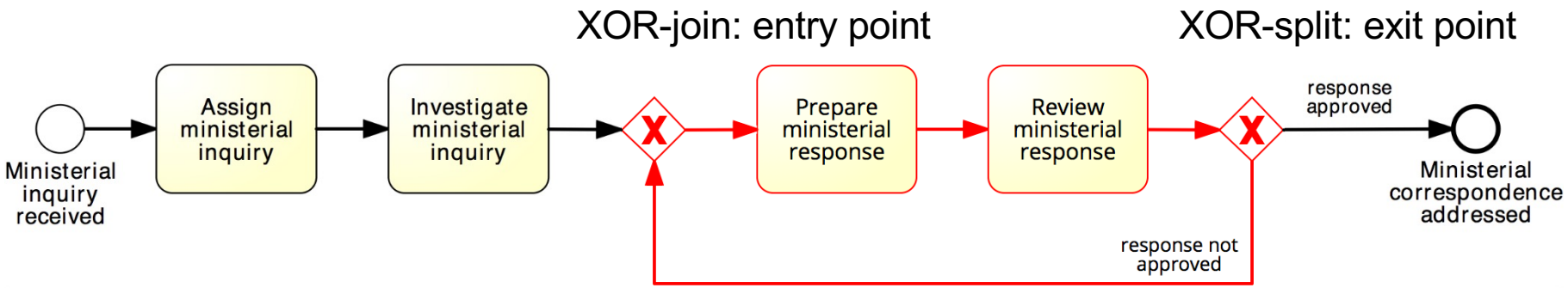
- Model in blocks
  - Pair up each AND-split with an AND-join and each XOR-split with a XOR-join, whenever possible
  - Exception: sometimes a XOR-split leads to two end events – different outcomes (cf. order management example)

Rework and repetition

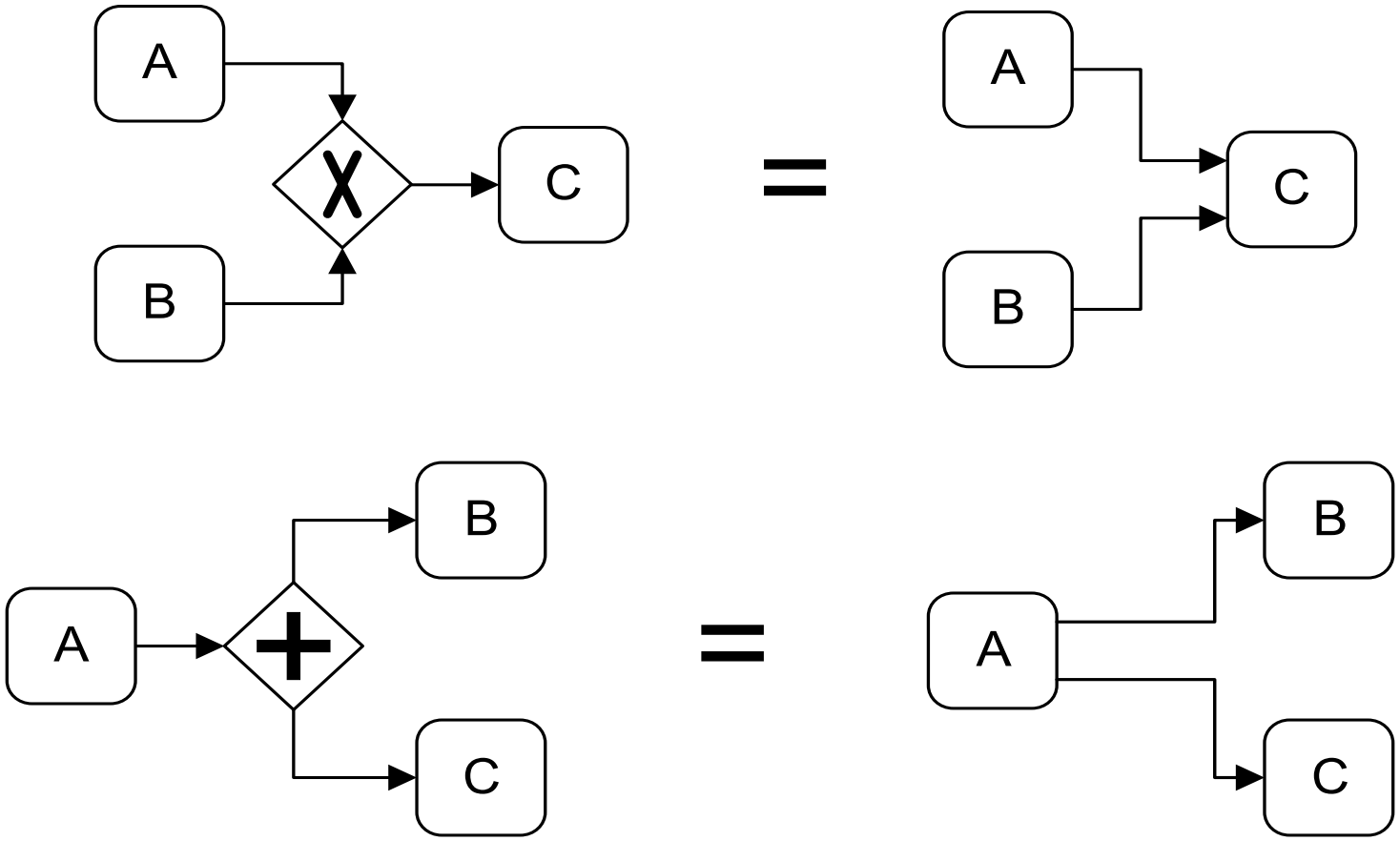
Address ministerial correspondence

In the minister's office, when a ministerial inquiry has been received, it is registered into the system. Then the inquiry is investigated so that a ministerial response can be prepared.

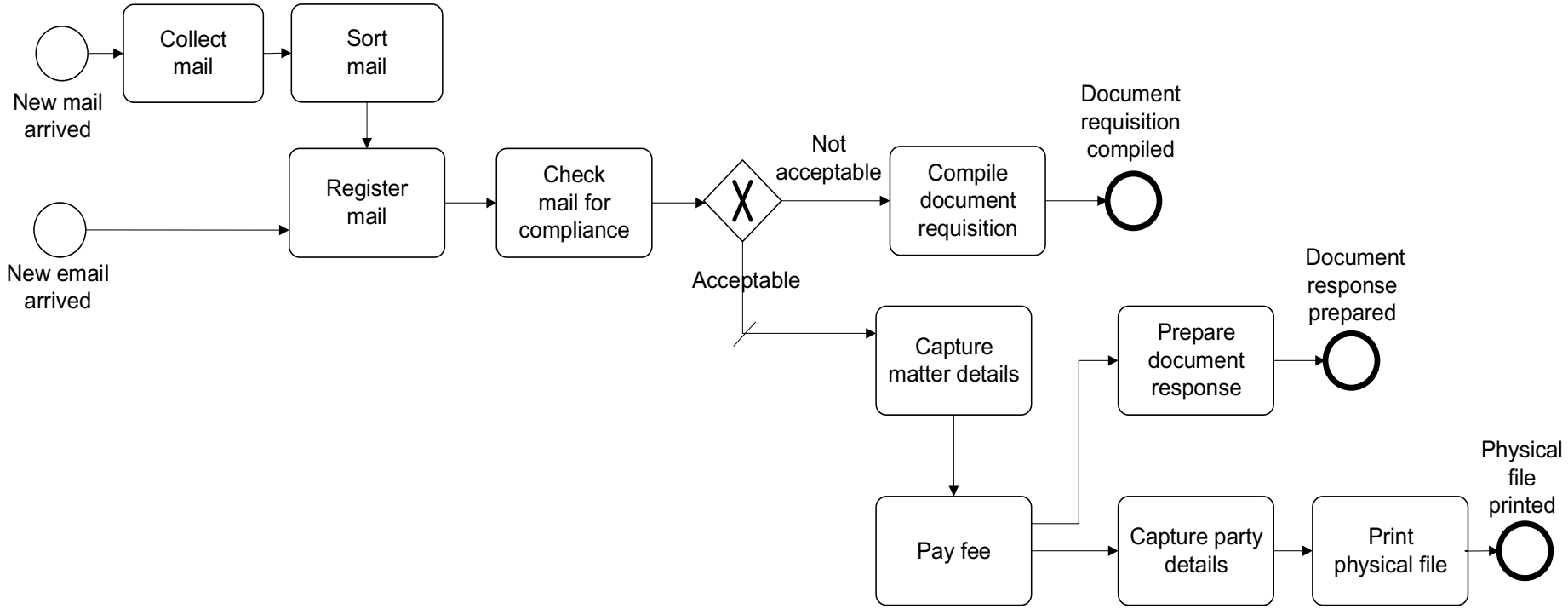
The finalization of a response includes the preparation of the response itself by the cabinet officer and the review of the response by the principal registrar. If the registrar does not approve the response, the latter needs to be prepared again by the cabinet officer for review. The process finishes only once the response has been approved.



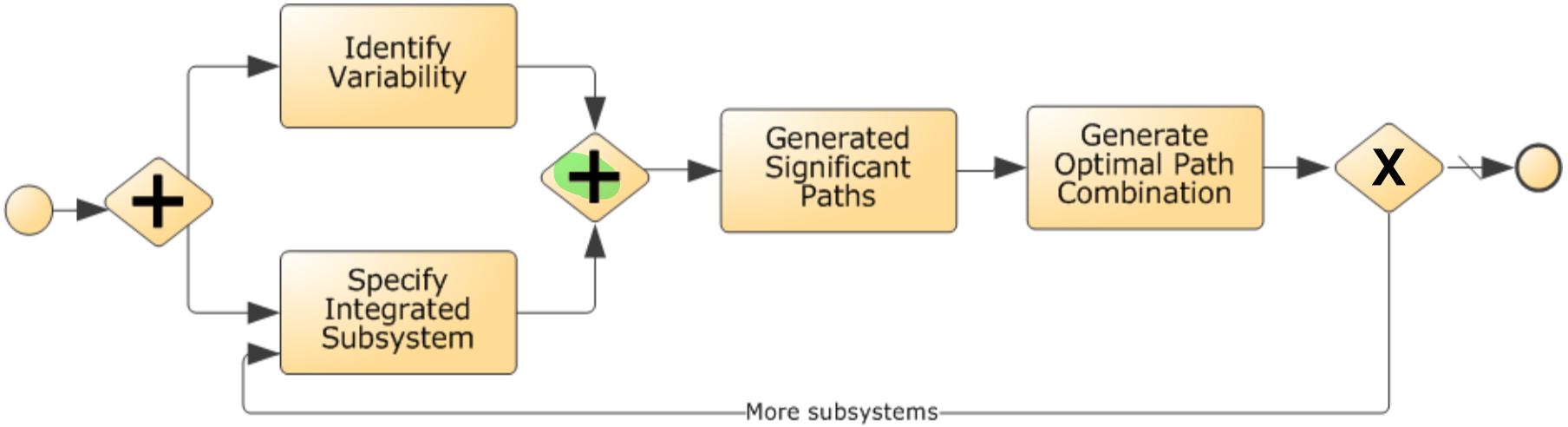
Quick Note: Implicit vs. explicit gateways



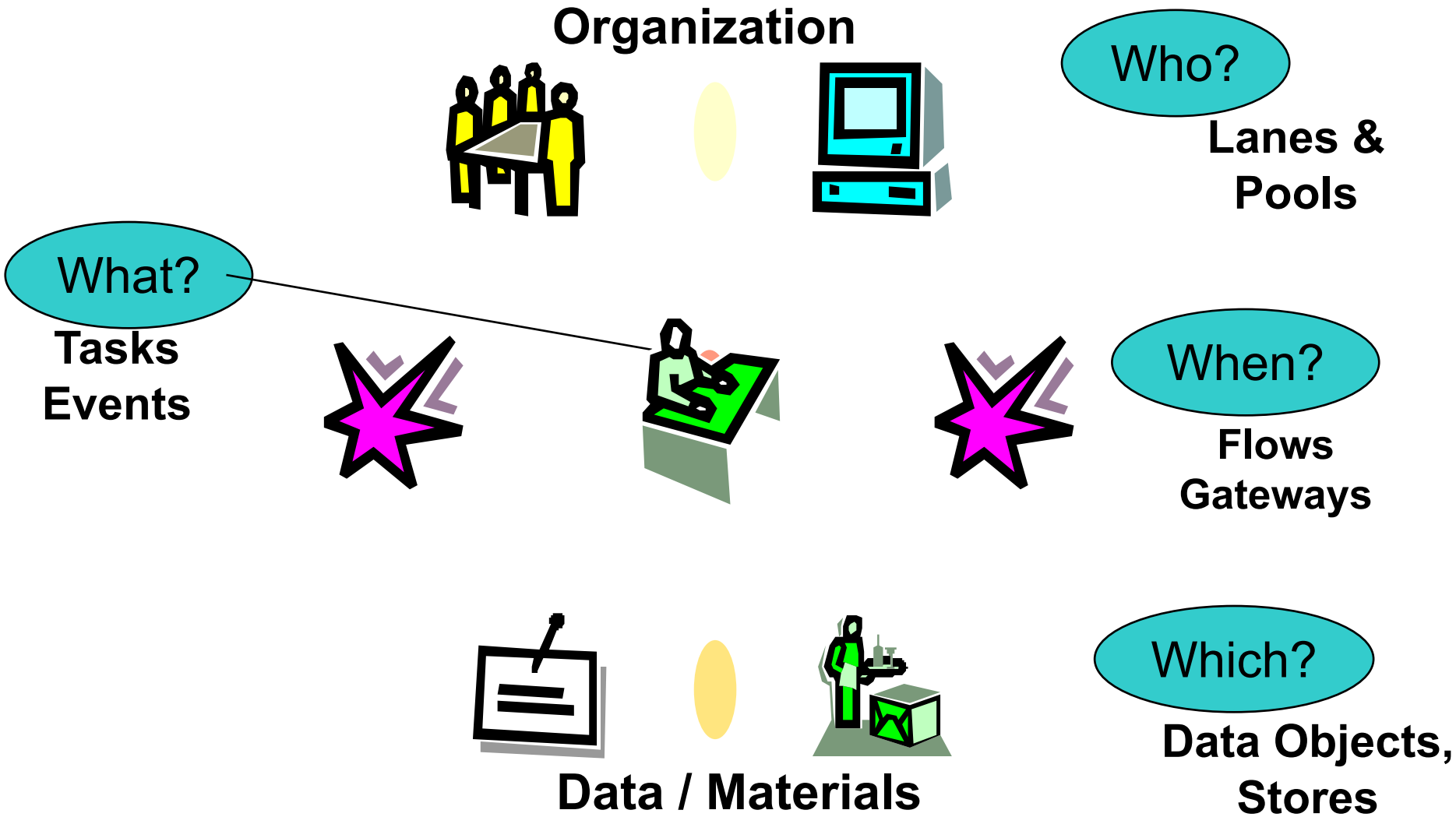
How this process starts? How it ends?



What's wrong with this model? How to fix it?



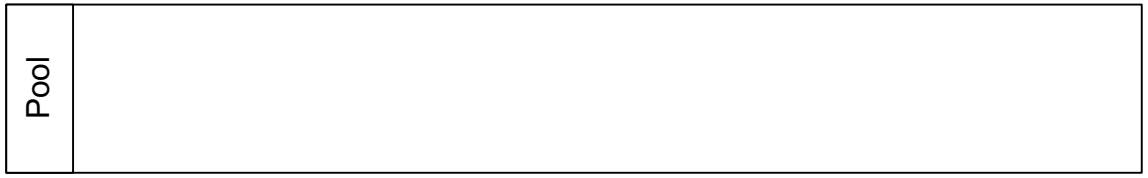
Process Modelling Viewpoints



Organizational Elements in BPMN – Pools & Lanes

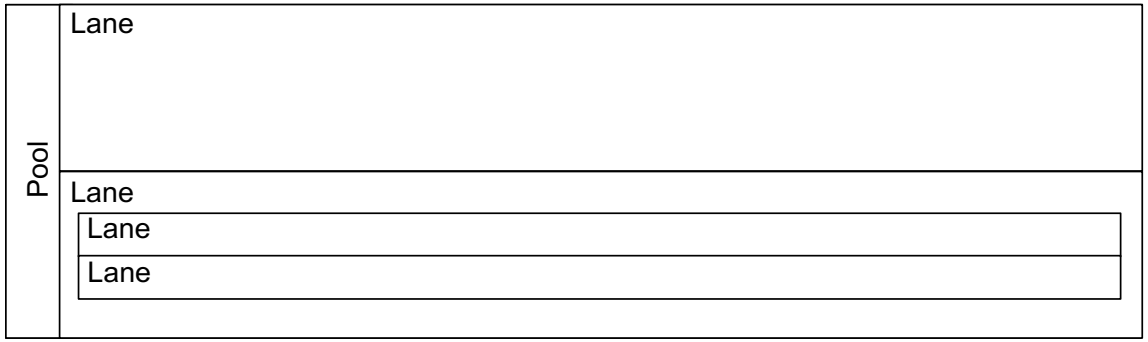
Pool

Captures a resource class. Generally used to model a business party (e.g. a whole company)

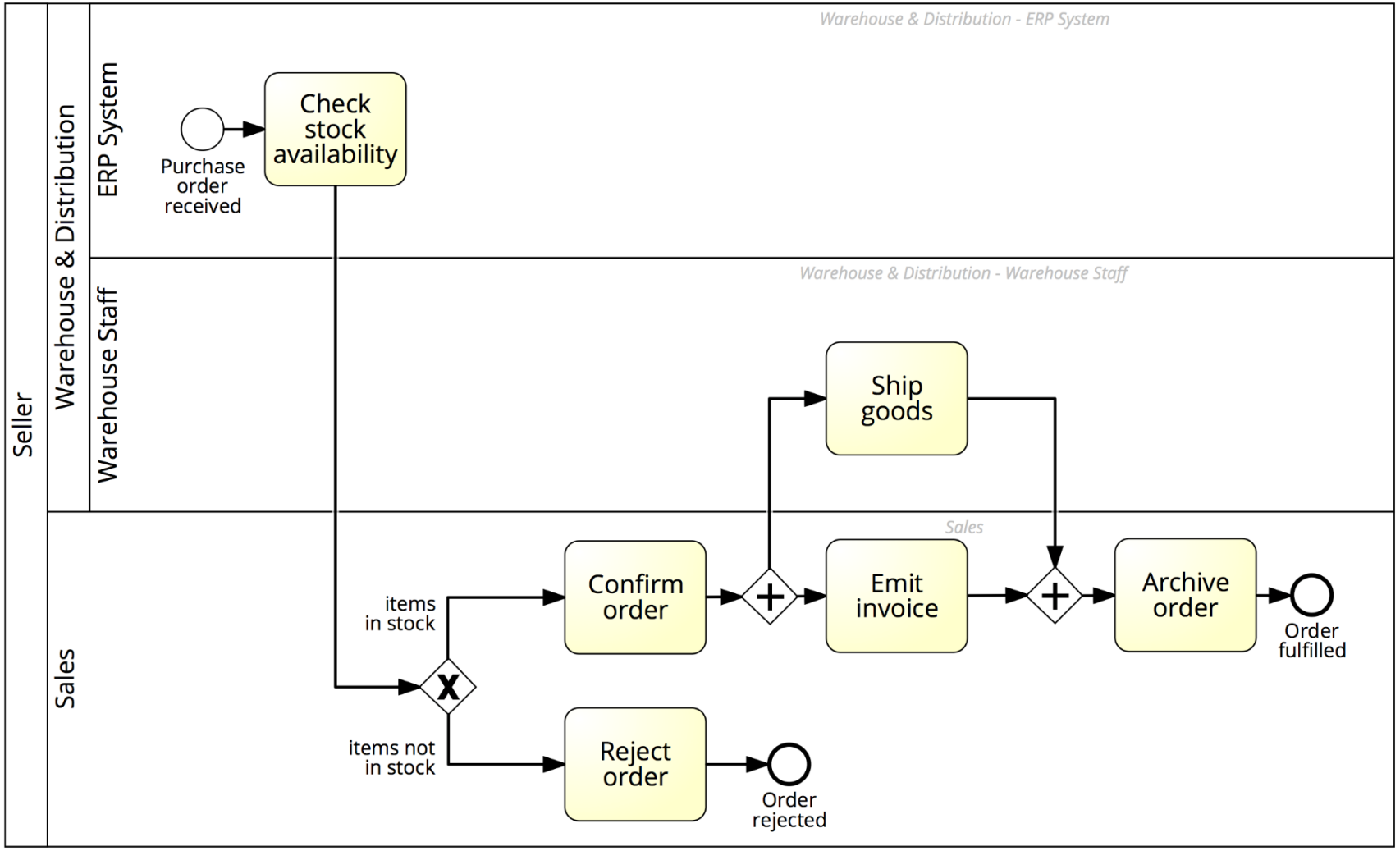


Lane

A *resource sub-class* within a pool. Generally used to model departments (e.g. shipping, finance), internal roles (e.g. Manager, Associate), software systems (e.g. ERP, CRM)

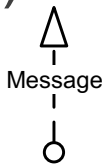


Order-to-cash process with lanes



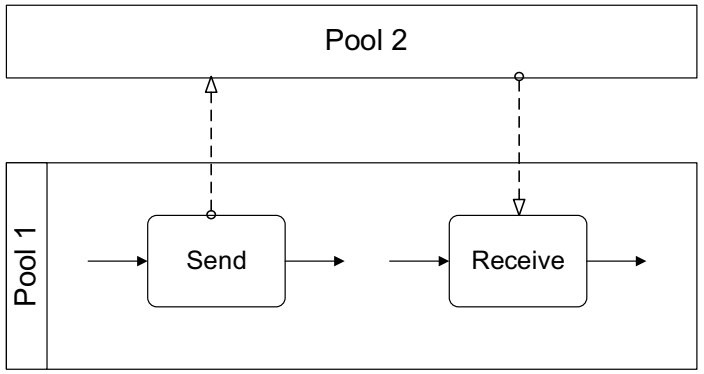
Message Flow

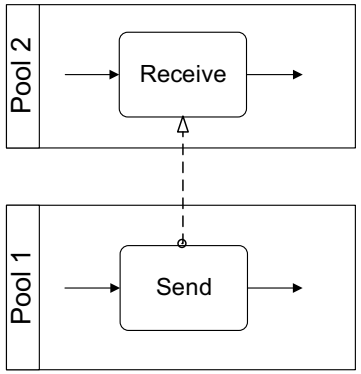
A *Message Flow* represents a flow of information between two process parties (Pools)



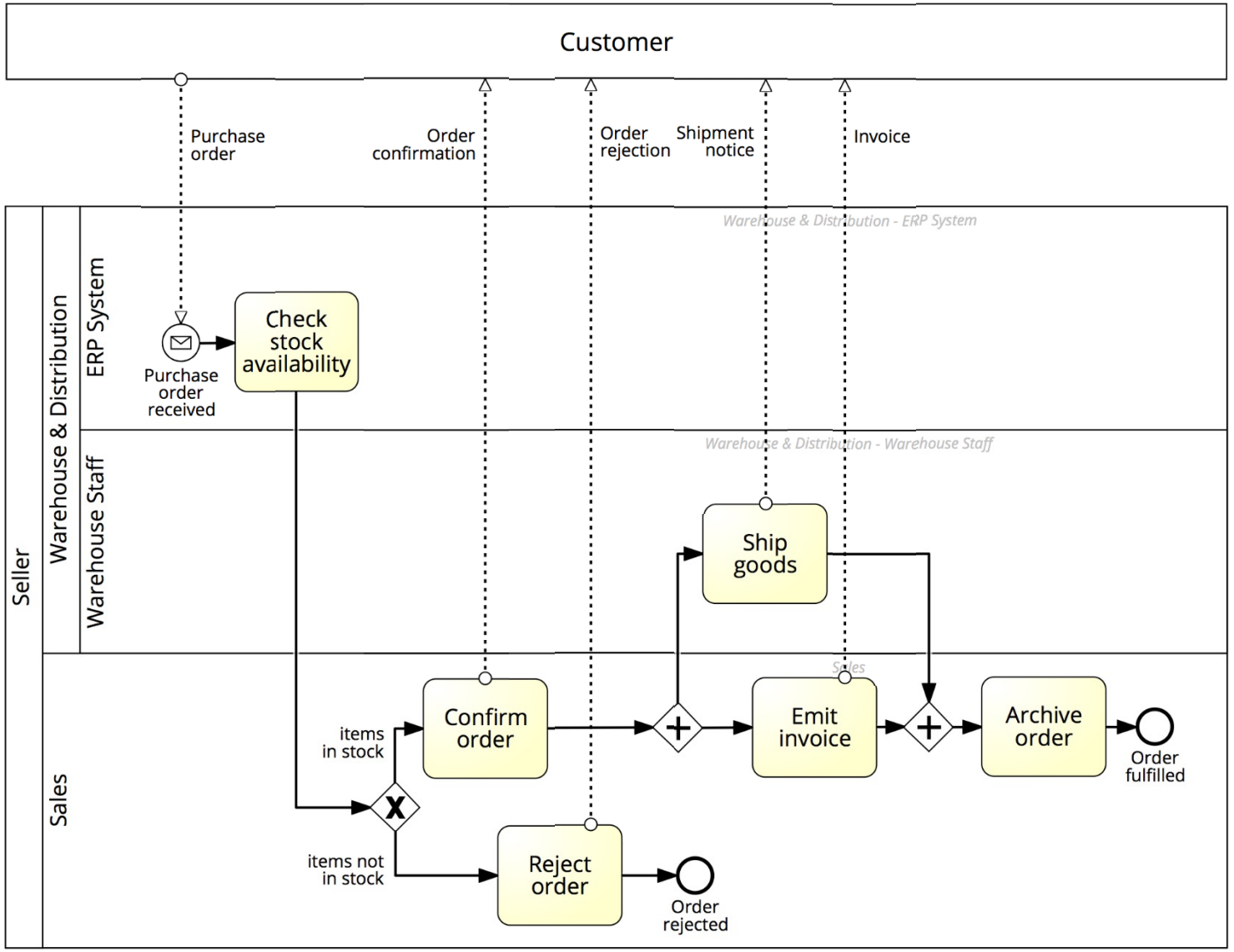
A Message Flow can connect:

- directly to the boundary of a Pool ➔ captures an *informative* message to/from that party
- to a specific activity or event within that Pool ➔ captures a message that triggers a specific activity/event within that party



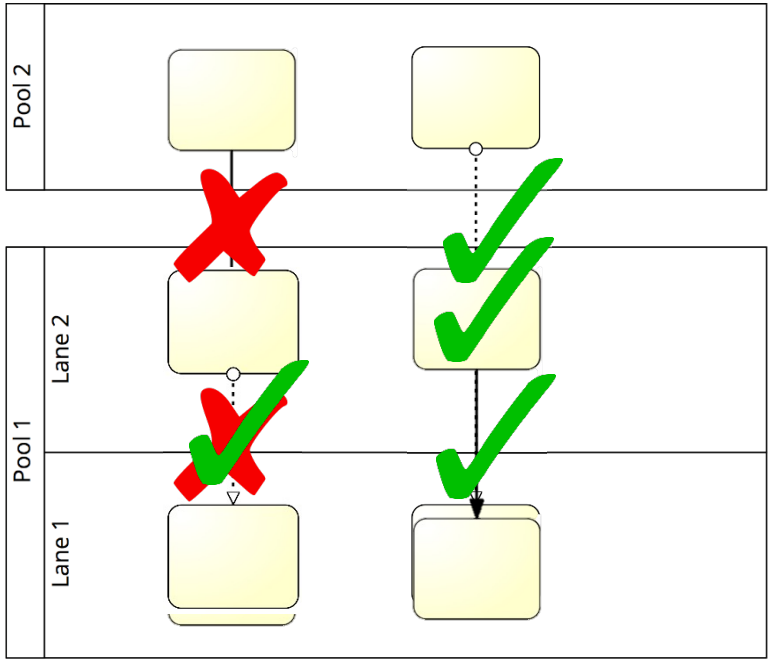


Order-to-cash process with a black-box customer pool



Pools, Lanes and Flows: syntactic rules

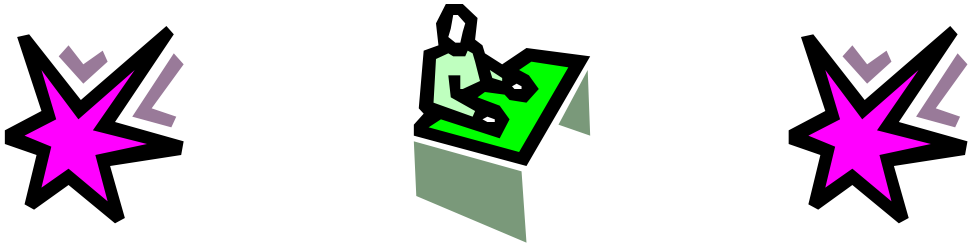
1. A Sequence Flow **cannot** cross the boundaries of a Pool (message flows can)
2. Both Sequence Flow and Message Flow **can cross** the boundaries of Lanes
3. A Message Flow **cannot connect** two flow elements within the same pool

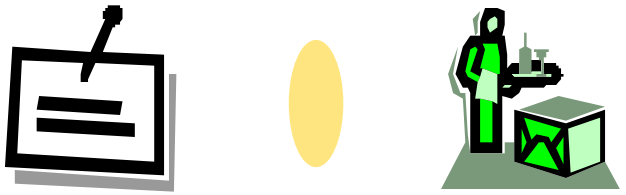


One more guideline…

- Start modeling with one single “white-box” pool
  - Initially, put the events and tasks in only one pool – the pool of the party who is running the process
  - Leave all other pools “black-boxed”
  - Once you have modeled this way, and once the process diagram inside the white-box pool is complete, you can model the details (events and tasks) in the other pools <u>if that is useful</u>.
  - In this course we will only model processes with one single white-box pool – all other pools are black-box

Process Modelling Viewpoints



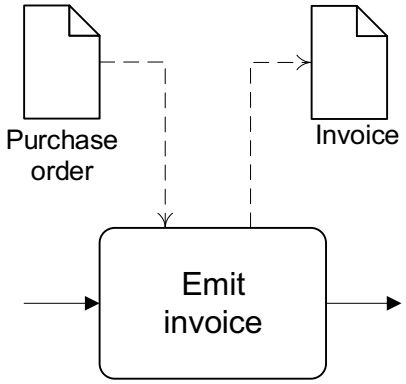


Data / Materials

Which?

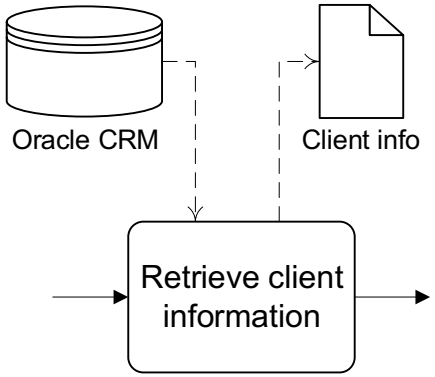
Data Objects, Stores

BPMN Information Artifacts



A *Data Object* captures an artifact required (input) or produced (output) by an activity.

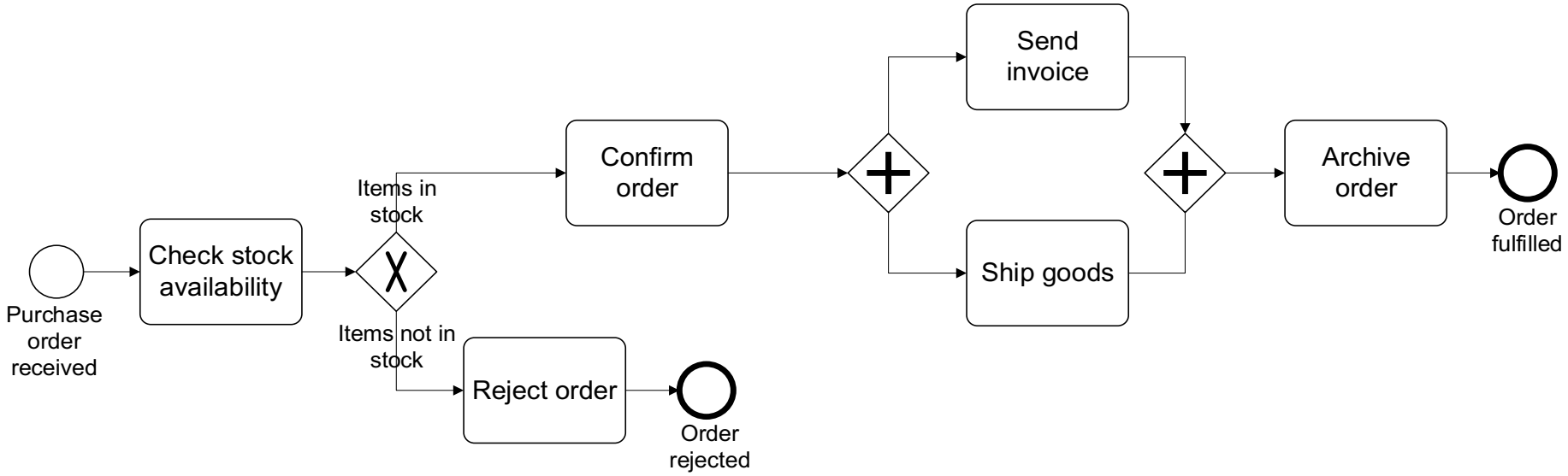
- Can be physical or electronic



A *Data Store* is a place containing data objects that must be persisted beyond the duration of a process instance.

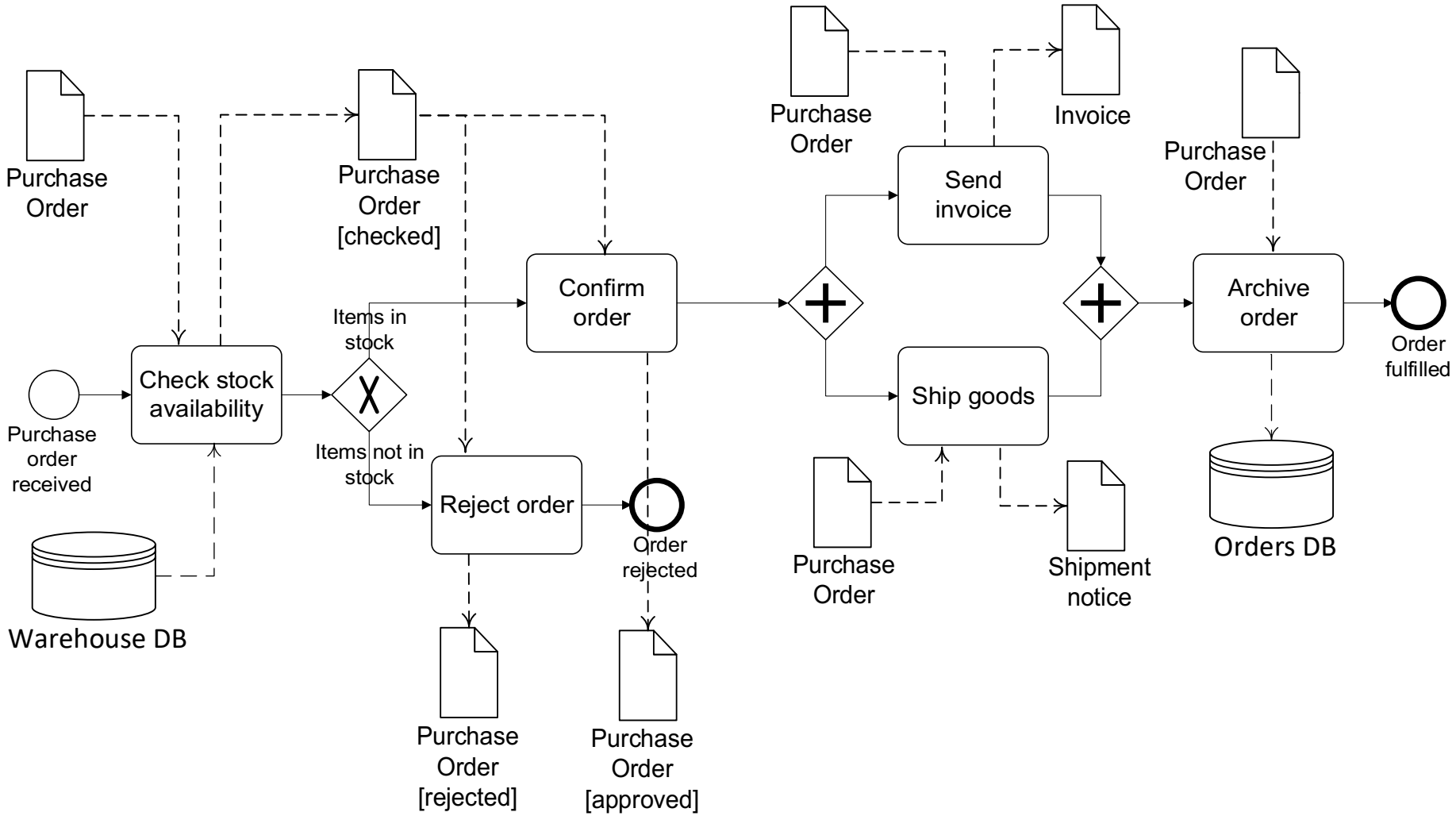
It is used by an activity to store (as output) or retrieve (as input) data objects.

Order-to-cash process, again



The purchase order document serves as an input to the stock availability check. Based on the outcome of this check, the status of the document is updated, either to “approved” or “rejected”. If the order is approved, an invoice and a shipment notice are produced.

Model with information artifacts

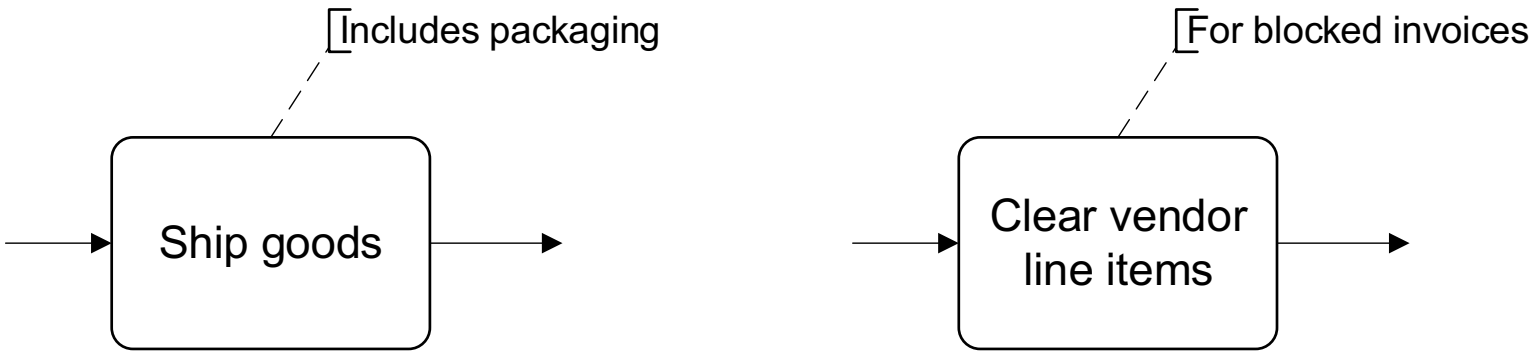


Beware: This diagram is a too detailed. It is for illustration purposes. In practice, try to only model the most important data objects and associations. Keep the model readable.

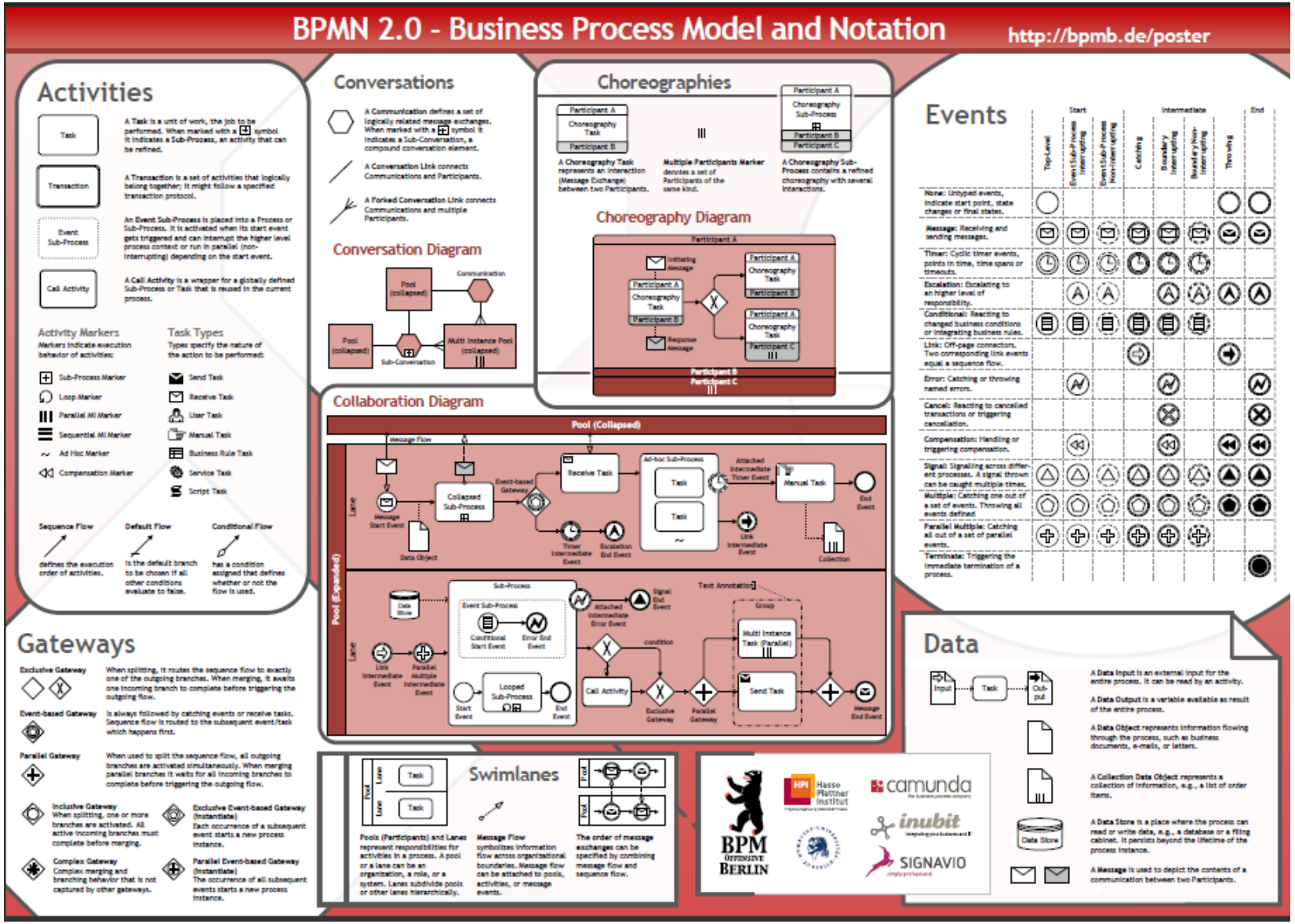
A Final Note: BPMN Text Annotations

A *Text Annotation* is a mechanism to provide additional text information to the model reader

- **Doesn't affect** the flow of tokens through the process



BPMN Poster


BPMN 2.0 - Business Process Model and Notation
http://bpmb.de/poster
Activities
Conversations
Choreographies
Events
Gateways
Data
Swimlanes
Conversation Diagram
Choreography Diagram
Collaboration Diagram
BPM Berlin
camunda
inubit
SIGNAVIO

Acknowledgements

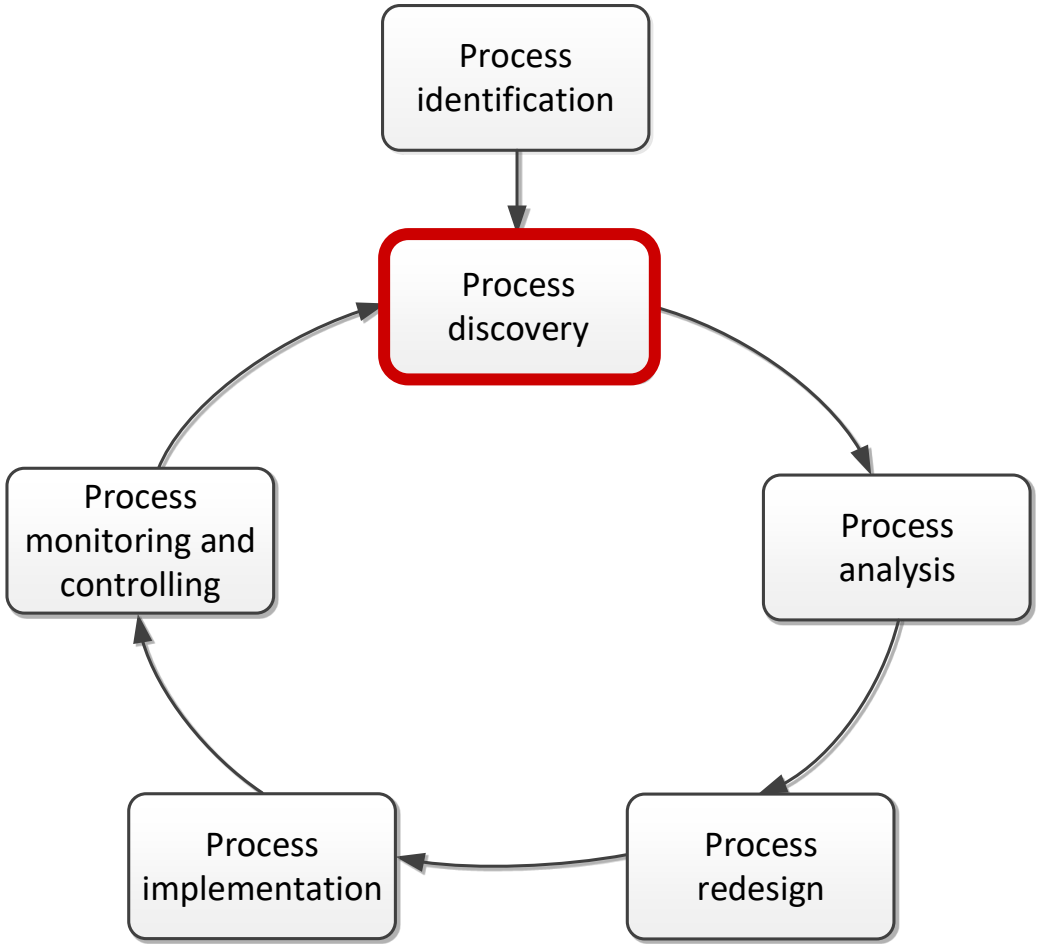
- All material comes from Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A. Reijers, authors of the “Fundamentals of Business Process Management” book.

INFO-H420
Management of Data Science and Business Workflows
Part I
2. Process Modeling

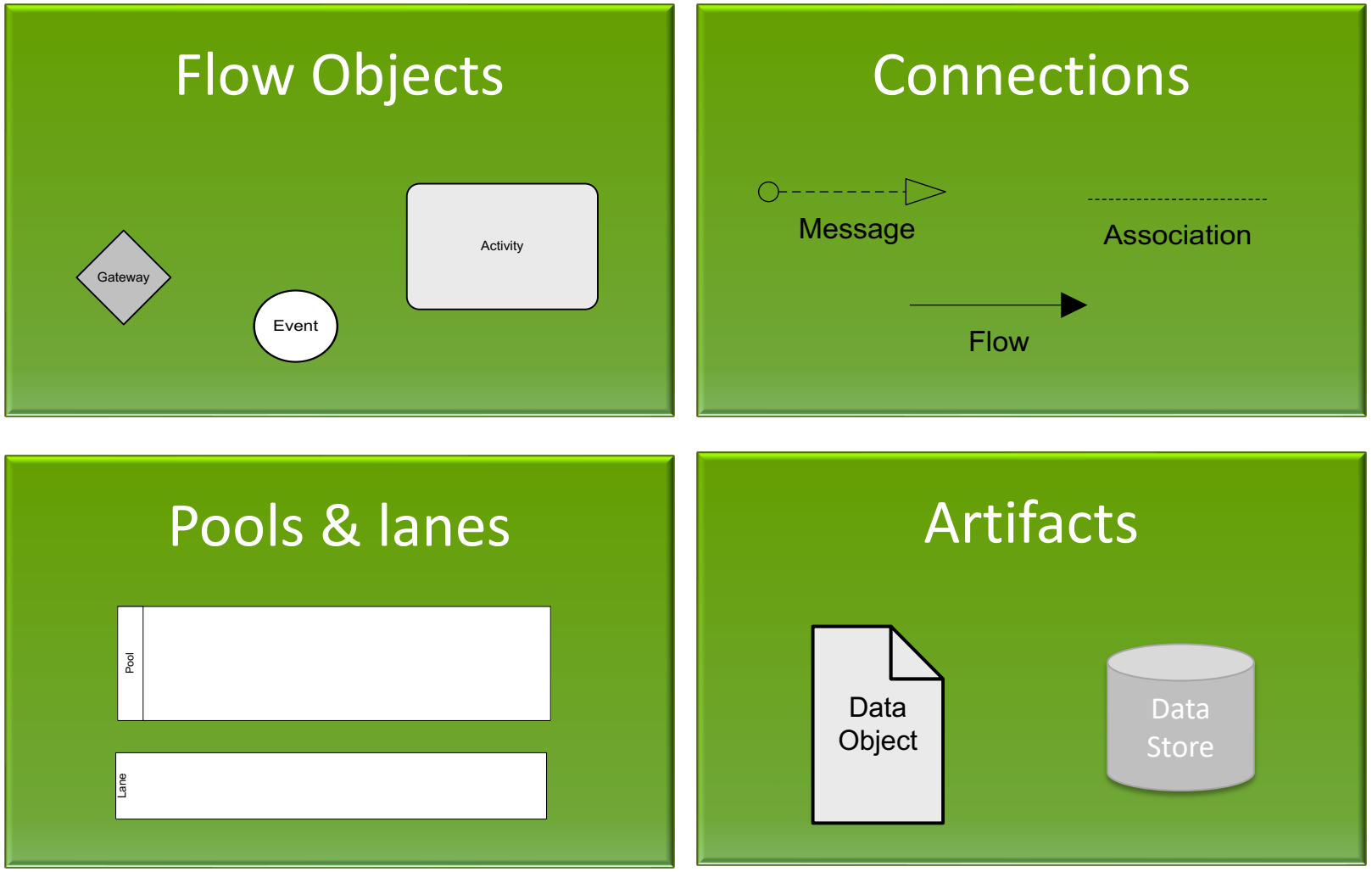
Dimitris SACHARIDIS

2023-2024

BPM structure



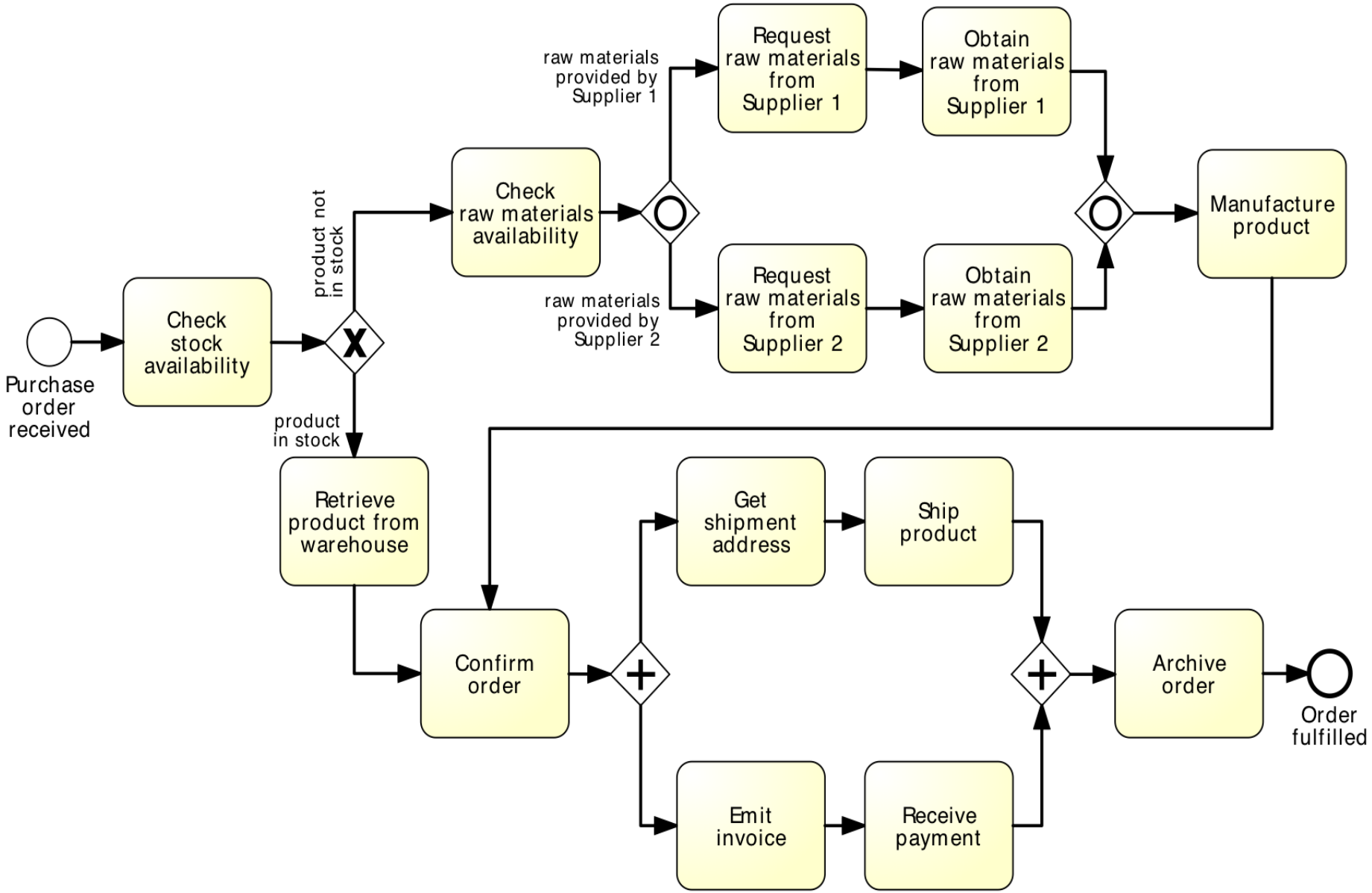
BPMN Main Elements - Recap



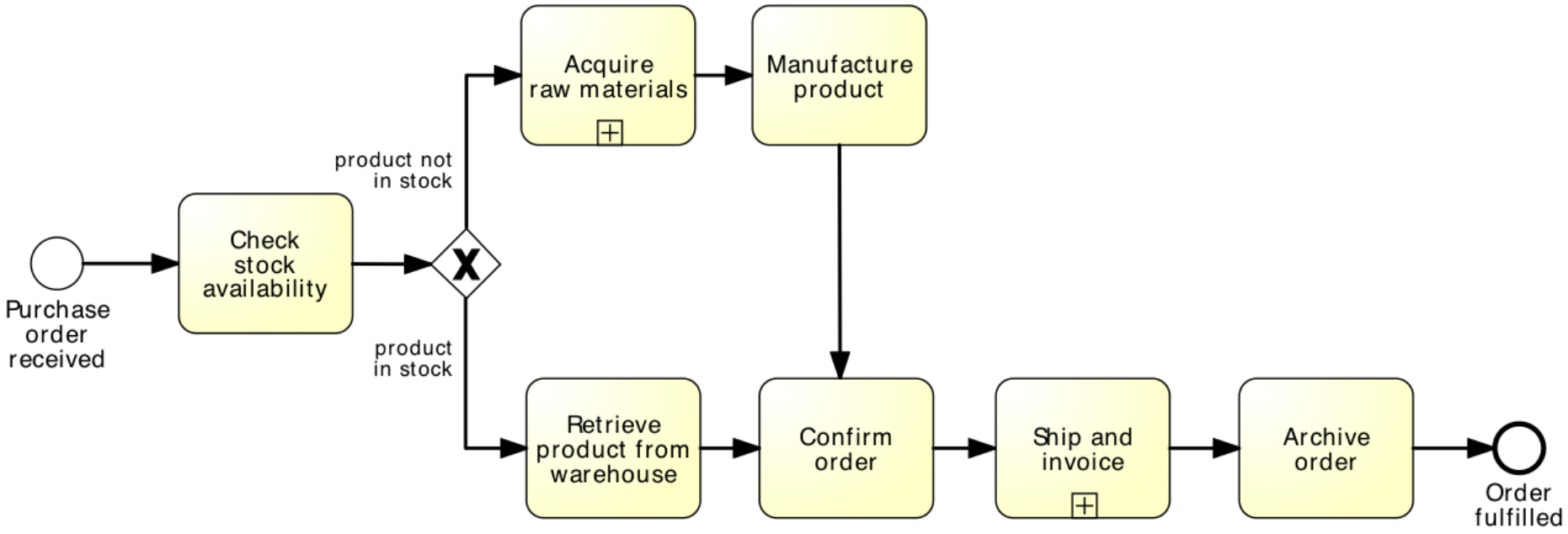
BPMN Gateways

Exclusive (XOR)	Parallel (AND)	Inclusive (OR)
• Exclusive decision take one branch • Exclusive merge Proceed when one branch has completed	• Parallel split take all branches • Parallel join proceed when all incoming branches have completed	• Inclusive decision take one or several branches depending on conditions • Inclusive merge proceed when all active incoming branches have completed

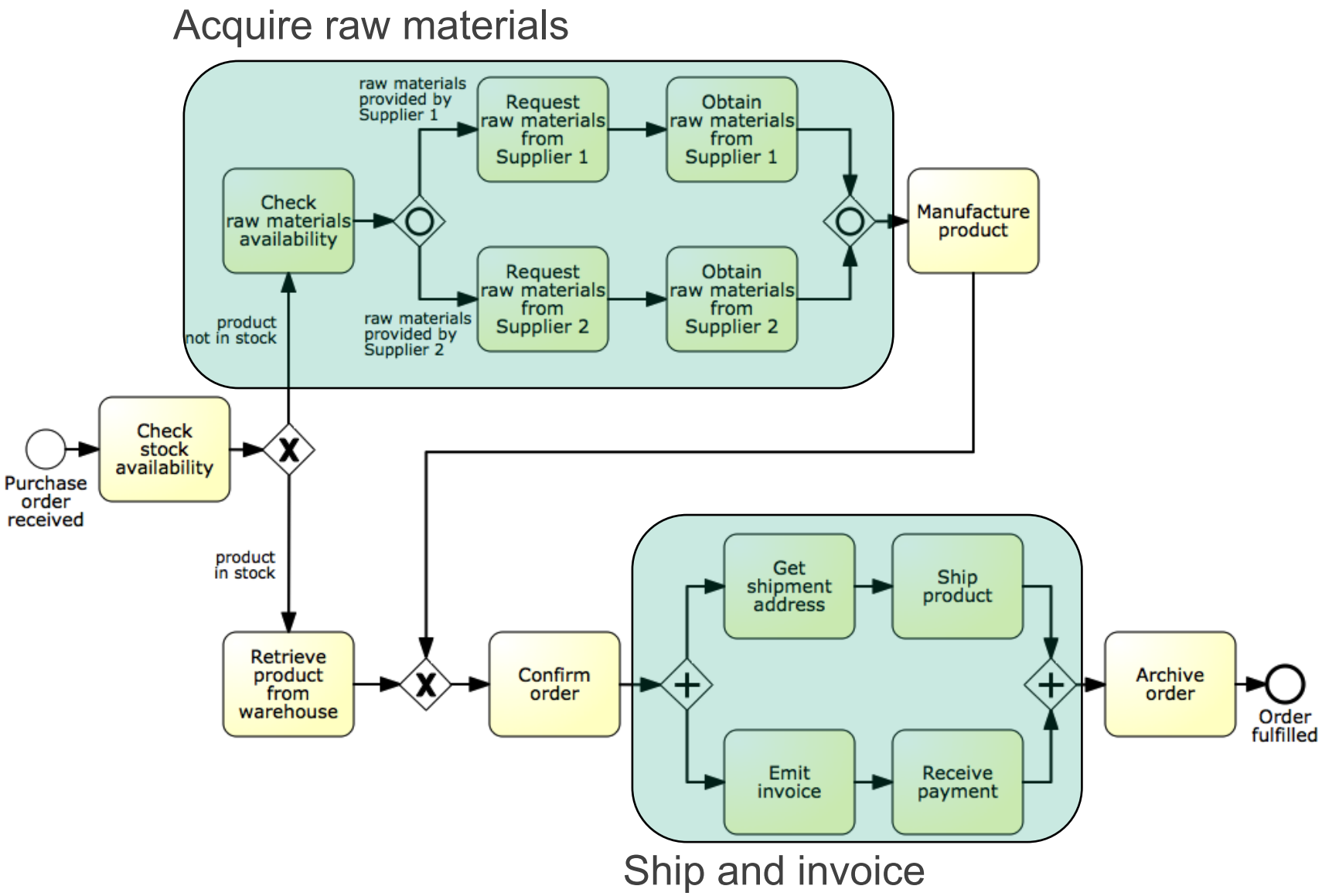
Critique this model



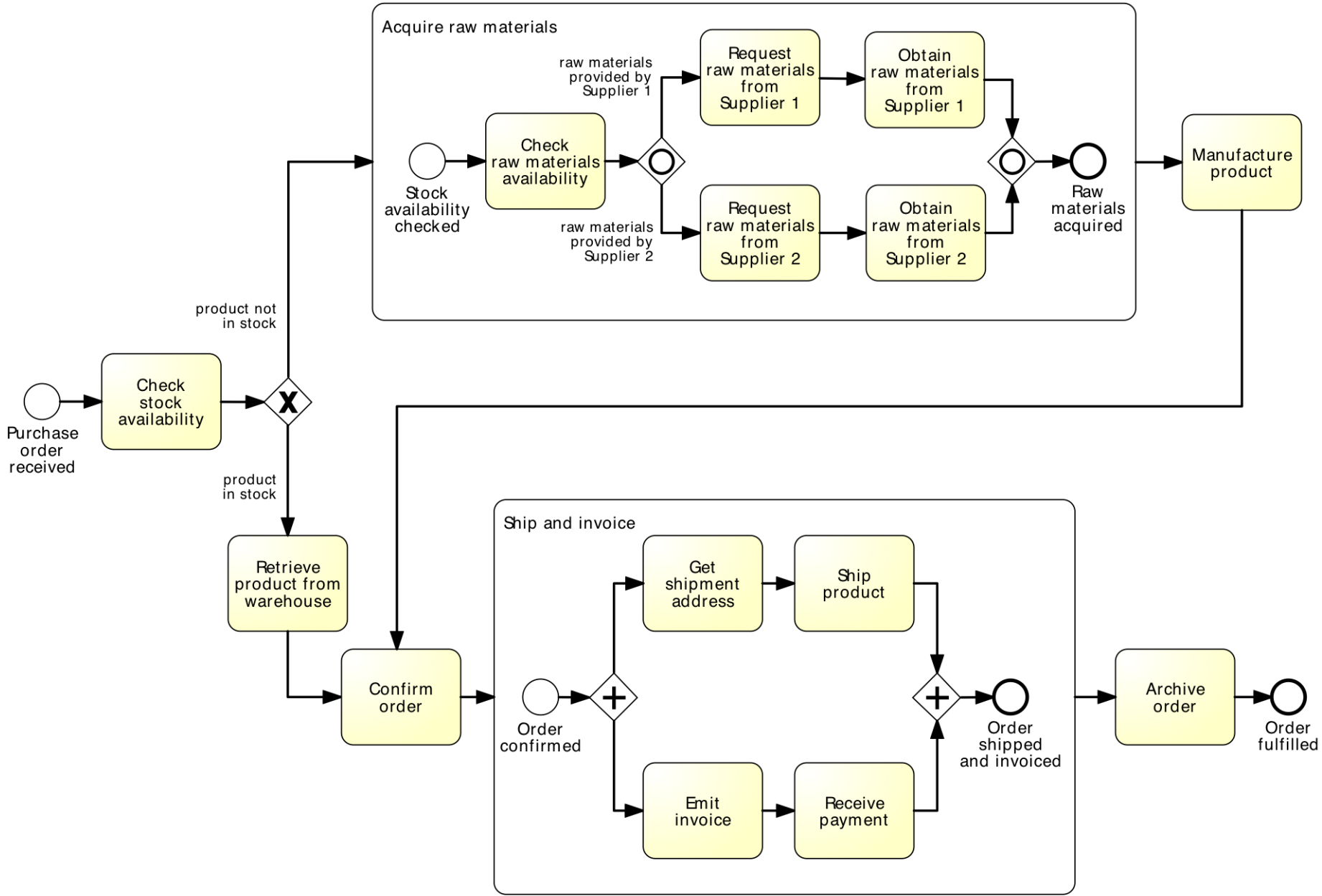
Is this better?



Identifying sub-processes



Using the Expanded Sub-Process Notation



Sub-processes

- An activity in a process can invoke a separate sub-process
- Use this feature to:
  1. Decompose large models into smaller ones, making them easier to understand and maintain

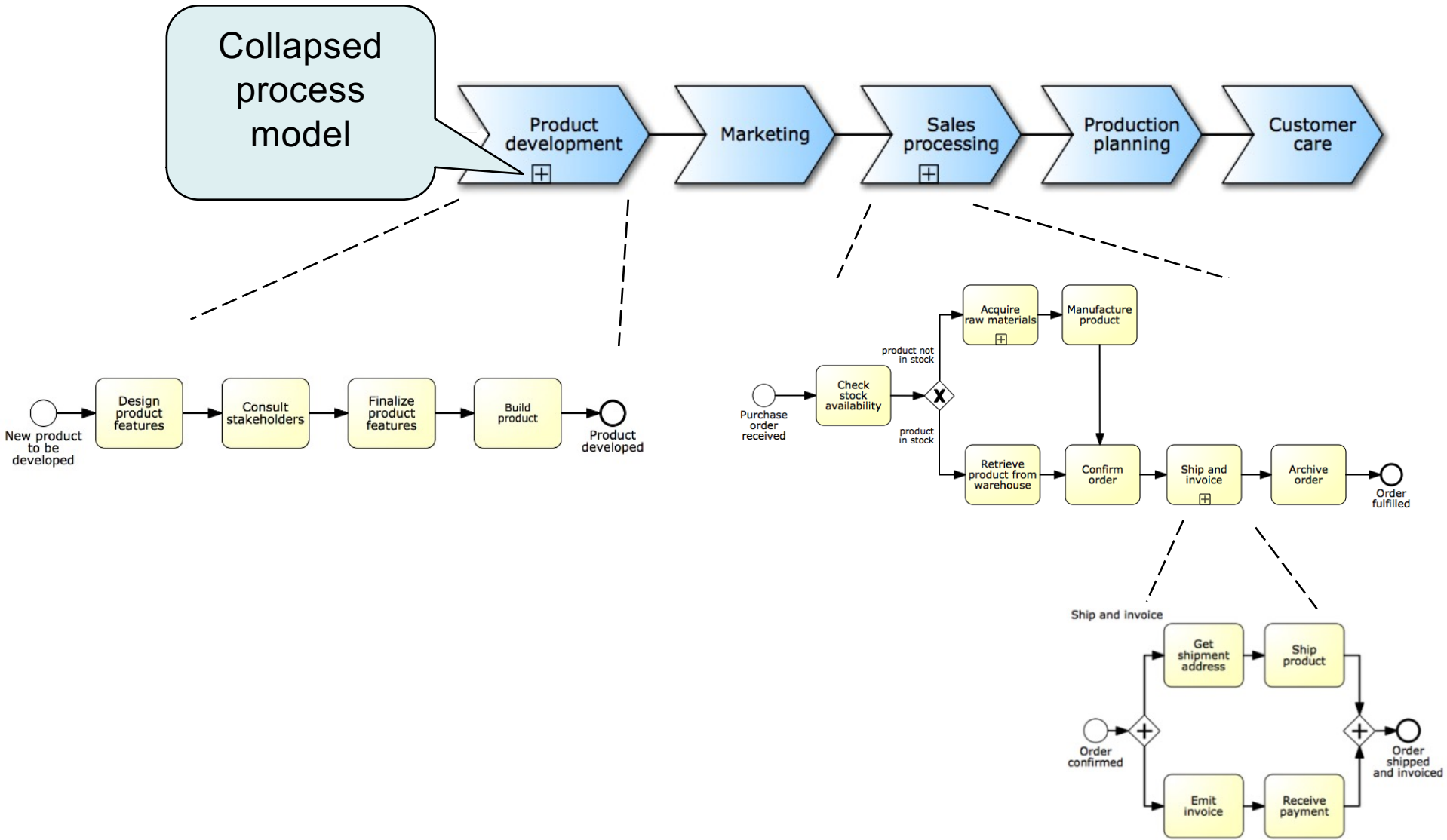
Guideline: Multi-level modeling

- Level 1: **value chain**
  - Simple linear description of the phases of the process
  - No gateways
  - Each activity chain is a sub-process
- Level 2+: expand each activity in the value chain, add incrementally the following:
  - Decisions, handoffs (lanes, pools)
  - Parallel gateways, different types of events
  - Data objects & data stores
  - And as much detail as you need, and no more

Guideline: Multi-level modeling (cont.)

- At each level, decompose according to:
  - Logical milestones towards achieving the outcome of the process
  - Major objects used in the process
- Decompose until processes are of “reasonable” size
  - e.g., up to 20 nodes (tasks+events+gateways) per model

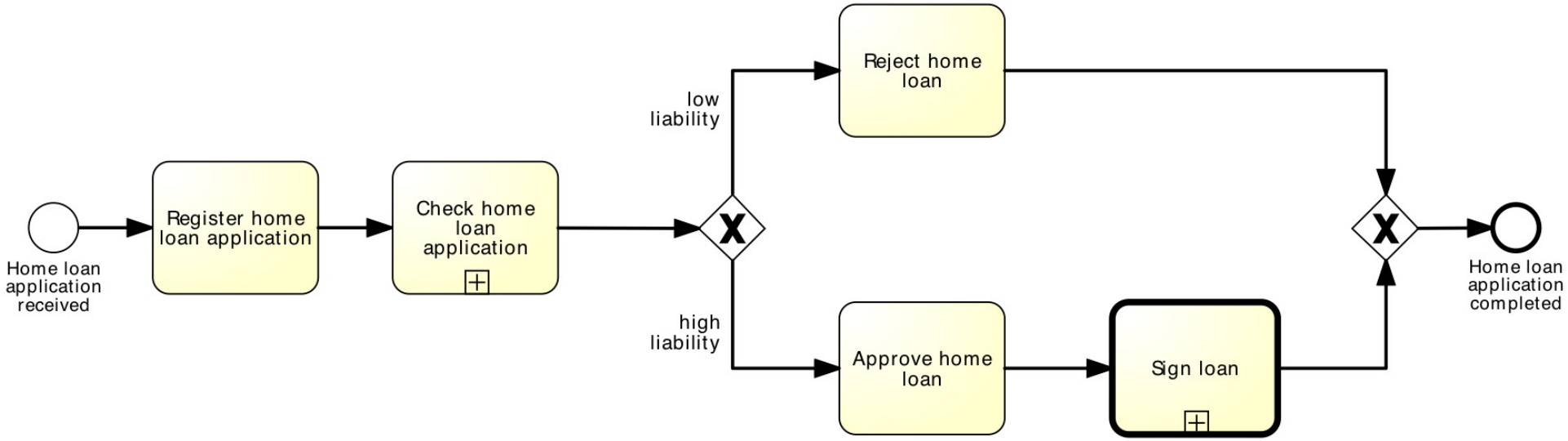
Non-BPMN value chain “chevron” notation

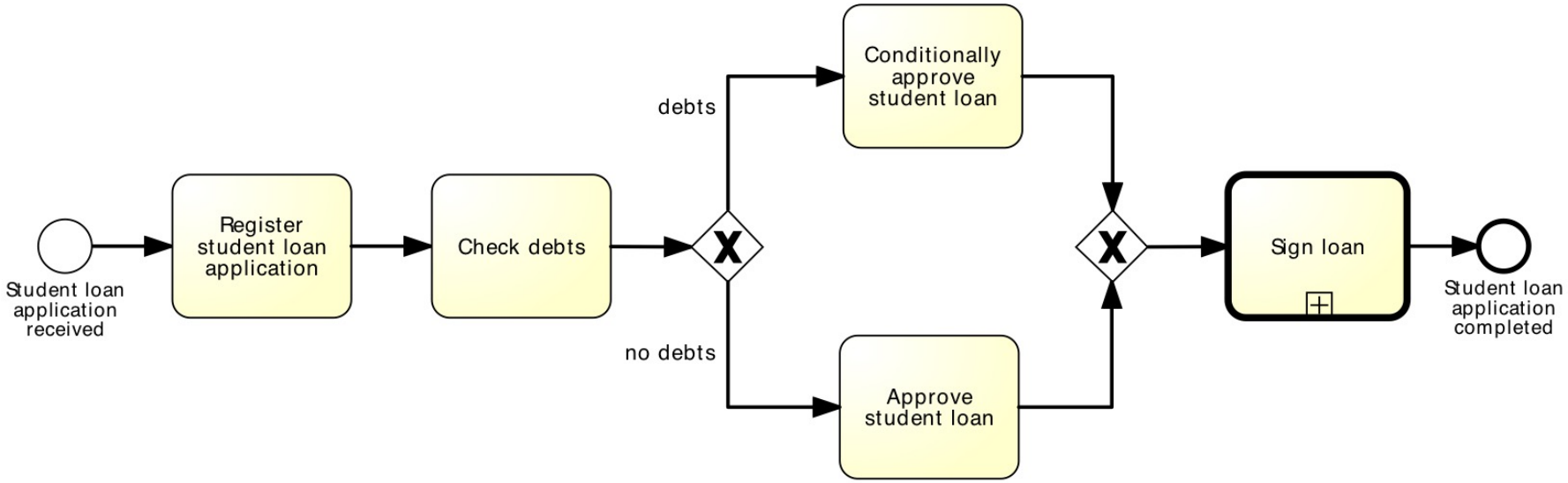


Sub-processes

- An activity in a process can invoke a separate sub-process
- Use this feature to:
  1. Decompose large models into smaller ones, making them easier to understand and maintain
  2. Share common fragments across multiple processes

Shared sub-process





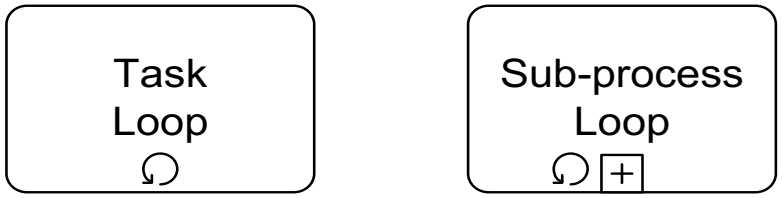
Sub-processes

- An activity in a process can invoke a separate sub-process
- Use this feature to:
  1. Decompose large models into smaller ones, making them easier to understand and maintain
  2. Share common fragments across multiple processes
  3. Delimit parts of a process that can be:
     - Repeated
     - Interrupted

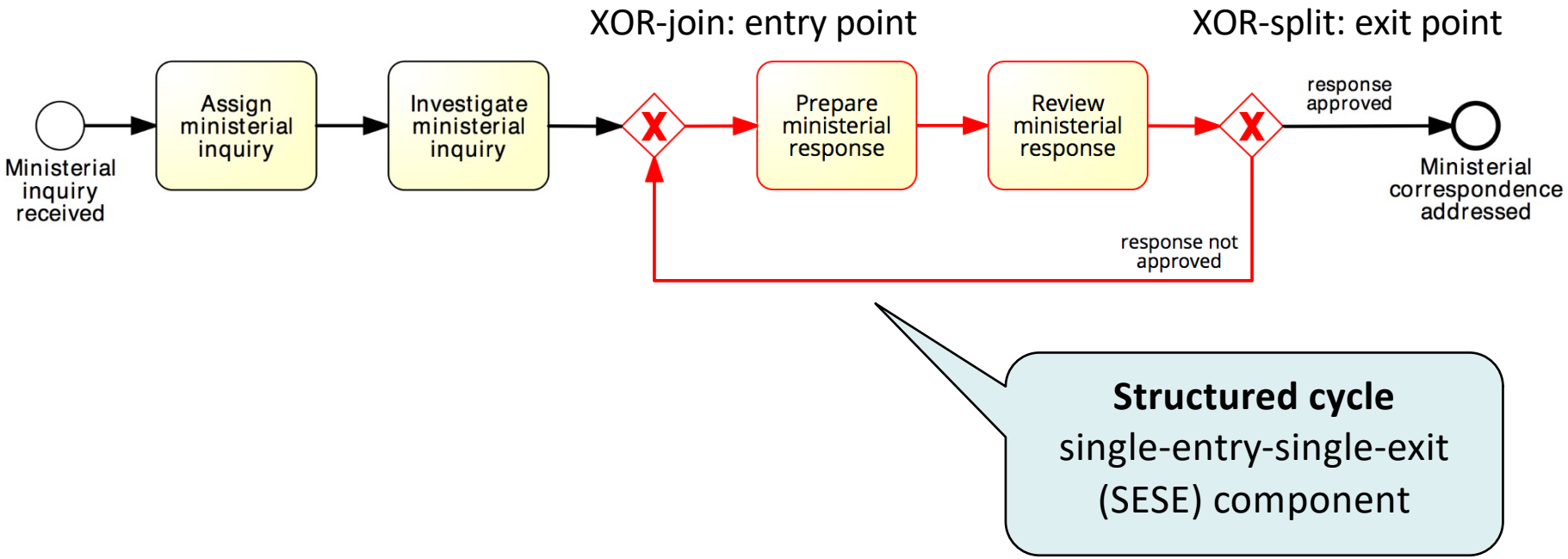
STRUCTURED REPETITION

Block-structured repetition: Activity loop

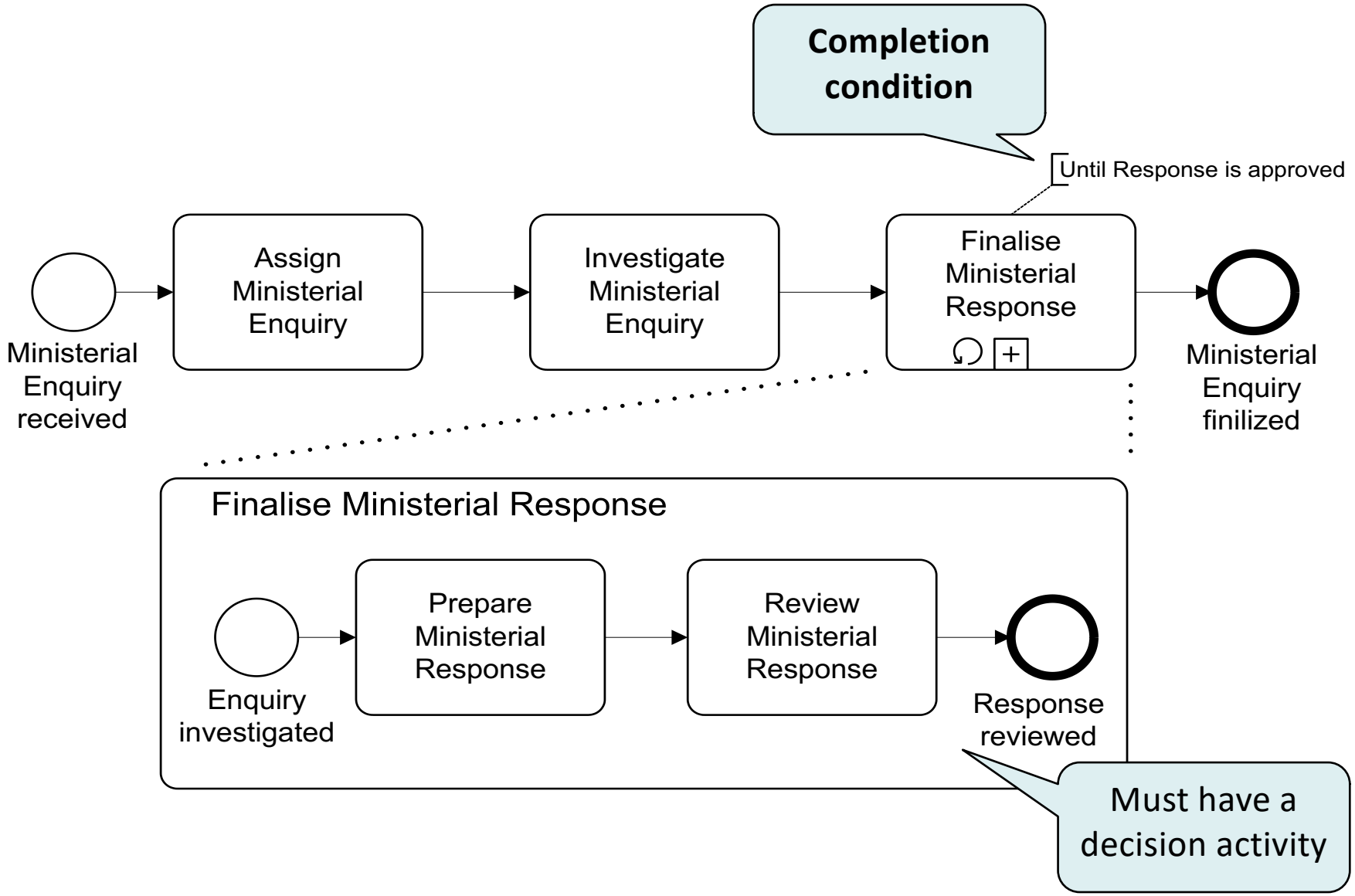
Activity loop markers allow us to state that a task or a sub-process may be repeated multiple times



More on rework and repetition

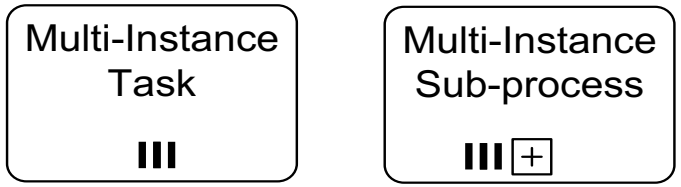


Example: block-structured repetition



Parallel repetition: multi-instance activity

The multi-instance activity provides a mechanism to indicate that an activity is executed ***multiple times concurrently***



Useful when the same activity needs to be executed for multiple entities or data items, such as:

- Request quotes from multiple suppliers
- Check the availability for each line item in an order separately
- Send and gather questionnaires from multiple witnesses in the context of an insurance claim

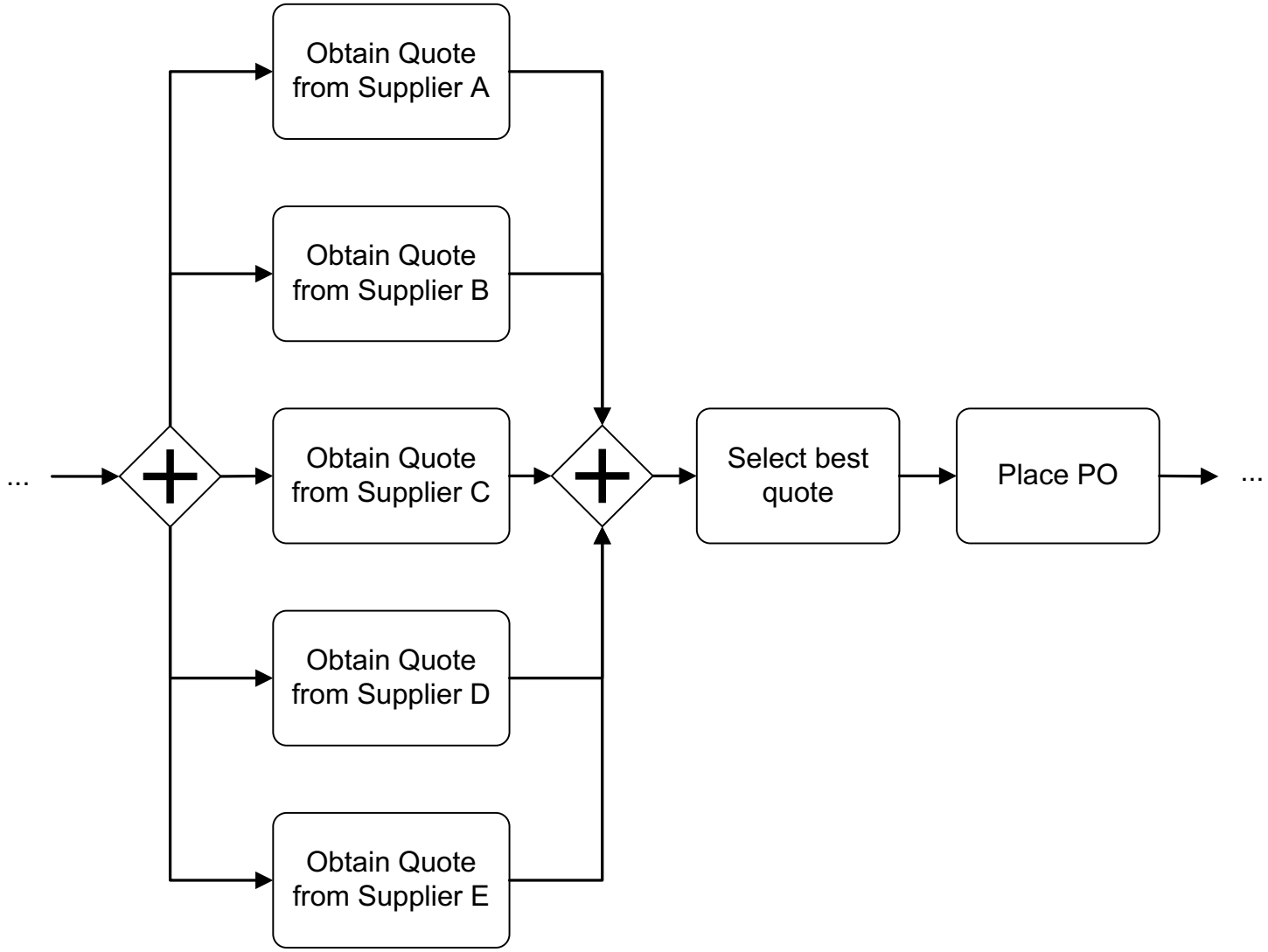
Example: multi-instance activity

Procurement

In procurement, typically a quote is to be obtained from all preferred suppliers (assumption: five preferred suppliers exist). After all quotes are received, they are evaluated and the best quote is selected. A corresponding purchase order is then placed.

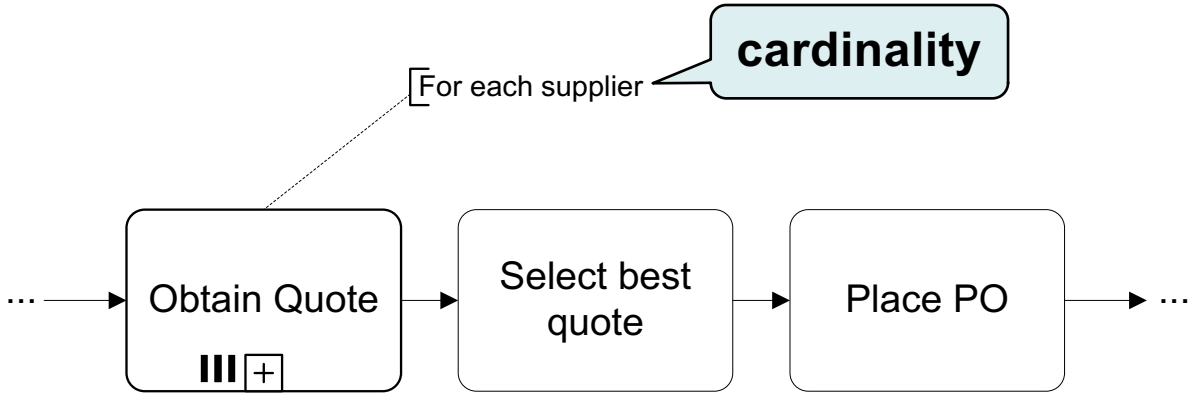
Is this elegant?

Procurement



Solution: with multi-instance activity

Procurement



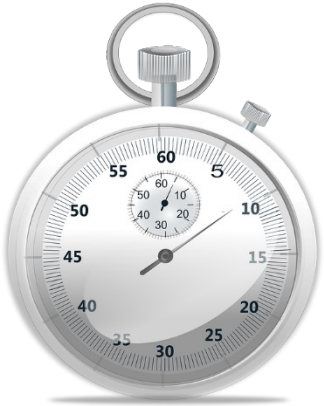
EVENTS

Events

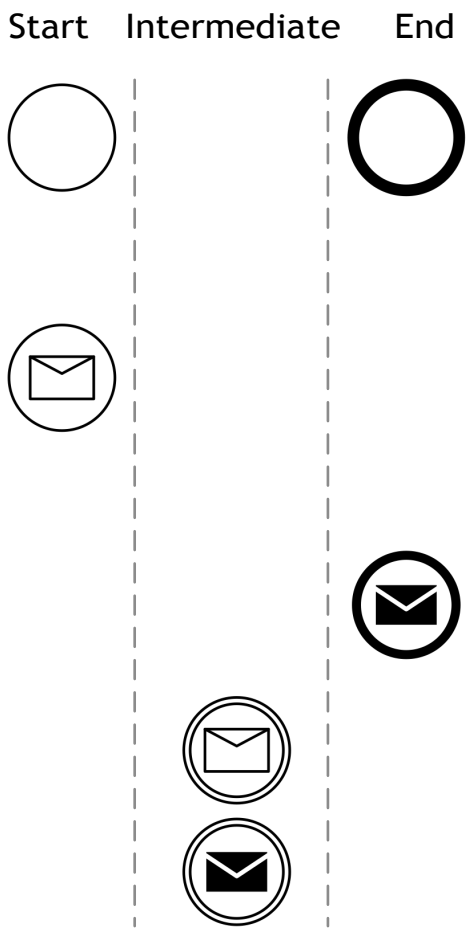
In BPMN, events model something instantaneous happening during the execution of a process

Types of event:

- Start
- Intermediate
- End



BPMN event types



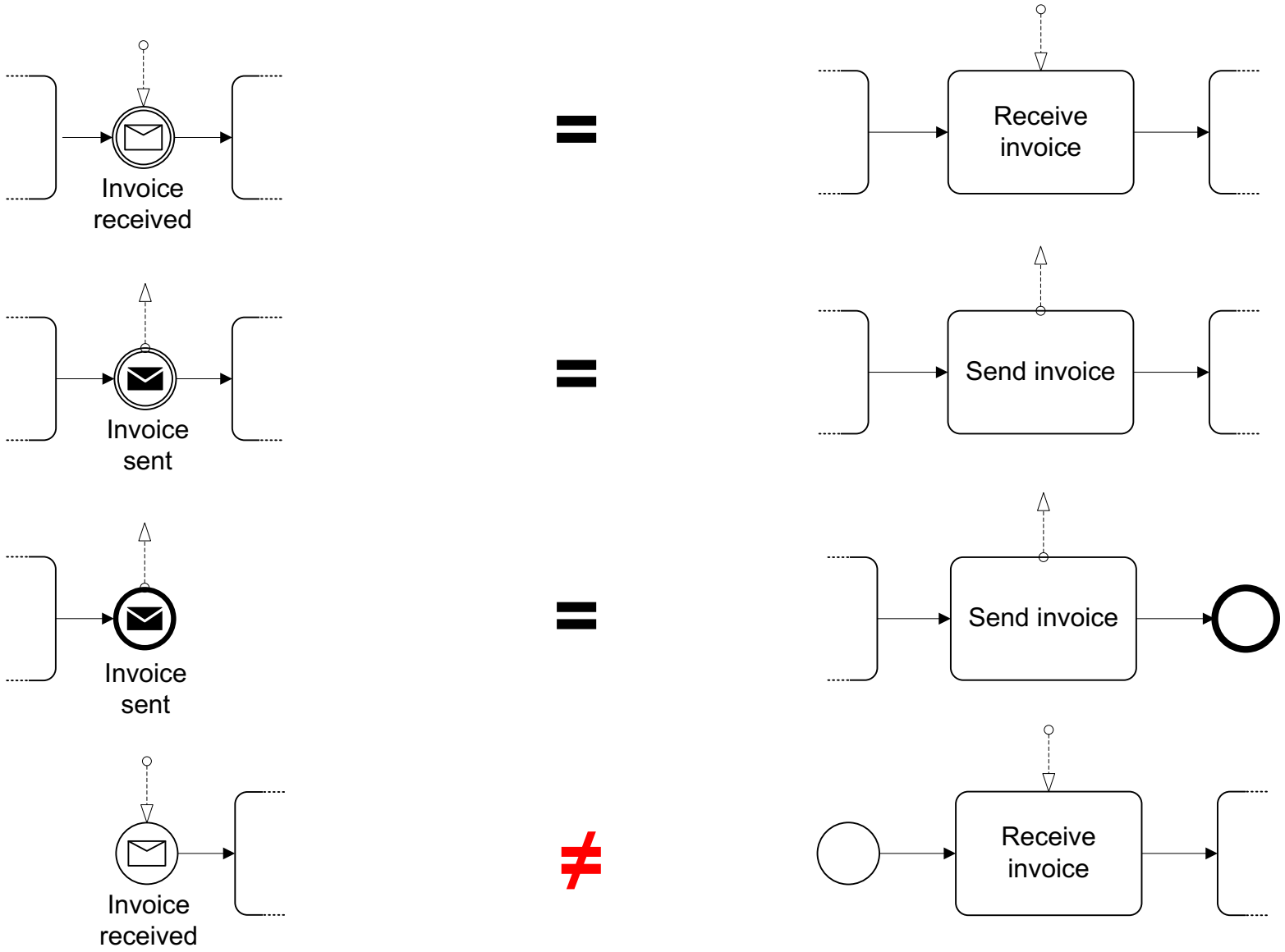
Untyped Event – Indicates that an instance of the process is created (start) or completed (end), without specifying the cause for creation/completion

Start Message Event – Indicates that an instance of the process is created when a message is **received**

End Message Event – Indicates that an instance of the process is completed when a message is **sent**

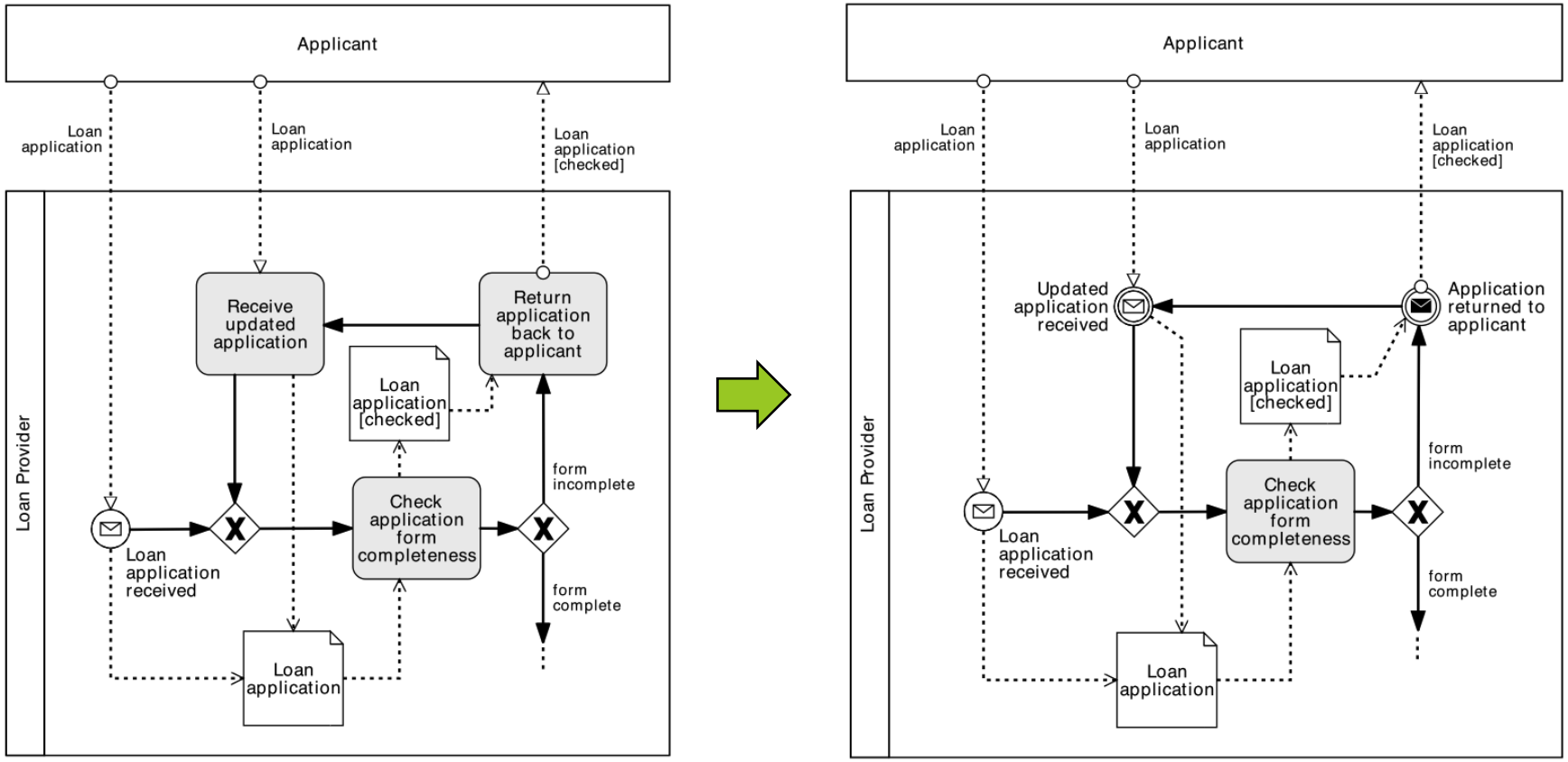
Intermediate Message Event – Indicates that an event is expected to occur during the process. The event is triggered when a message is **received** or **sent**

Comparison with sending/receiving tasks

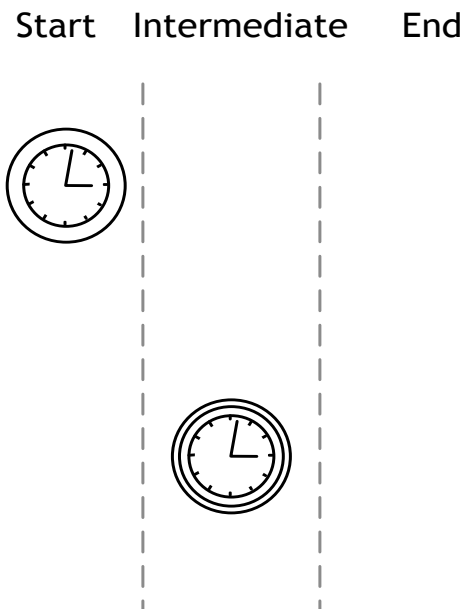


So, when to use what?

Use message events only when the corresponding activity would simply send or receive a message and do nothing else



Temporal events



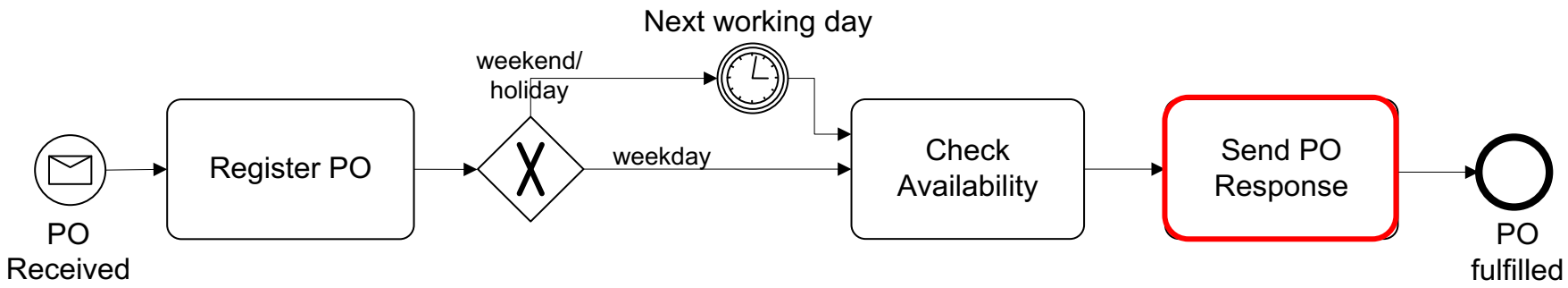
Start Timer Event – Indicates that an instance of the process is created at certain date(s)/time(s), e.g. start process at 6pm every Friday

Intermediate Timer Event – Triggered at certain date(s)/ time(s), or after a time interval has elapsed since the moment the event is “enabled” (delay)

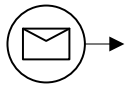
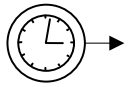
Example

PO handling

A Purchase Order (PO) handling process starts when a PO is received. The PO is first registered. If the current date is not a working day, the process waits until the following working day before proceeding. Otherwise, an availability check is performed and a PO response is sent back to the customer.



Recap: Message and Timer events

Start	Intermediate		End
Catching	**Catching**	**Throwing**	**Throwing**
 process starts upon message received	message received during the process	message sent during the process	process ends upon message sent
 process starts when time event occurs	time event occurred (to model delay)		

Data-based vs. Event-based decision

With the XOR-split gateway, a branch is chosen based on conditions that evaluate over available **data**

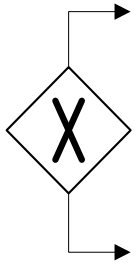
→ The choice can be made immediately after the token arrives from the incoming flow

Sometimes, the choice must be delayed until an **event** happens

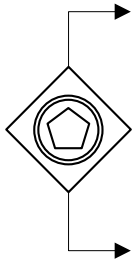
→ The choice is based on a “race” among events

Two types of XOR split:

data-driven XOR-split



event-driven XOR split



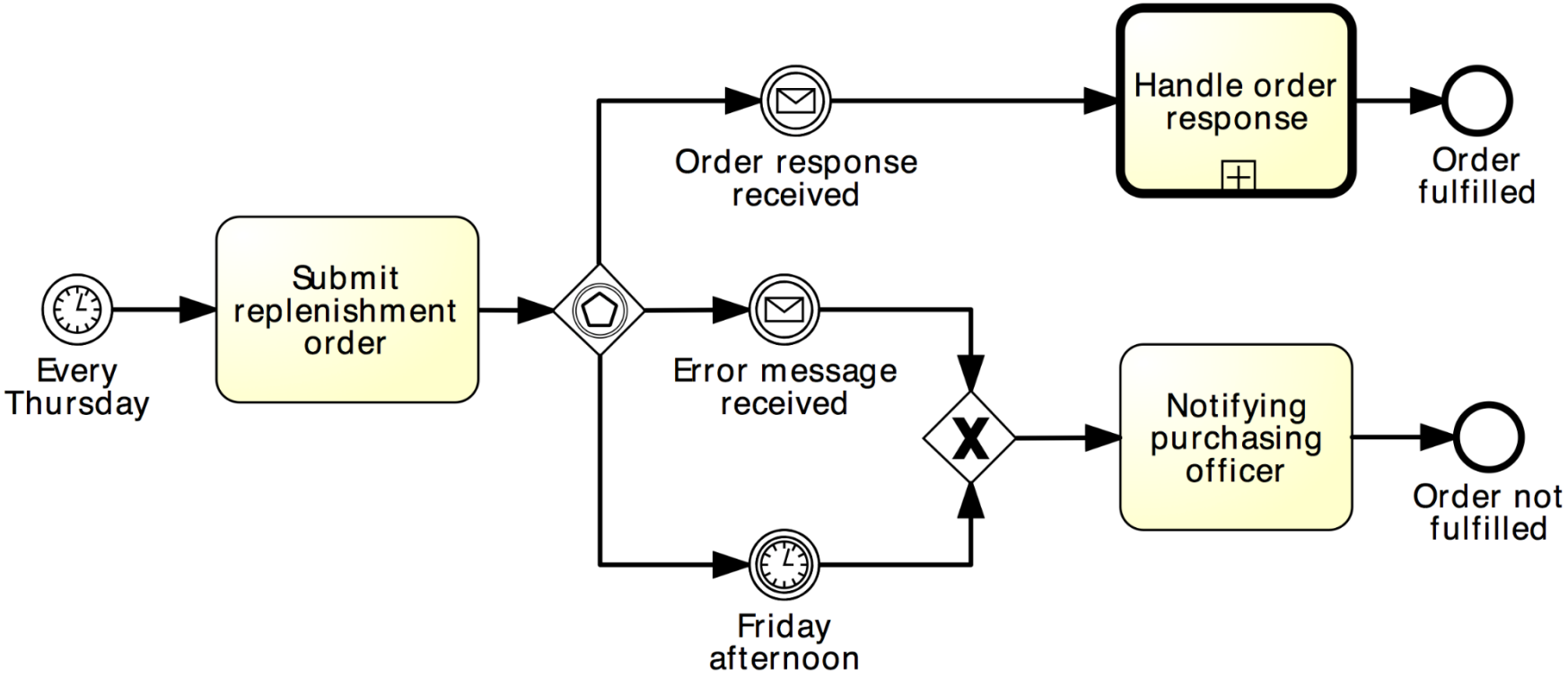
Choices outside our control…

Stock replenishment

A restaurant chain submits a purchase order (PO) to replenish its warehouses every Thursday. The restaurant chain's procurement system expects to receive either a "PO Response" or an error message. However, it may also happen that no response is received at all due to system errors or due to delays in handling the PO on the supplier's side. If no response is received by Friday afternoon or if an error message is received, a purchasing officer at the restaurant chain's headquarters should be notified. Otherwise, the PO Response is processed normally.

Solution: event-driven XOR split

Stock replenishment



EXCEPTION HANDLING

Abortion (terminate event)

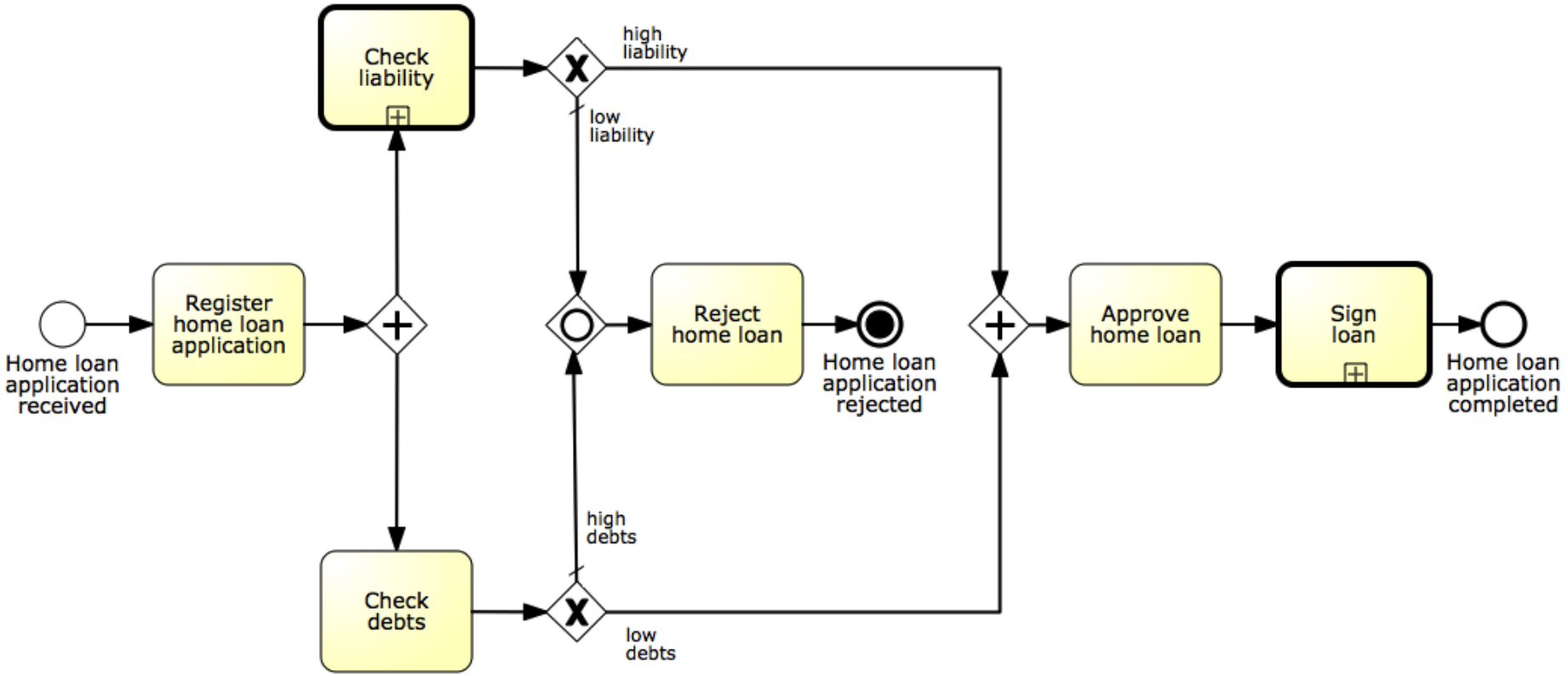
Exceptions are events that deviate a process from its “normal” course

The simplest form of exception is to notify that there is an exception (negative outcome)

This can be done via the Terminate end event: it forces the whole process to *abort* (“wipes off” all tokens left behind, if any)

Example: terminate event

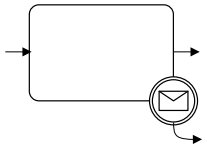
Abort the process by removing all tokens…



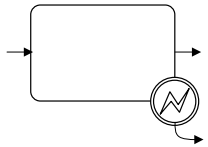
Exception handling

Handling exceptions often involves stopping a sub-process and performing a special activity

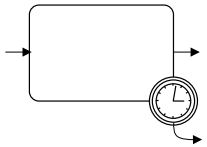
Types of exceptions for an activity (task/sub-process) in BPMN:



External: something goes wrong outside the process, and the execution of the current activity must be interrupted. Handled with the Message event



Internal: something goes wrong inside an activity, whose execution must thus be interrupted. Handled with the Error event



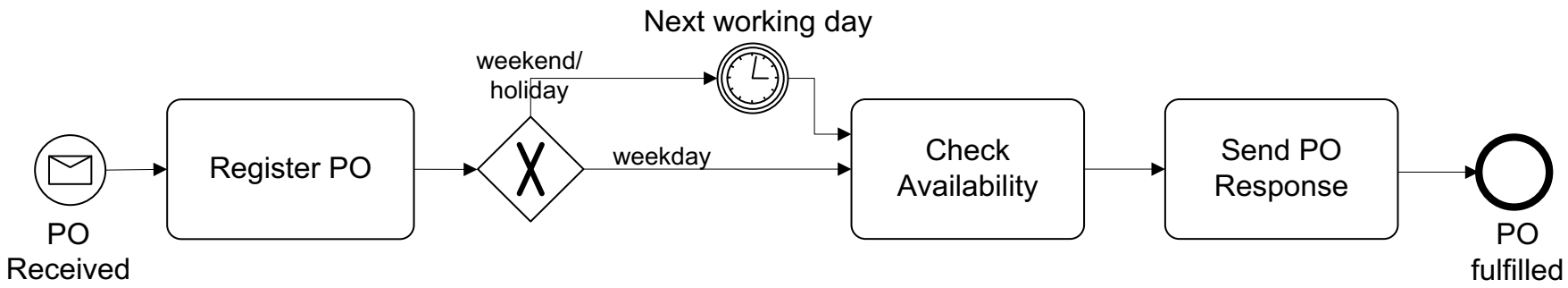
Timeout: an activity takes too long and must be interrupted. Handled with the Timer event

All these events are catching intermediate events. They stop the enclosing activity and start an exception handling routine.

Let’s extend our PO handling process

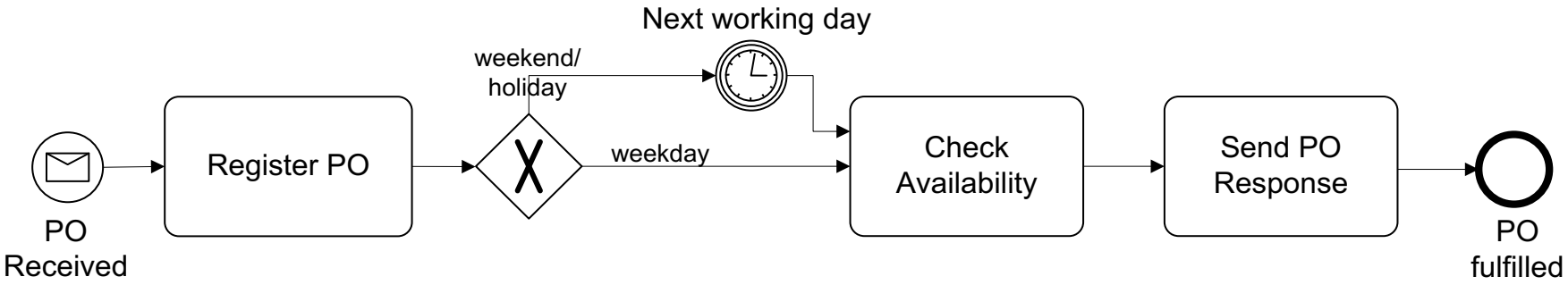
PO handling

A PO handling process starts when a PO is received. The PO is first registered. If the current date is not a working day, the process waits until the following working day before proceeding. Otherwise, an availability check is performed and a PO response is sent back to the customer.



Let’s extend our PO handling process

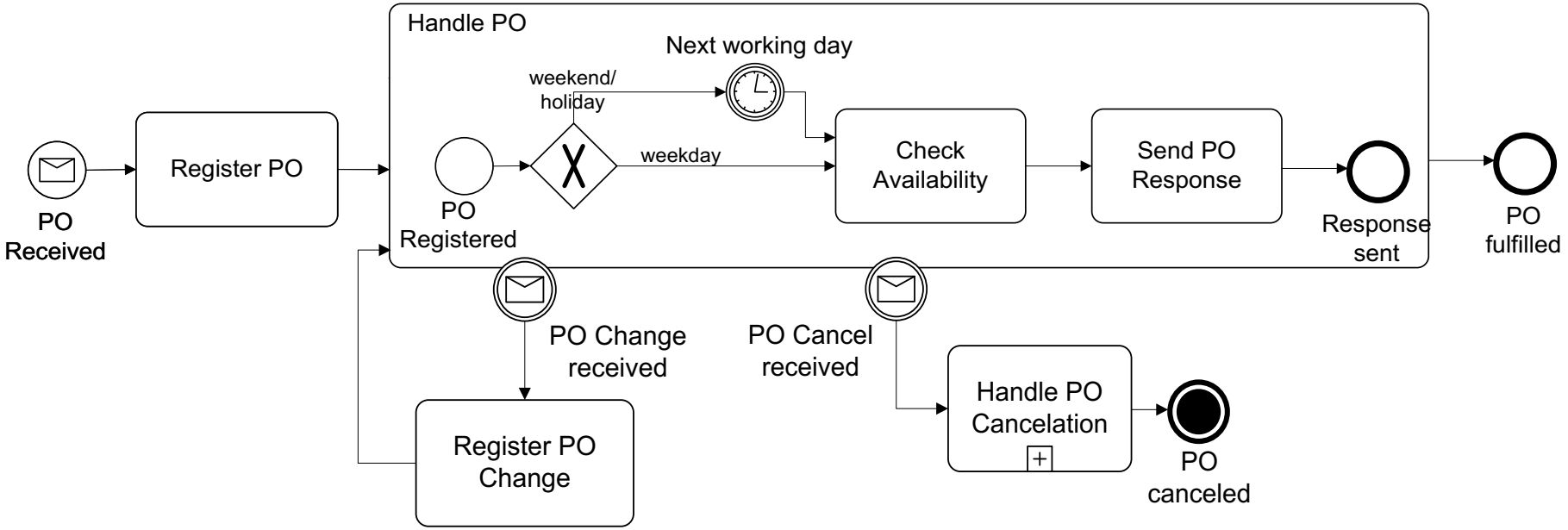
PO handling



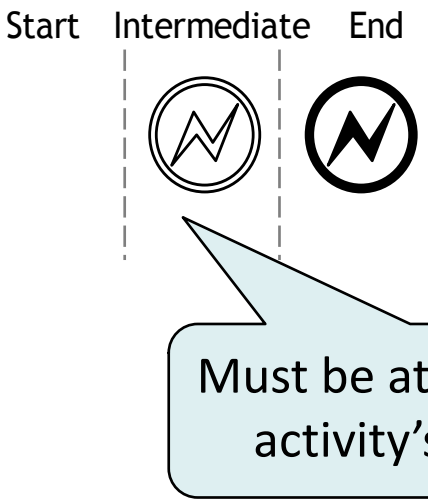
A PO change request may be received anytime after the PO is registered. This request includes a change in quantity or line items. When such a request is received, any processing related to the PO must be stopped. The PO change request is then registered. Thereafter, the process proceeds as it would do after a normal PO is registered. Further, if the customer sends a PO cancelation request after the PO registration, the PO processing must be stopped and the cancelation request must be handled.

Solution: exception handling

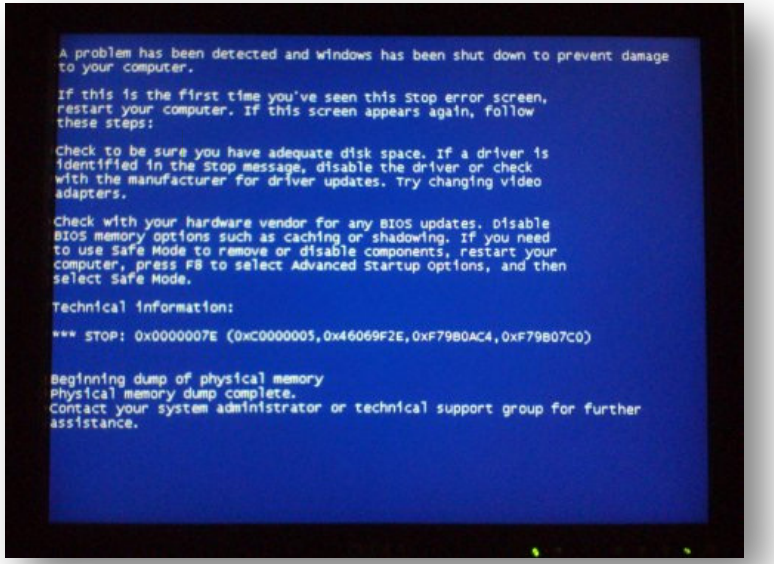
PO handling



Internal exception: error event



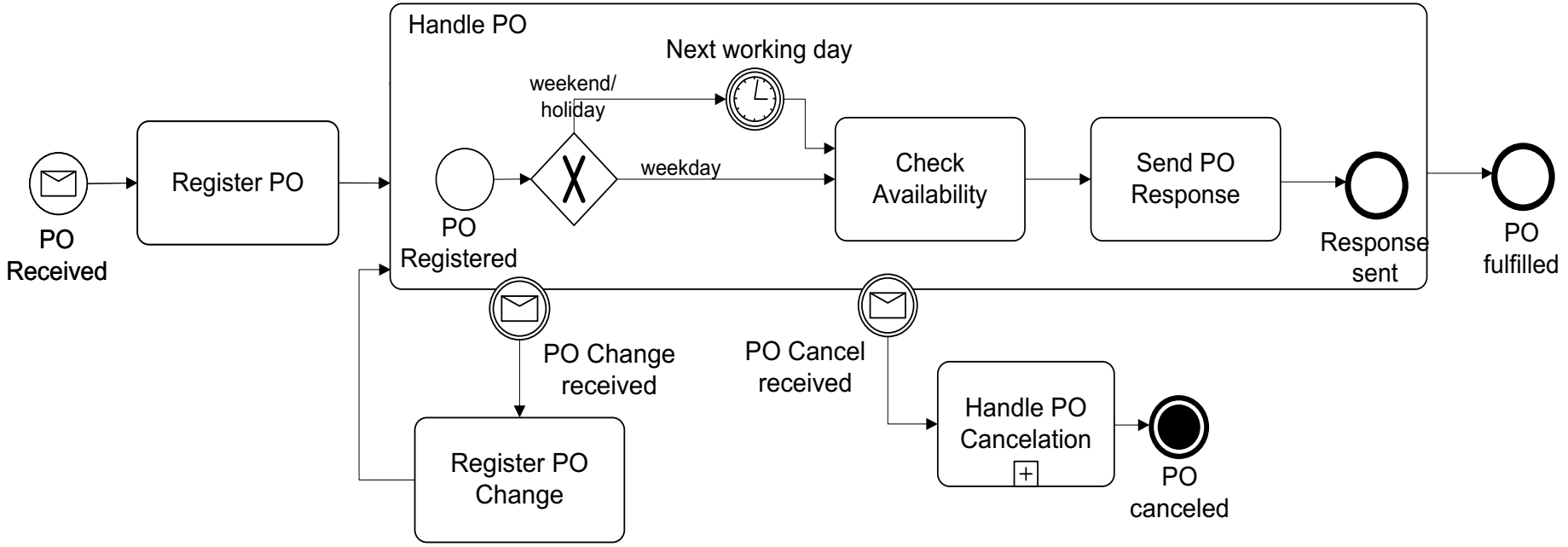
Error Event – Indicates an error: the "end" version generates an error event while the "catching intermediate" version consumes it when <u>attached</u> to the boundary of an activity



Example: internal exception

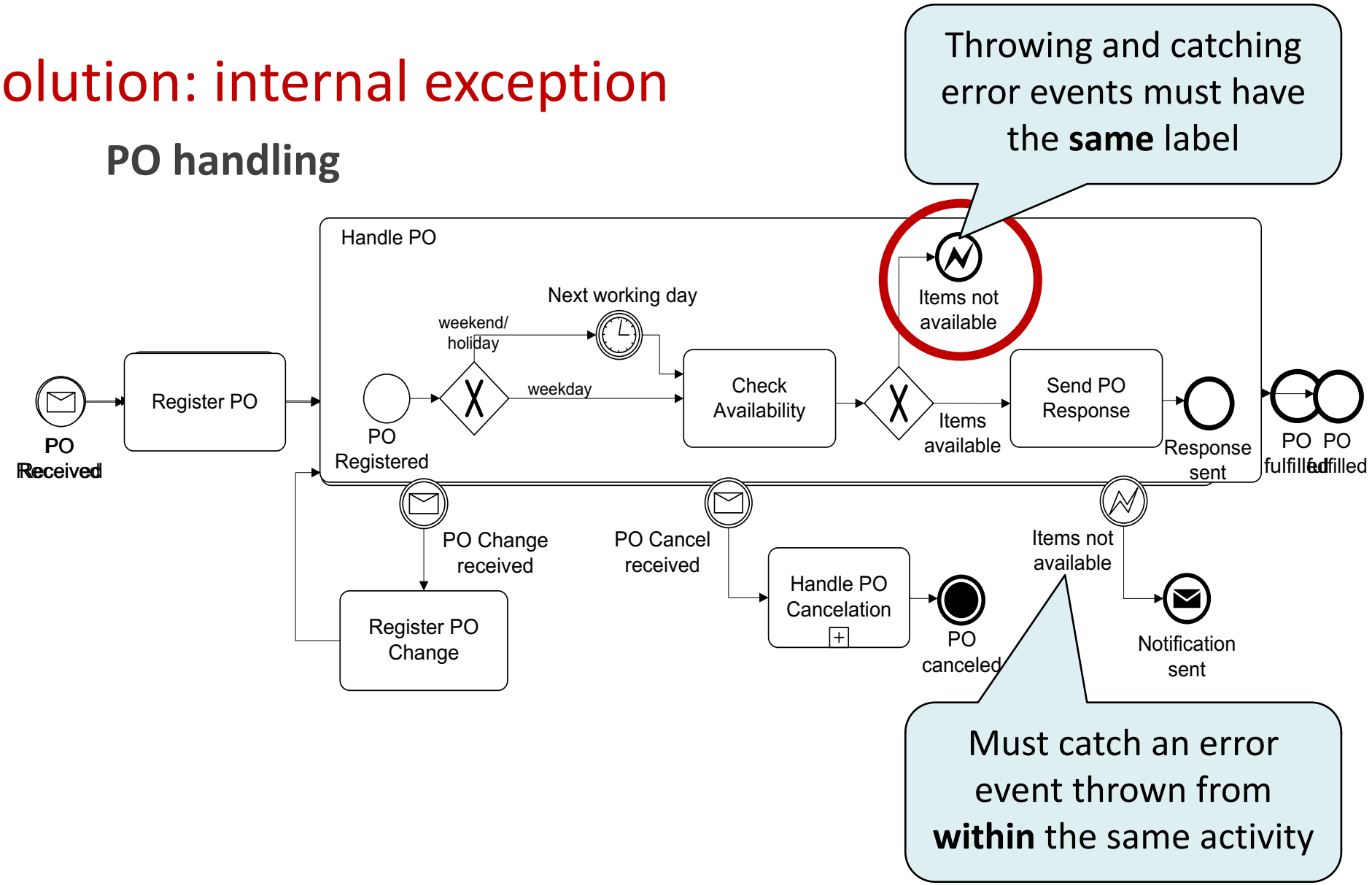
PO handling

Consider again our “PO Handling process” example with the following extension: if an item is not available, any processing related to the PO must be stopped. Thereafter, the client needs to be notified that the PO cannot be further processed.



Solution: internal exception

PO handling



Example: activity timeout

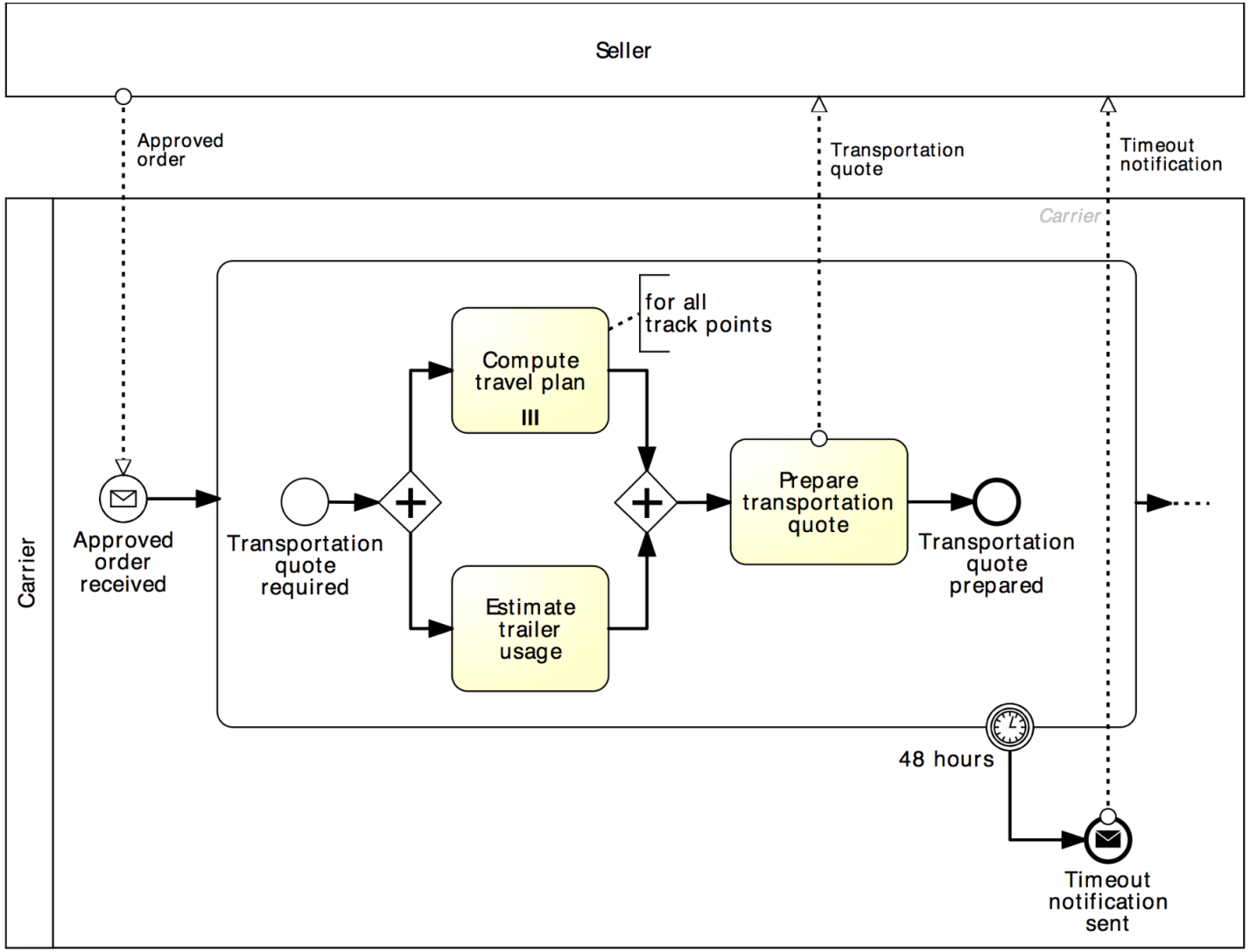
Order-to-transportation quote

Once a wholesale order has been confirmed, the supplier transmits this order to the carrier for the preparation of the transportation quote. In order to prepare the quote, the carrier needs to compute the route plan (including all track points that need to be traversed during the travel) and estimate the trailer usage.

By contract, wholesale orders have to be dispatched within four days from the receipt of the order. This implies that transportation quotes have to be prepared within 48 hours from the receipt of the order to remain within the terms of the contract.

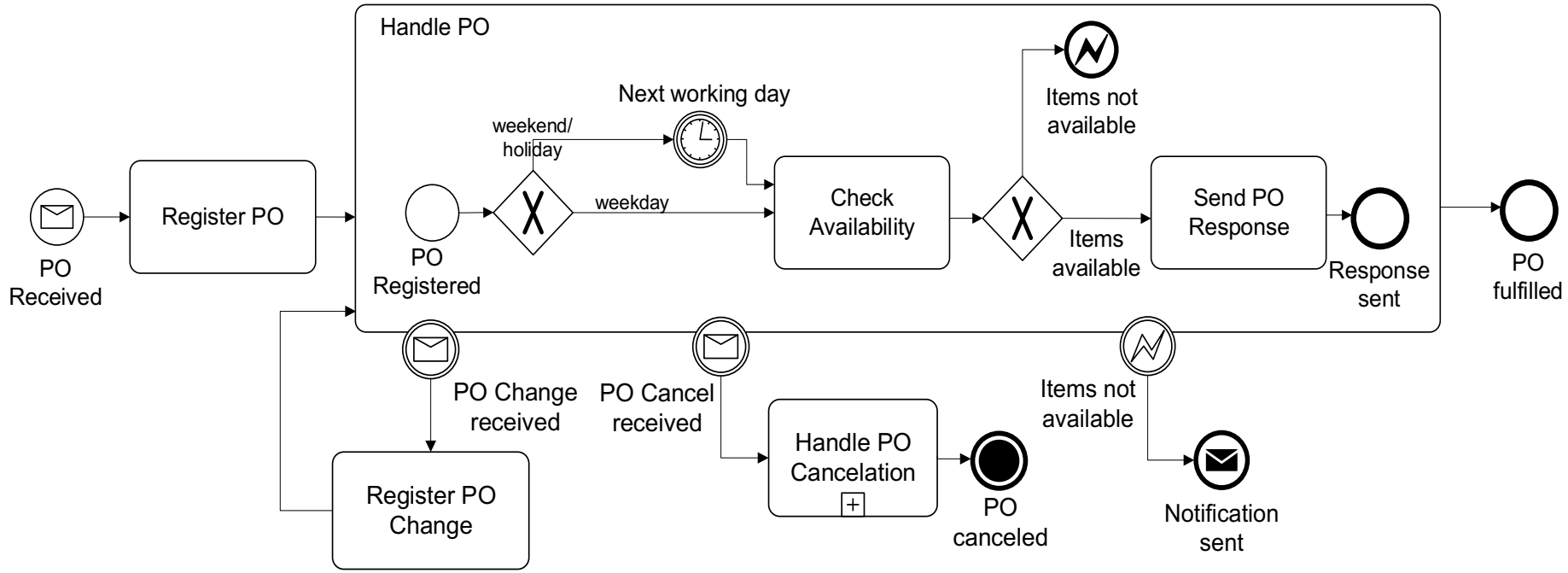
Solution: activity timeout

Order-to-transportation quote



More on the PO handling example…

PO handling

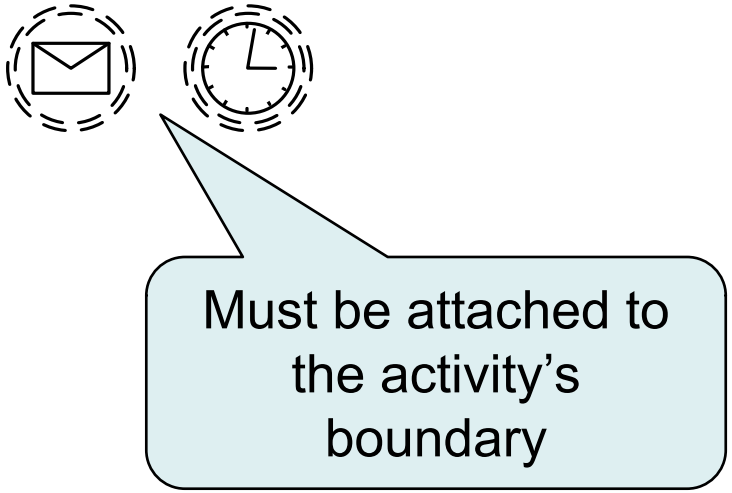


The customer may send a request for address change after the PO registration. When such a request is received, it is just registered, without further action.

Non-interrupting boundary events

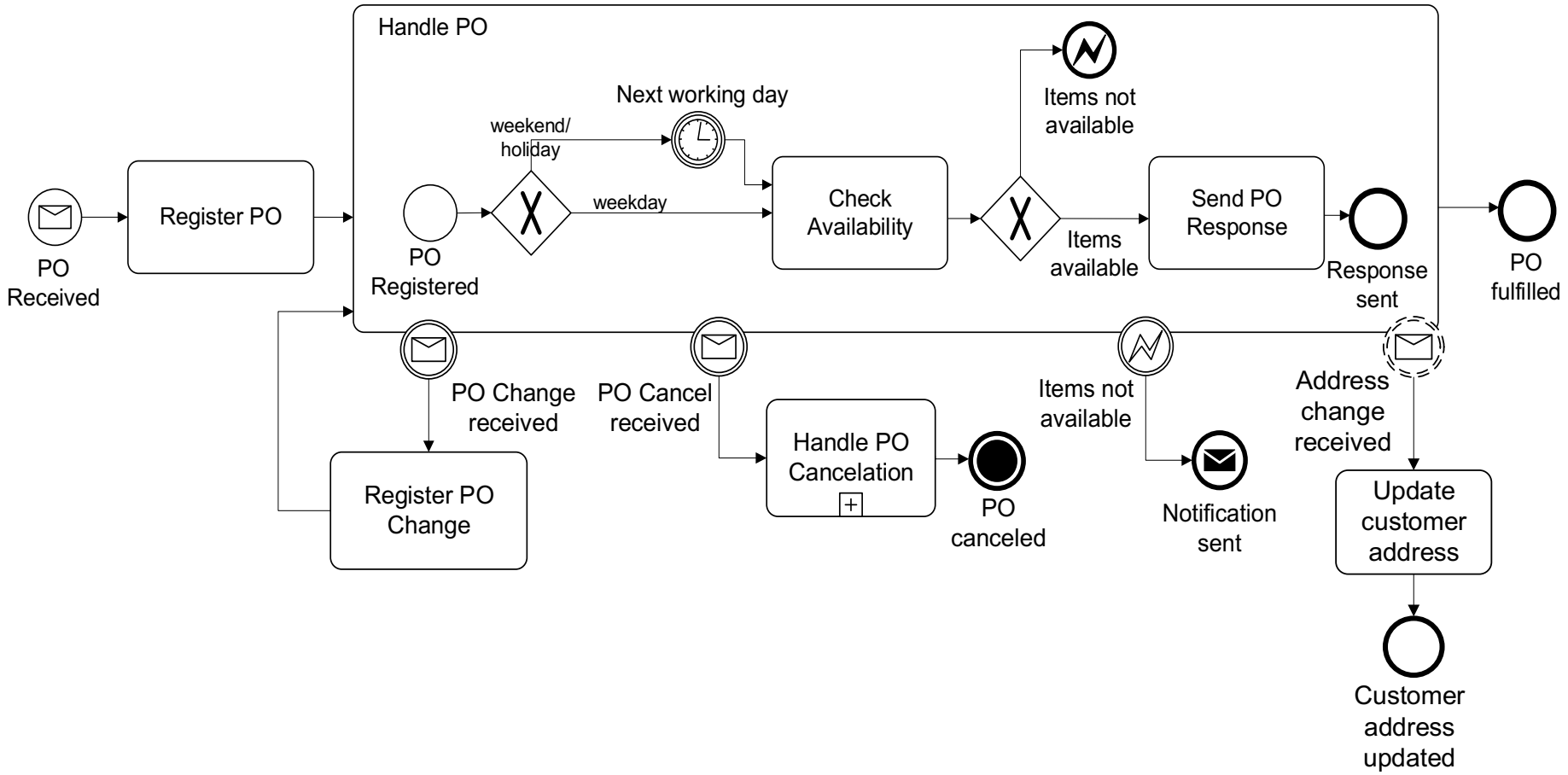
Sometimes we may need to trigger an activity **in parallel** to the normal flow, i.e. without interrupting the normal flow.

This can be achieved by using *non-interrupting* boundary events



Solution: non-interrupting boundary events

PO handling



Summary

- In this lecture we have learned about:
  - BPMN sub-processes
  - Repetition markers: loop marker and parallel multi-instance marker
  - Events: timer, message and error events
  - Event-based choice gateway
  - Boundary events: interrupting and non-interrupting
  - Error events (throw and catch)

Acknowledgements

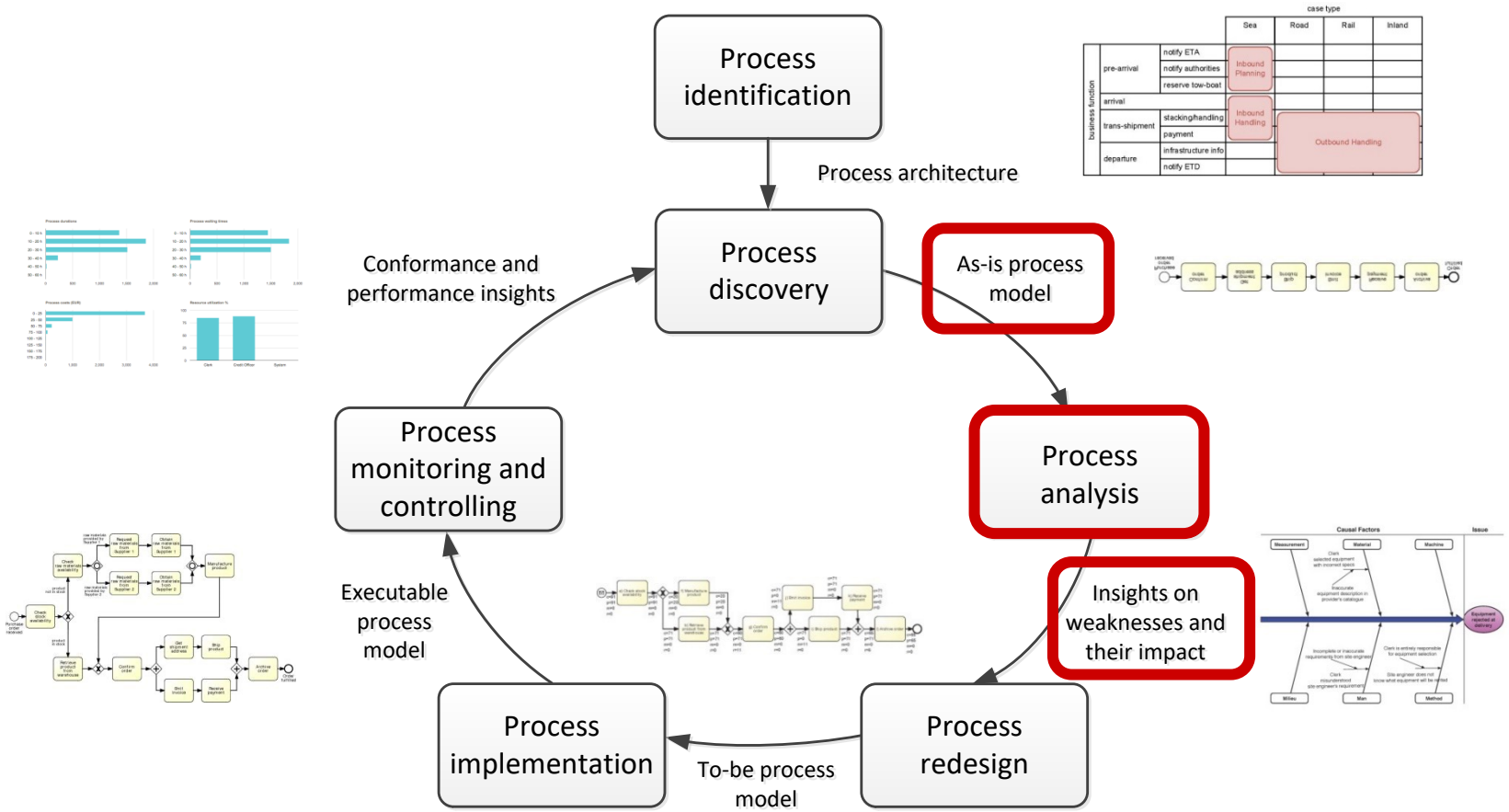
- All material comes from Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A. Reijers, authors of the “Fundamentals of Business Process Management” book.

INFO-H420
Management of Data Science and Business Workflows
Part I
4. Qualitative Process Analysis

Dimitris SACHARIDIS

2023-2024

Process Analysis



Process Analysis Techniques

Qualitative analysis

- Value-Added & Waste Analysis
- Issue Register & PICK Charts
- Root-Cause Analysis

Quantitative Analysis

VALUE-ADDED ANALYSIS

Value-added analysis

1. Decorticate the process tasks into steps
   - Steps performed before a task
   - The task itself, possibly decomposed into smaller steps
   - Steps performed after a task, in preparation for the next task

2. Classify each step
   - Value-adding (VA)
   - Business value-adding (BVA)
   - Non-value-adding (NVA)

Value-adding steps

Maximize

Produce value or satisfaction to the customer

Criteria

- Is the customer willing to pay for this step?
- Would the customer agree that this step is necessary to achieve their goals?
- If the step is removed, would the customer perceive that the end product or service is less valuable?

Examples

- Order-to-cash process: Confirm delivery date, Deliver products
- University admission process: Assess application, Notify admission outcome

Business value-adding steps

Minimize

Necessary or useful for the business to operate

Criteria

- Is this step required in order to collect revenue, to improve or grow the business?
- Would the business (potentially) suffer in the long-term if this step was removed?
- Does it reduce risk of business losses?
- Is this step required in order to comply with regulatory requirements?

Example

- Order-to-cash process: *Check* purchase order, *Check* customer's credit worthiness, Issue invoice, Collect payment, Collect customer feedback
- University admission process: *Verify* completeness of application, *Check* validity of degrees, *Check* validity of language test results

Non-value-adding steps

Remove

Everything else besides VA and BVA. Steps the customer would be unwilling to pay for

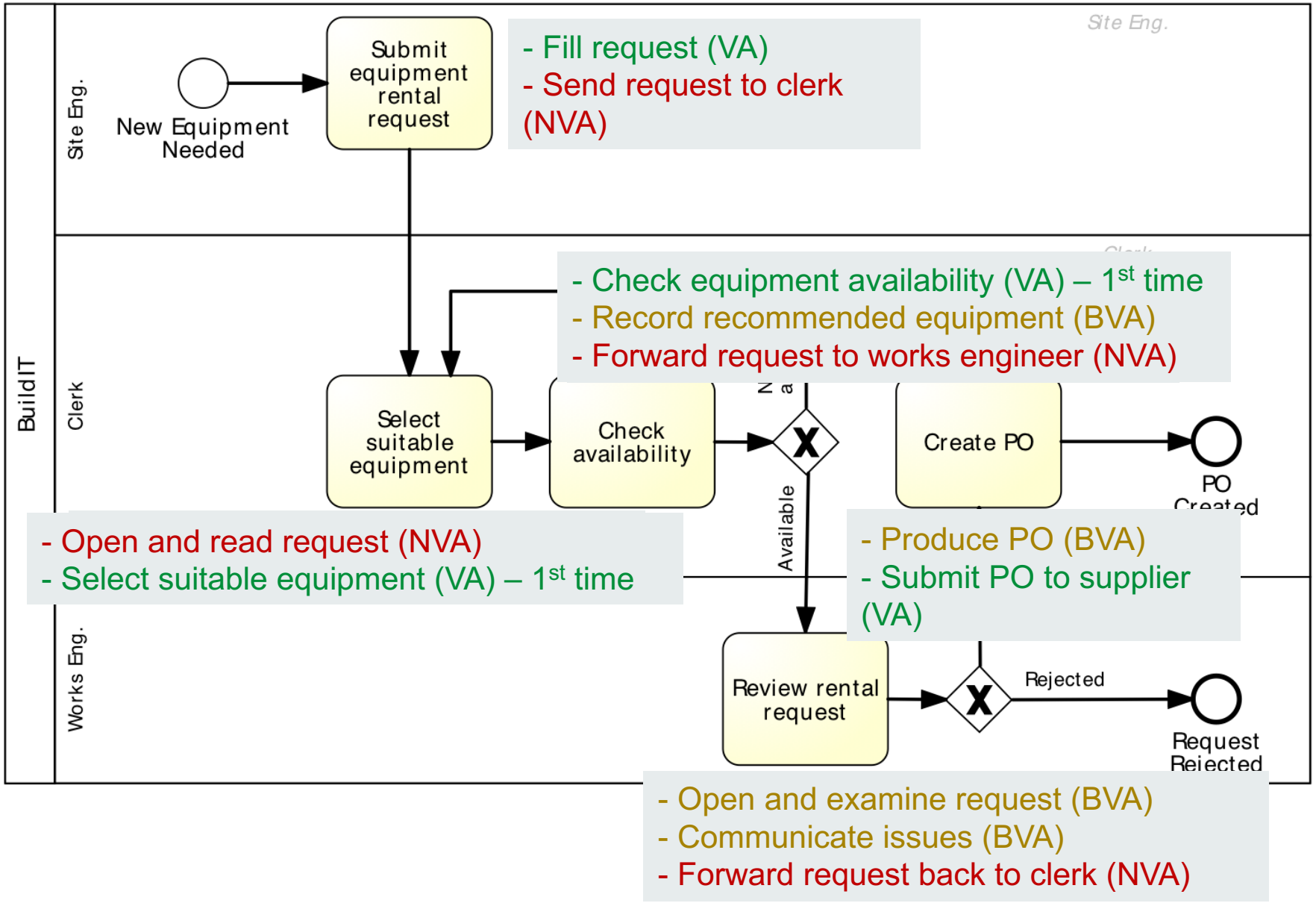
Includes

1. Handovers, context switches
2. Waiting times, delays
3. Rework or defect correction

Examples

- Order-to-cash process: *Forward* PO to warehouse, *Re-send* confirmation, *Receive* rejected products
- University admission process: *Forward* applications to committee, *Receive* admission results from committee

Extract of Equipment Rental Process

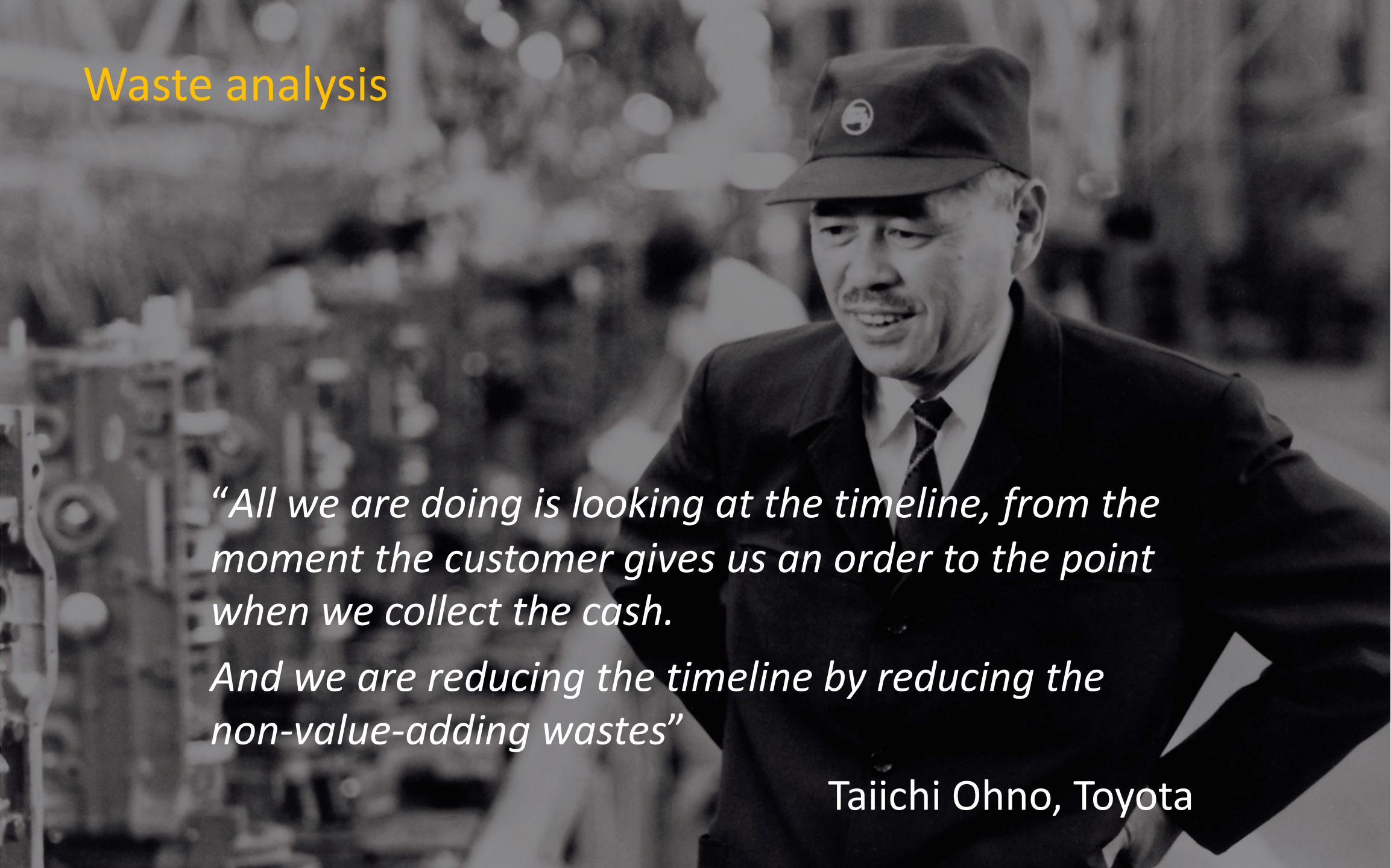


Equipment Rental Process – VA Analysis

Step	Performer	Classification
Fill request	Site engineer	VA
Send request to clerk	Site engineer	NVA
Open and read request	Clerk	NVA
Select suitable equipment	Clerk	VA
Check equipment availability	Clerk	VA
Record recommended equipment & supplier	Clerk	BVA
Forward request to works engineer	Clerk	NVA
Open and examine request	Works engineer	BVA
Communicate issues	Works engineer	BVA
Forward request back to clerk	Works engineer	NVA
Produce PO	Clerk	BVA
Send PO to supplier	Clerk	VA

WASTE ANALYSIS

Waste analysis



"All we are doing is looking at the timeline, from the moment the customer gives us an order to the point when we collect the cash.

And we are reducing the timeline by reducing the non-value-adding wastes"

Taiichi Ohno, Toyota

Seven sources of waste

Move

- Transportation
- Motion

Hold

- Inventory
- Waiting

Over-do

- Defects
- Over-Processing
- Over-Production

Move

Transportation

Send or receive materials or documents (incl. electronic) taken as input or output by the process activities

Example

University admission process: to apply for admission at a university, students fill in an online form. When a student submits the online form, a PDF document is generated. The student is requested to download it, sign it, and send it by post together with the required documents:

1. Certified copies of degree and academic transcripts
2. Results of language test
3. CV

When the documents arrive at the admissions office, an officer checks their completeness. If a document is missing, an e-mail is sent to the student. The student has to send the missing documents by e-mail or post depending on document type.

Motion

- Motion of resources internally within the process
- Common in manufacturing processes, less common in service processes

Examples

- Application-to-approval process: a process worker moves around the organization to collect signatures

Hold

Inventory

- Materials inventory
- Work-in-process (WIP)

Example

- University admission process: About 3000 applications are handled concurrently
- Vehicle inspection process: when a vehicle does not pass the first inspection, it is sent back for adjustments and left in a pending status. At a given point in time, about 100 vehicles are in the “pending” status across all inspection stations

Waiting

- Task waiting for materials or input data
- Task waiting for a resource
- Resource waiting for work (resource idleness)

Examples

- Application-to-Approval process: Request waiting for approver
- University admission process: Incomplete application waiting for additional documents; batch of applications waiting for committee to meet
- Vehicle inspection process: A technician at a base of the inspection station waiting for the next vehicle

Over-do

Defects

- Correcting or compensating for a defect or error
- Rework loops

Examples

- Travel approval process: Request sent back to requestor for revision
- University admission process: Application sent back to applicant for modification; request needs to be re-assessed later due to incomplete information
- Vehicle inspection process: A vehicle needs to come back to a station due to an omission

Over-processing

- Tasks performed unnecessarily given the outcome of the process
- Unnecessary perfectionism

Examples

- Travel approval process: 10% of approvals are trivially rejected at the end of the process due to lack of budget
- University admission process: Officers spend time verifying the authenticity of degrees, transcripts and language test results. In 1% of cases, these verifications uncover issues. Verified applications are sent to the admissions committee. The admission committee accepts 20% of the applications it receives
- Vehicle inspection process: technicians take time to measure vehicle emissions with higher accuracy than required, only to find that the vehicle clearly does not fulfill the required emission levels

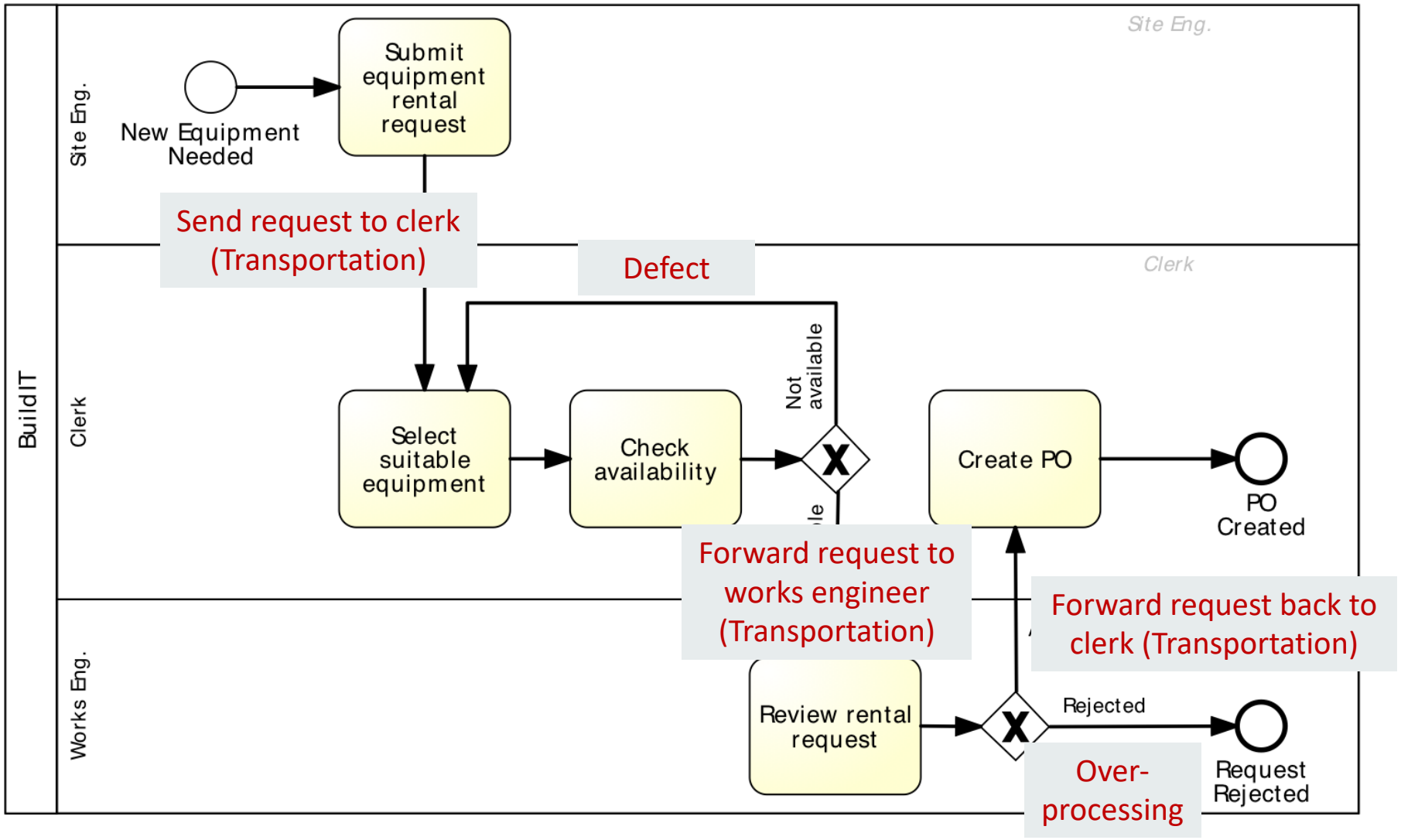
Over-production

- Unnecessary process instances are performed, producing outcomes that do not add value upon completion

Examples

- Quote-to-cash process: In 50% of cases, issued quotes do not lead to an order
- Travel approval process: In 5% of cases, travel requests are approved but the travel is cancelled
- University admission process: About 3000 applications are submitted, but only 600 are considered eligible after assessment

Equipment rental process: wastes



Equipment rental process: wastes

Transportation

- Site engineer sends request to clerk
- Clerk forwards to works engineer
- Works engineer sends back to clerk

Inventory

- Equipment kept longer than needed

Waiting

- Waiting for availability of works engineer to approve

Equipment rental process: wastes

Defect

- Selected equipment not available, alternative equipment sought
- Incorrect equipment delivered and returned to supplier

Over-processing

- Clerk finds available equipment and rental request is rejected by works engineer
- Rental requests being approved and then canceled by site engineer because no longer needed

Over-production

- Equipment being rented and not used at all by site engineer
- Equipment returned by site engineer because is incorrect

ISSUE REGISTER

Issue register

Purpose: to maintain, organize and prioritize perceived weaknesses of the process (issues)

Sources of issues:

- Input to the BPM project
- Collected during process discovery (e.g. during modelling workshops)
- Collected via *stakeholder analysis*
  - Customers
  - Process participants (workers)
  - Process owner / managers
  - Subcontractors, business partners

Issue register structure

Can take the form of a table with:

- Issue identifier
- Short name
- Description
- Data and Hypotheses
- Impact: Qualitative and Quantitative
- Possible improvement actions

Larger process improvement projects may require *issue trackers*

Issue example

Issue name

- Equipment kept longer than needed

Description

- Site engineers keep rented equipment longer than needed by asking for deadline extensions to the supplier

Data and hypotheses

- 3000 pieces of equipment rented p.a.
  In 10% of cases, equipment is kept two days more than needed
  Average rental cost is 100 per day

Quantitative impact

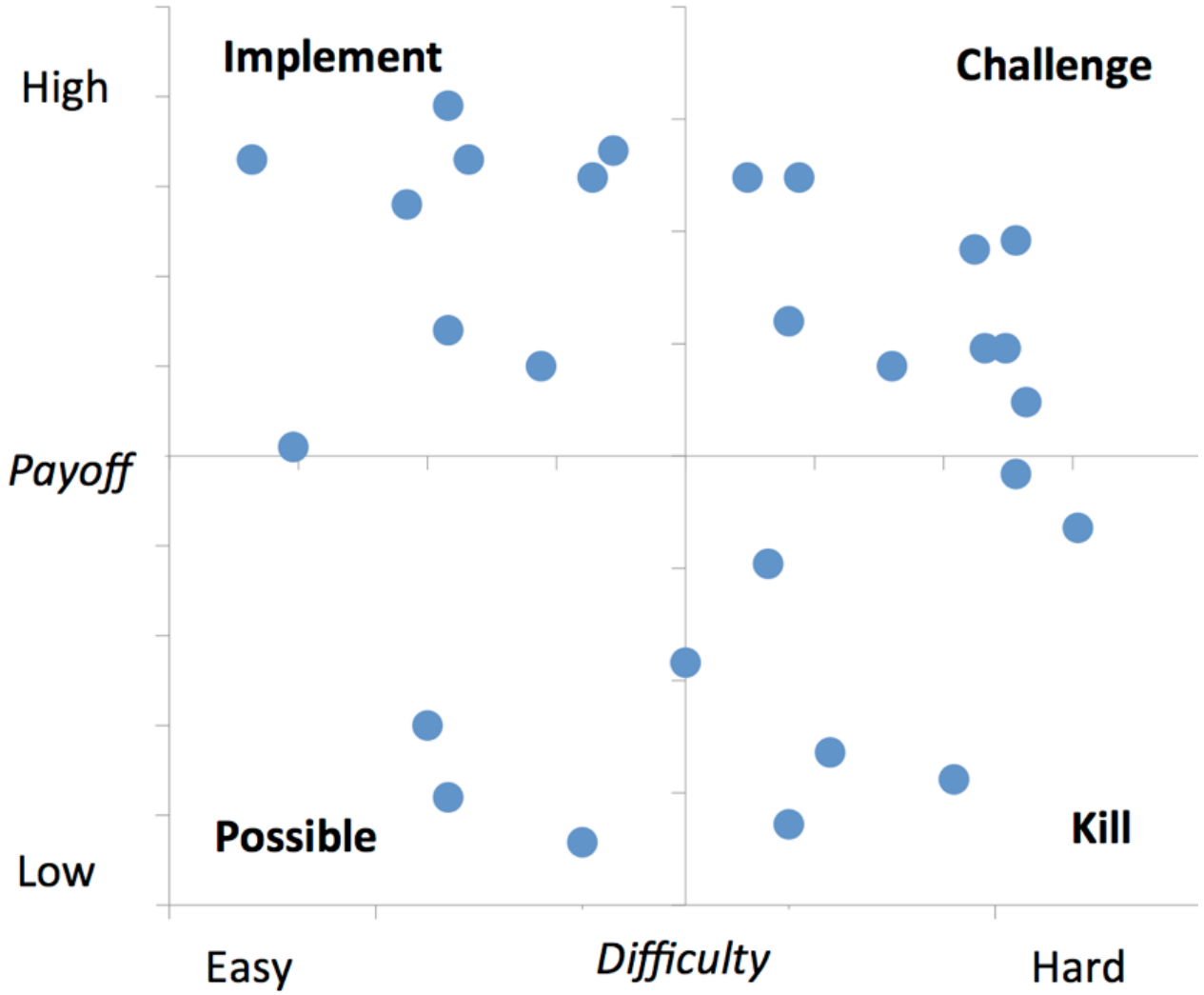
- $0.1 \times 3000 \times 2 \times 100 = 60{,}000$ p.a

Issue Register Example

Equipment rental process

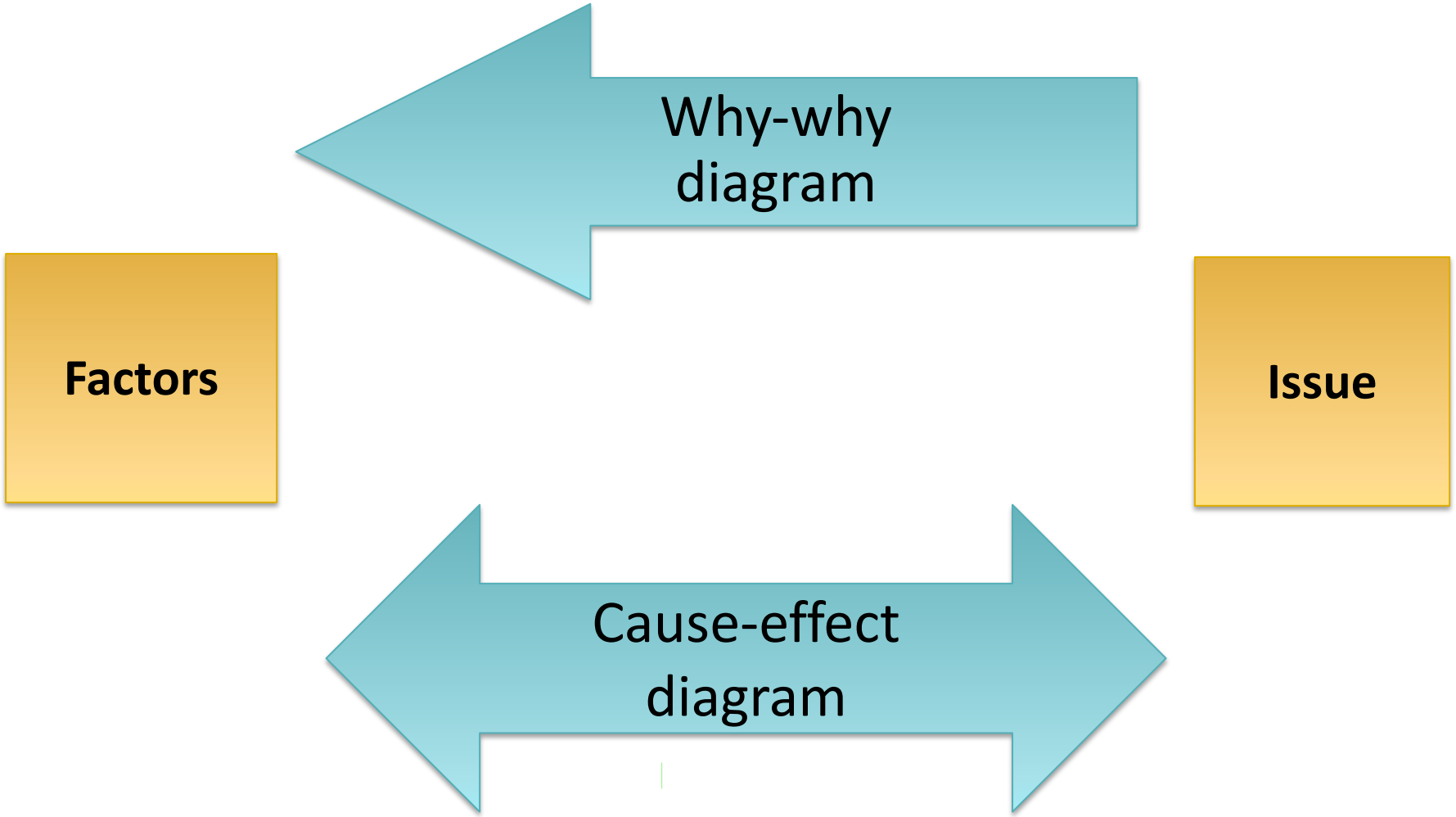
Name	Explanation	Data / Hypotheses	Qualitative Impact	Quantitative Impact
Equipment kept longer than needed	Site engineers keep equipment longer than needed via deadline extensions	3000 pieces of equipment rented p.a. In 10% of cases, equipment kept two days longer than needed. Rental cost is 100 per day		0.1 × 3000 × 2 × EUR 100 = EUR 60000 p.a.
Wrong equipment delivered	Site engineers reject delivered equipment due to non-conformance to their specifications	3000 pieces of equipment rented p.a. 5% of them are rejected due to an internal mistake For each equipment rejected due to an internal mistake, BuildIT is billed EUR 100.	Disrupted schedules. Employees stress and frustration	3000 × 0.05 × EUR 100 = EUR 15000 p.a.
Late payment fees	Late payment fees incurred because invoices are not paid by their due date	3000 pieces of equipment rented p.a. Average rental time is 4 days. Rental cost is EUR 100 per day. Each rental leads to one invoice. About 10% of invoices are paid late. Penalty for late payment is 2%.	Poor reputation with suppliers	0.1 × 3000 × 4 × EUR 100 × 0.02 = EUR 2400 p.a.

Issue Prioritization: PICK Chart



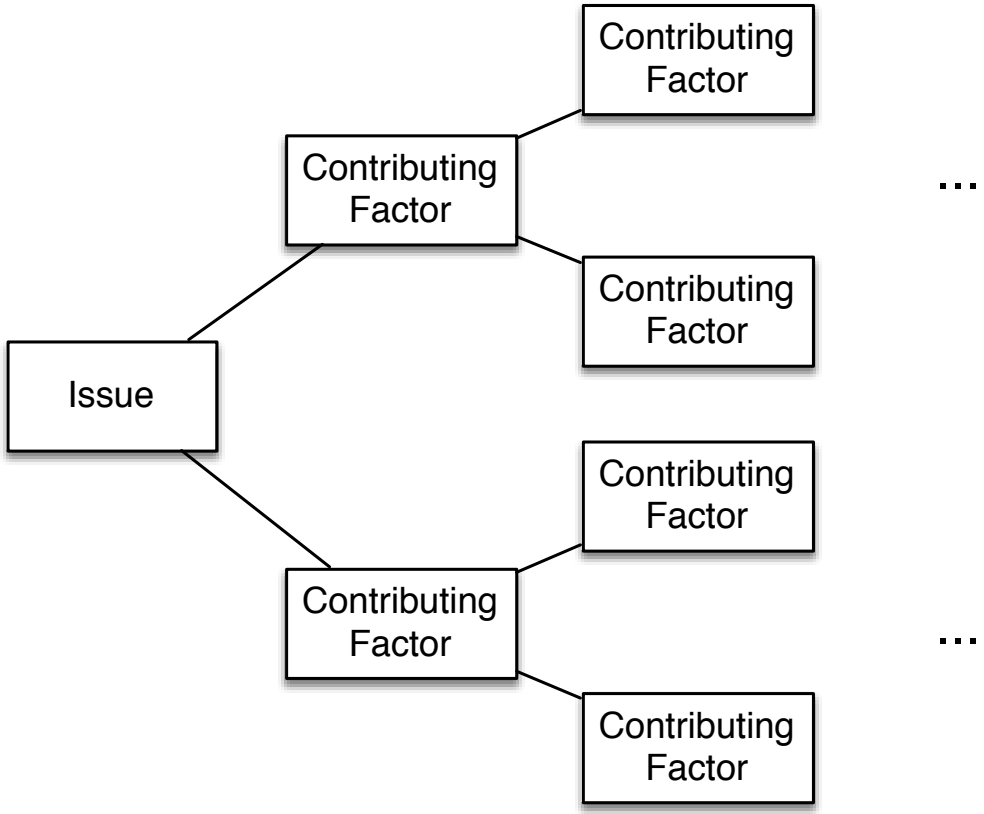
ROOT CAUSE ANALYSIS

Root-cause analysis

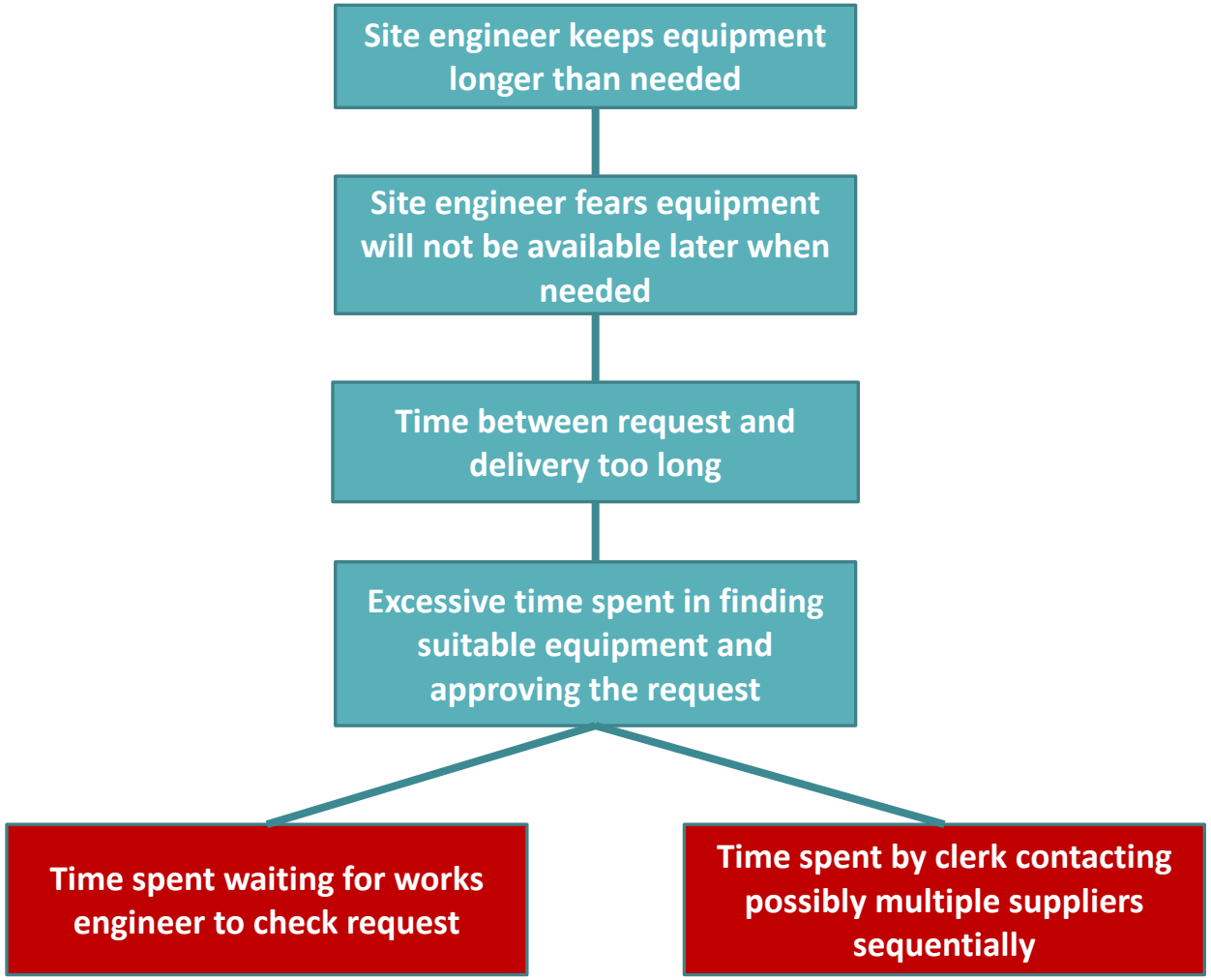


Why-why diagram

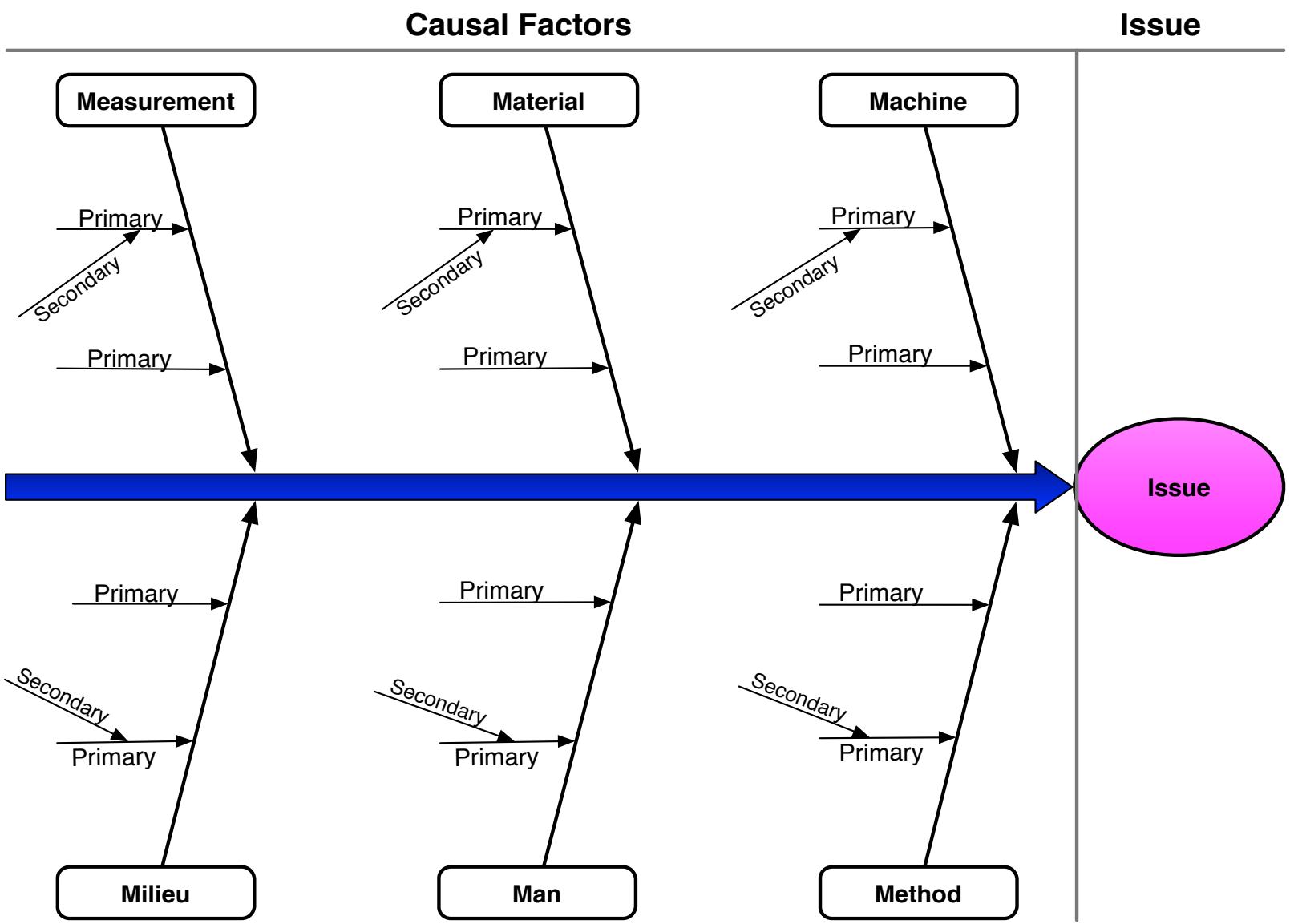
Five levels of nesting - “Five Why’s”



Why-why diagram example



Cause-effect (Fishbone) diagram



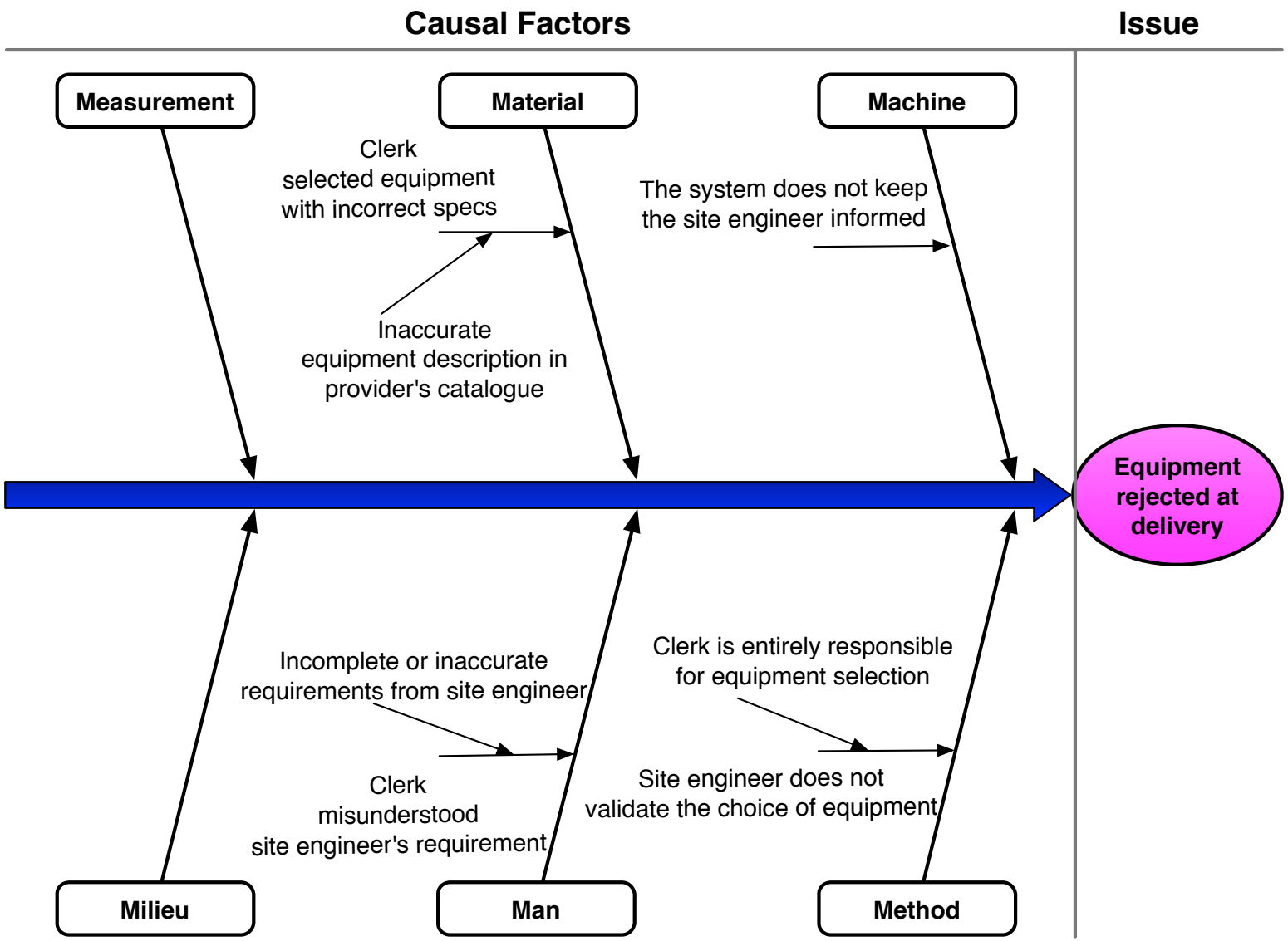
Categories of causes: Six Ms

1. **Machine:** factors stemming from technology used
   - Lack of suitable functionality in the supporting software applications
   - Poor User Interface (UI) design
   - Lack of integration between systems
2. **Method:** factors stemming from the way the process is designed, understood or performed
   - Unclear assignments of responsibilities
   - Unclear instructions
   - Insufficient training
   - Lack of timely communication
3. **Material:** factors stemming from input materials or data
   - Missing, incorrect or outdated data

Categories of causes: Six Ms

4. **Man:** factors stemming from wrong assessments or incorrect performance of steps attributable to:
   - Lack of training and clear instructions
   - Lack of motivation
   - Too high demands towards process workers
5. **Measurement:** factors stemming from reliance on:
   - Inaccurate estimations
   - Miscalculations
6. **Milieu:** factors outside the scope of the process
   - Delays caused because of unresponsive external actors
   - Sudden increases of workload due to special circumstances

Cause-effect diagram example



Summary

1. Segregate value-adding, business value-adding and non-value-adding steps
2. Identify waste
3. Collect and systematically organize issues, assess their impact
4. Analyze root causes of issues

Acknowledgements

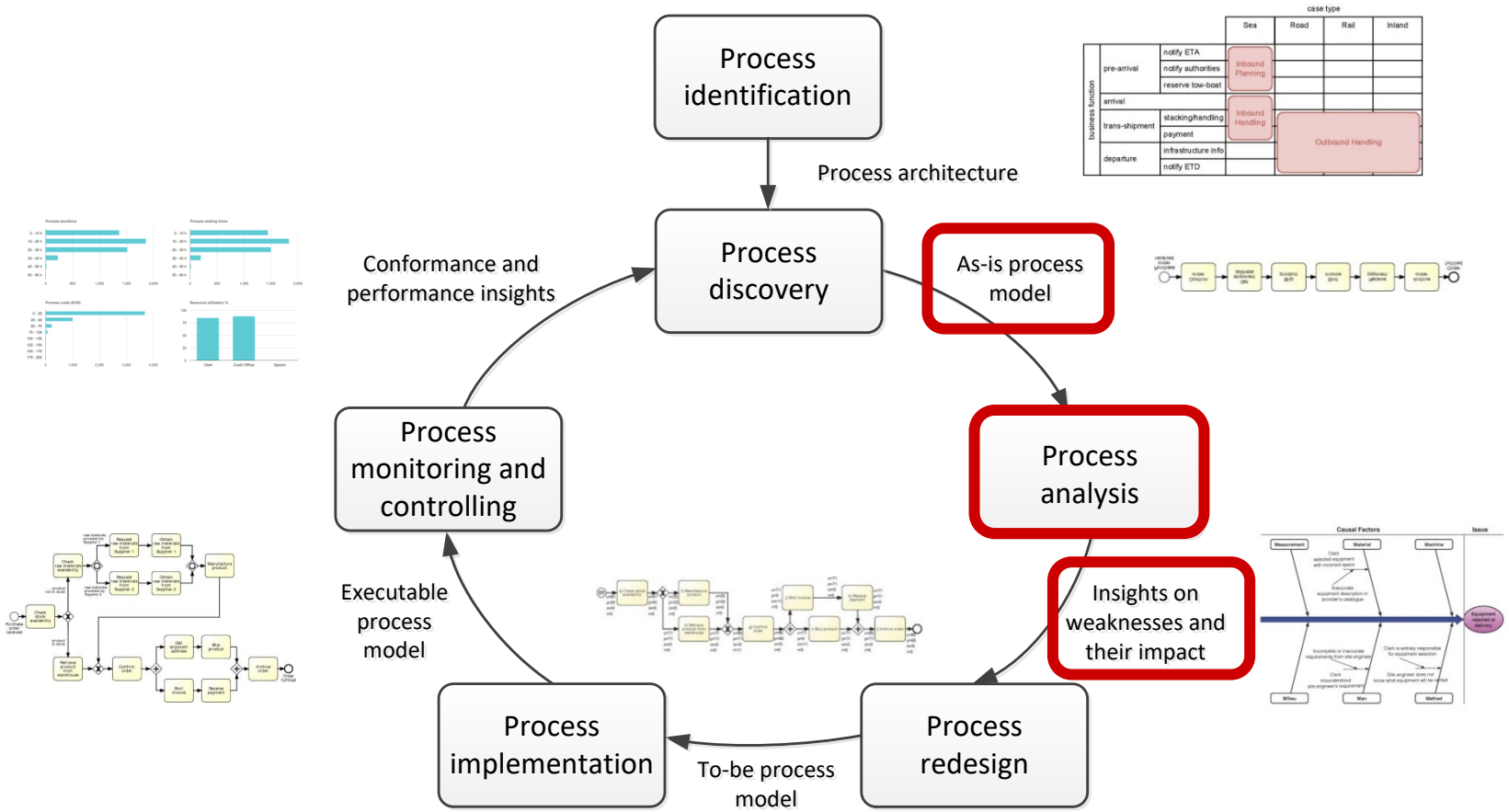
- All material comes from Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A. Reijers, authors of the “Fundamentals of Business Process Management” book.

INFO-H420
Management of Data Science and Business Workflows
Part I
5. Quantitative Process Analysis

Dimitris SACHARIDIS

2023-2024

BPM Lifecycle



Process Analysis Techniques

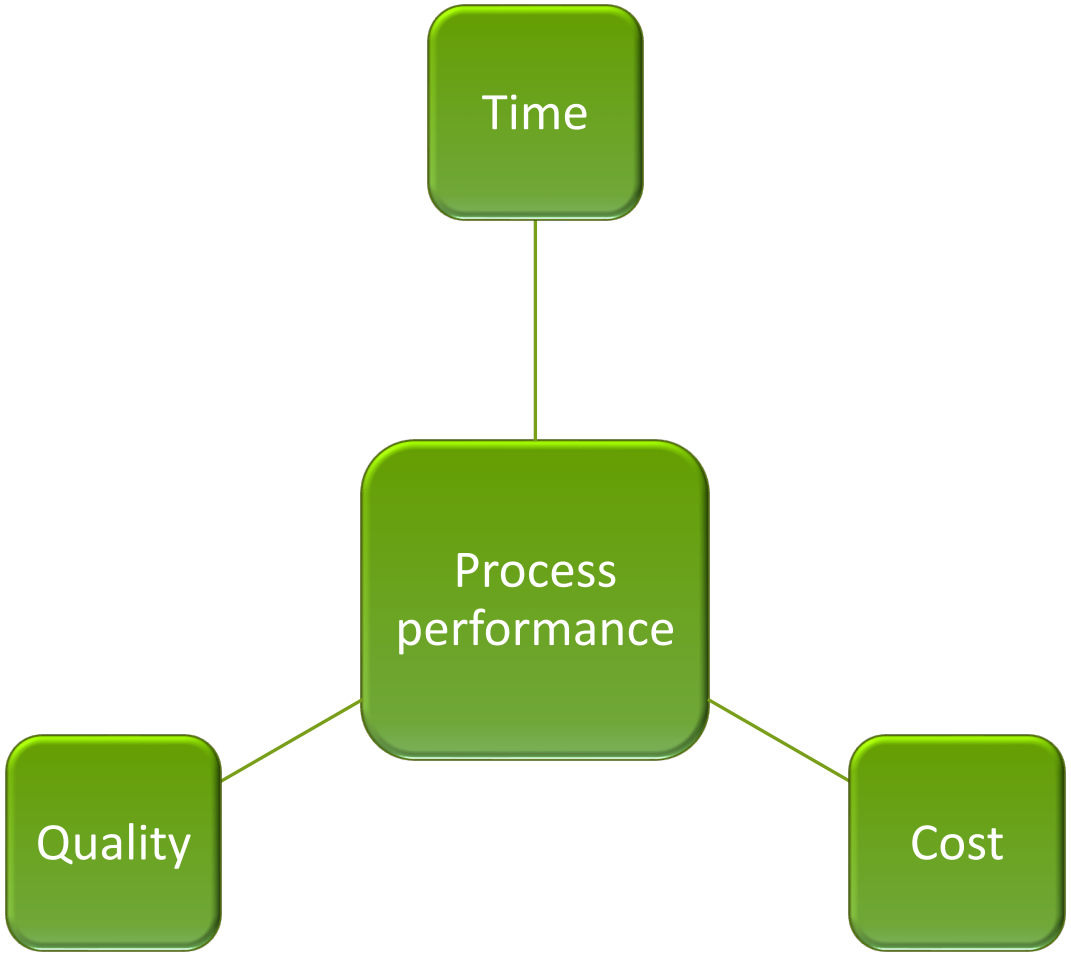
Qualitative analysis

- Value-Added & Waste Analysis
- Root-Cause Analysis
- Pareto Analysis
- Issue Register

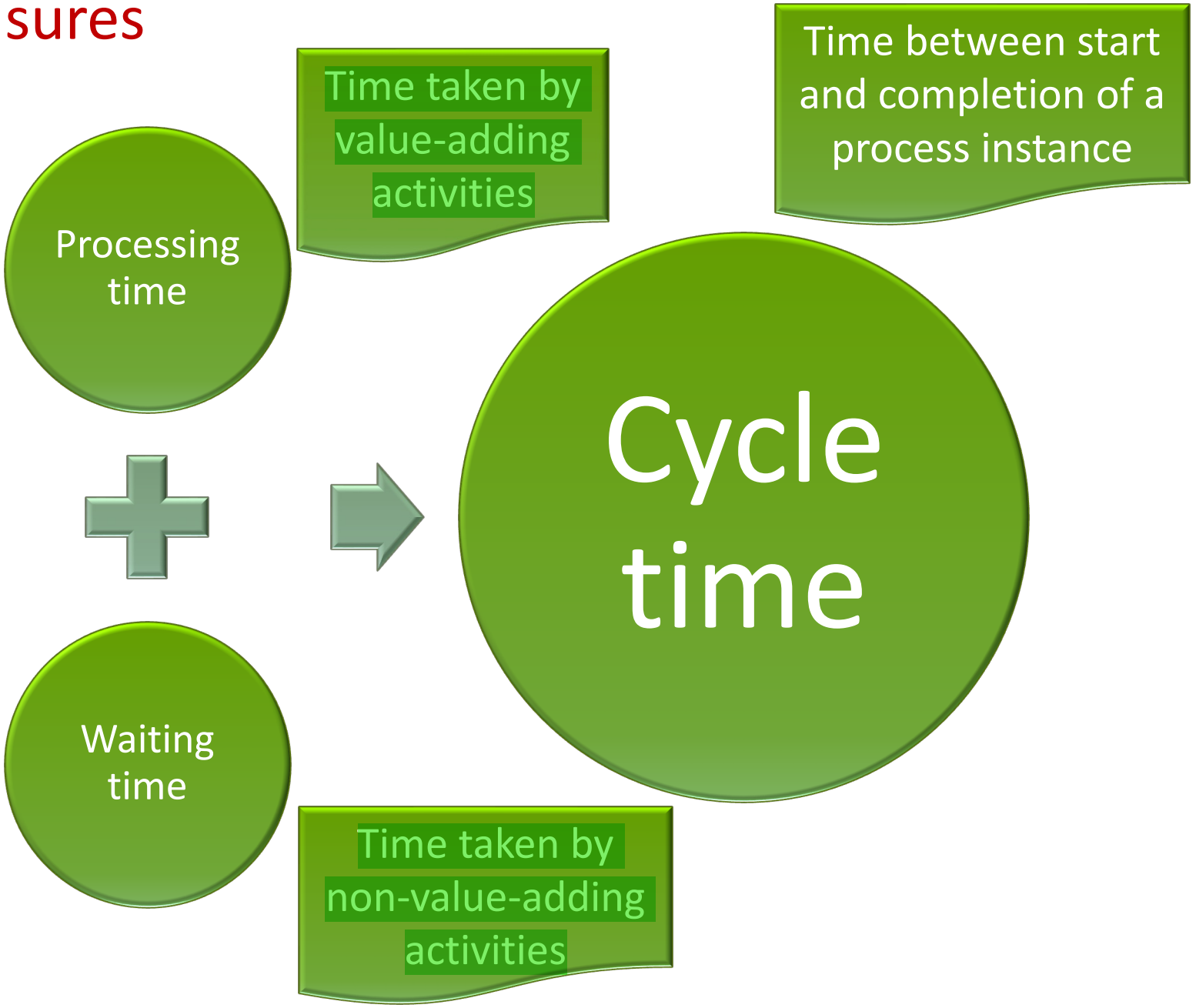
Quantitative Analysis

- Flow analysis
- Queuing analysis
- Simulation

Process performance

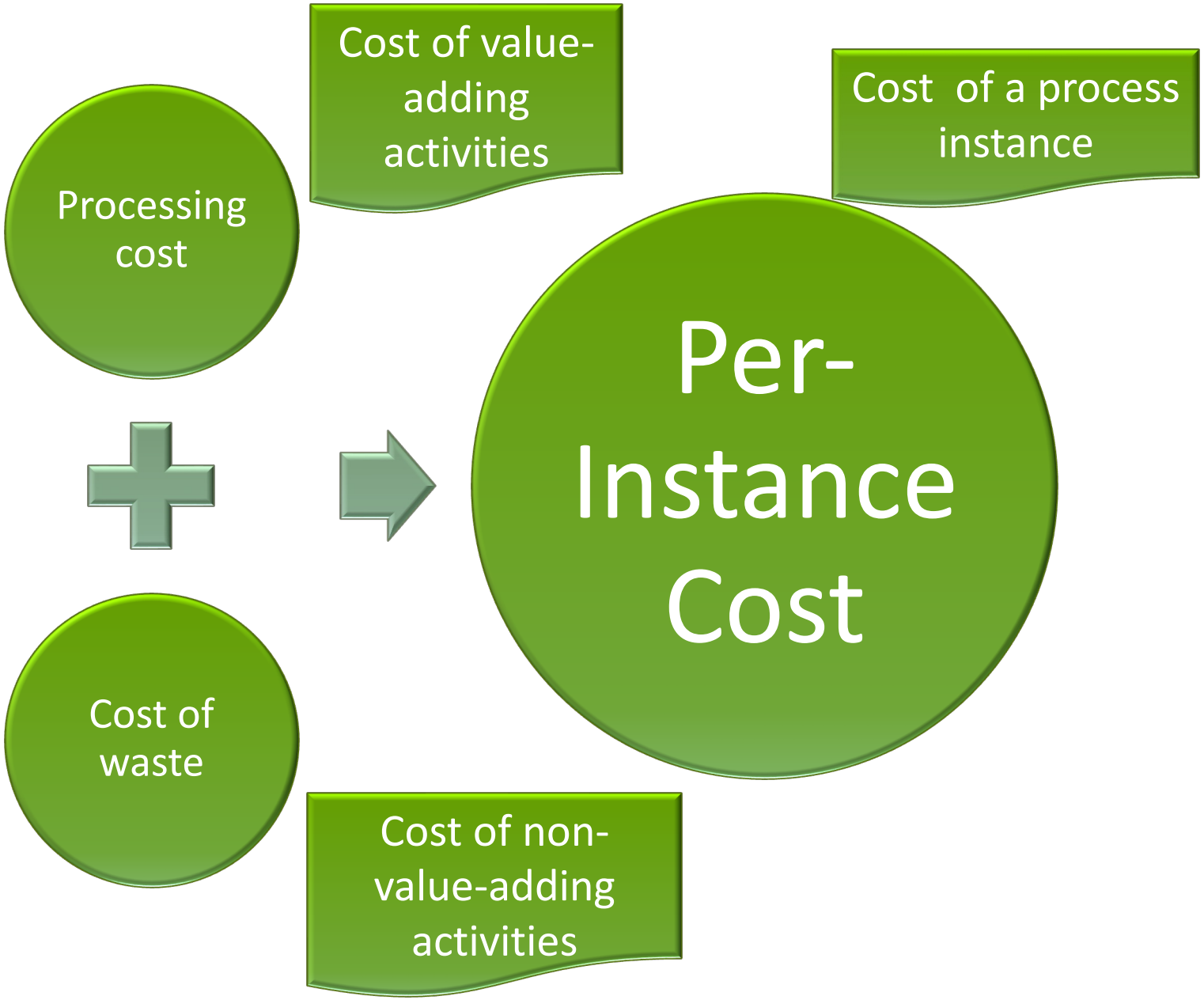


Time measures



Cycle time efficiency

Cost measures



Typical components of cost

Material cost

- Cost of tangible or intangible resources used per process instance

Resource cost

- Cost of person-hours employed per process instance

Resource utilization

Resource utilization = 60%
➔ on average resources are idle 40% of their allocated time

Resource utilization vs. waiting time

Typically, when resource utilization > 90%
➔ Waiting time increases steeply

Quality

Product quality

- Defect rate

Delivery quality

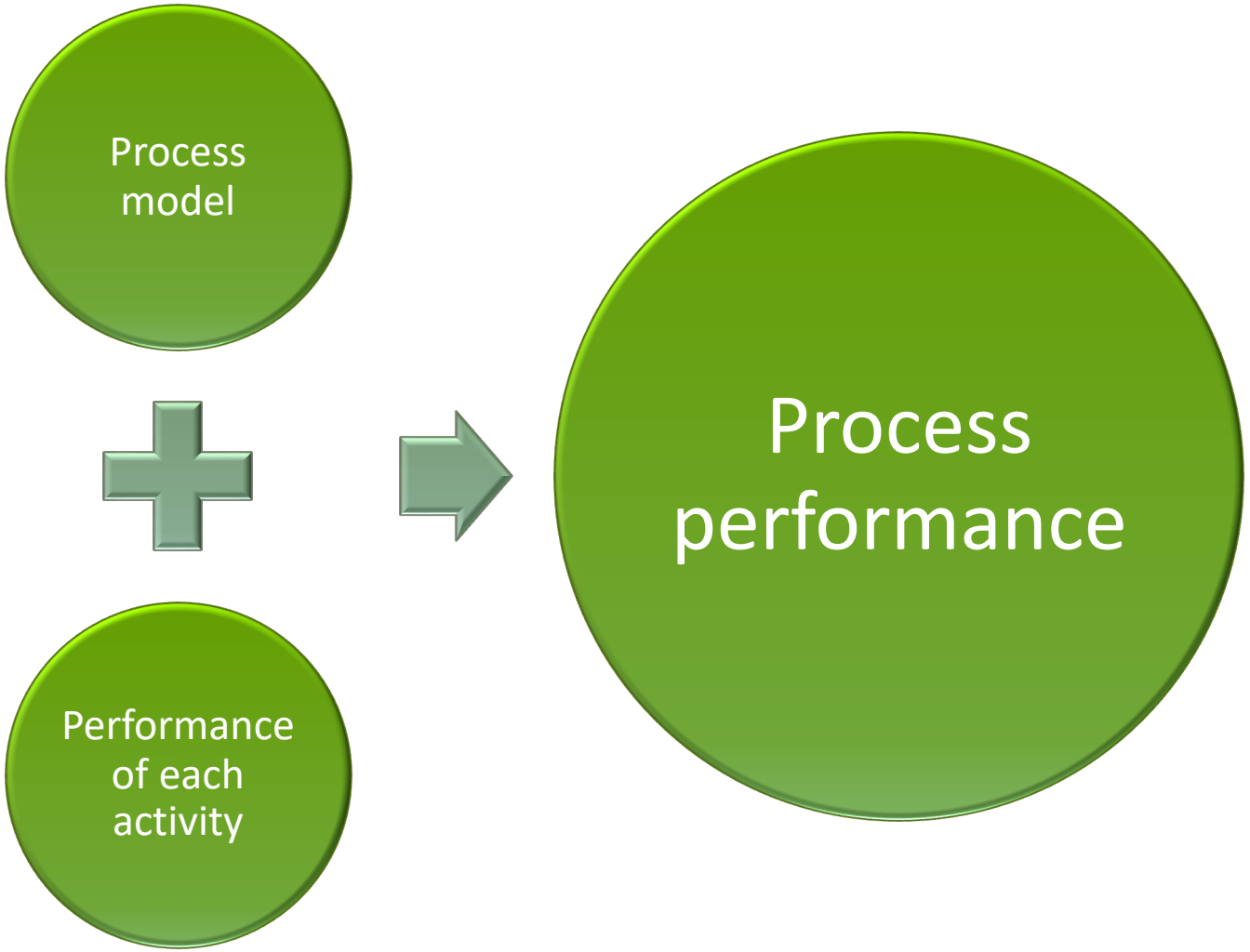
- On-time delivery rate
- Cycle time variance

Customer satisfaction

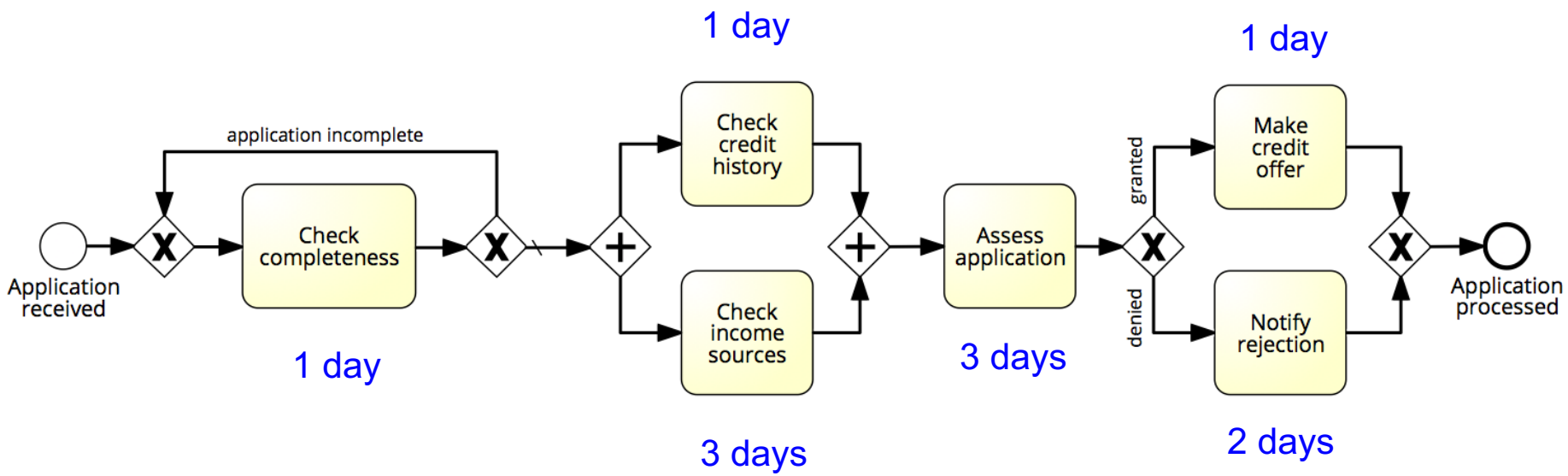
- Customer feedback score

FLOW ANALYSIS

Flow analysis



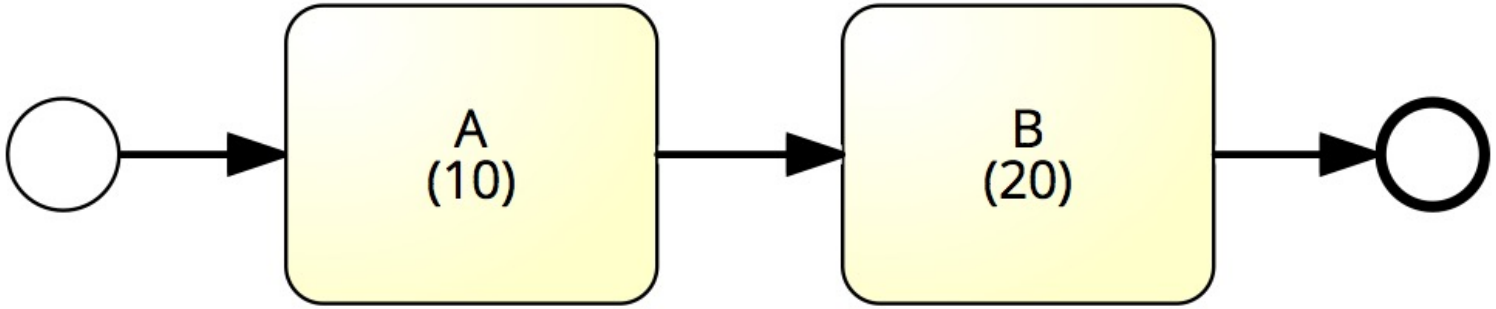
Flow analysis of cycle time



Cycle time = X days

Sequence – Example

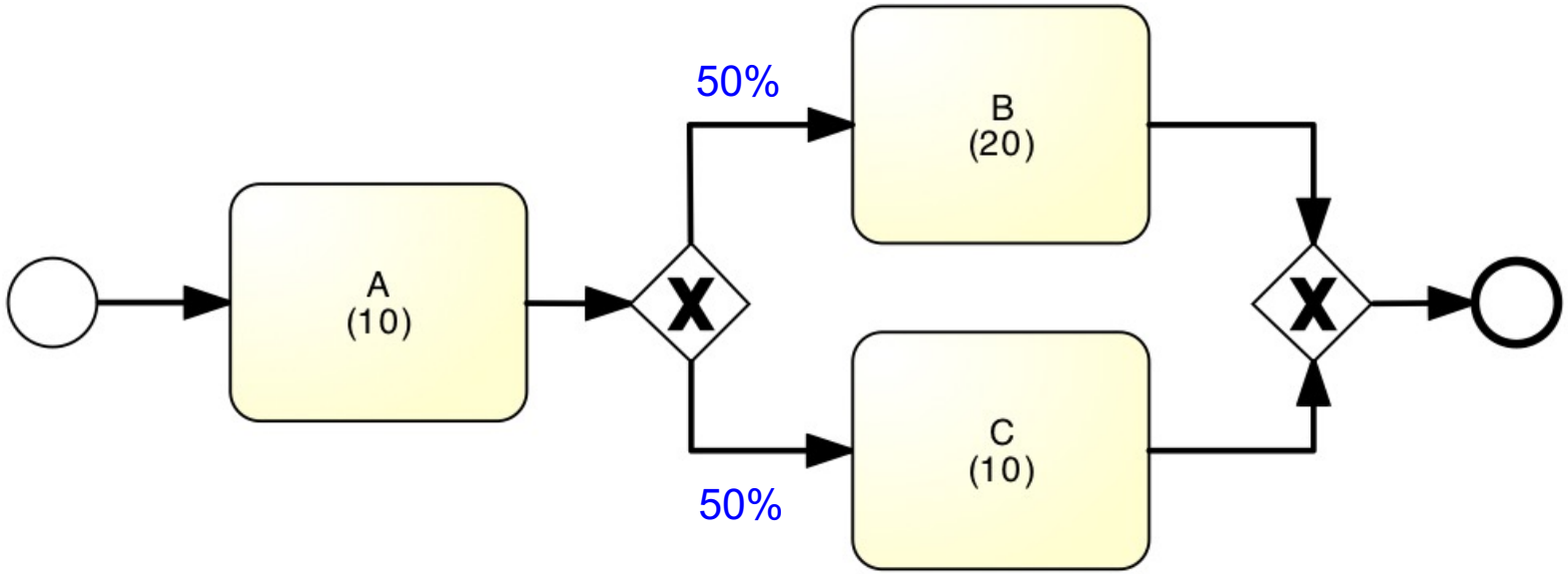
- What is the average cycle time?



Cycle time = 10 + 20 = 30

Example: Alternative Paths

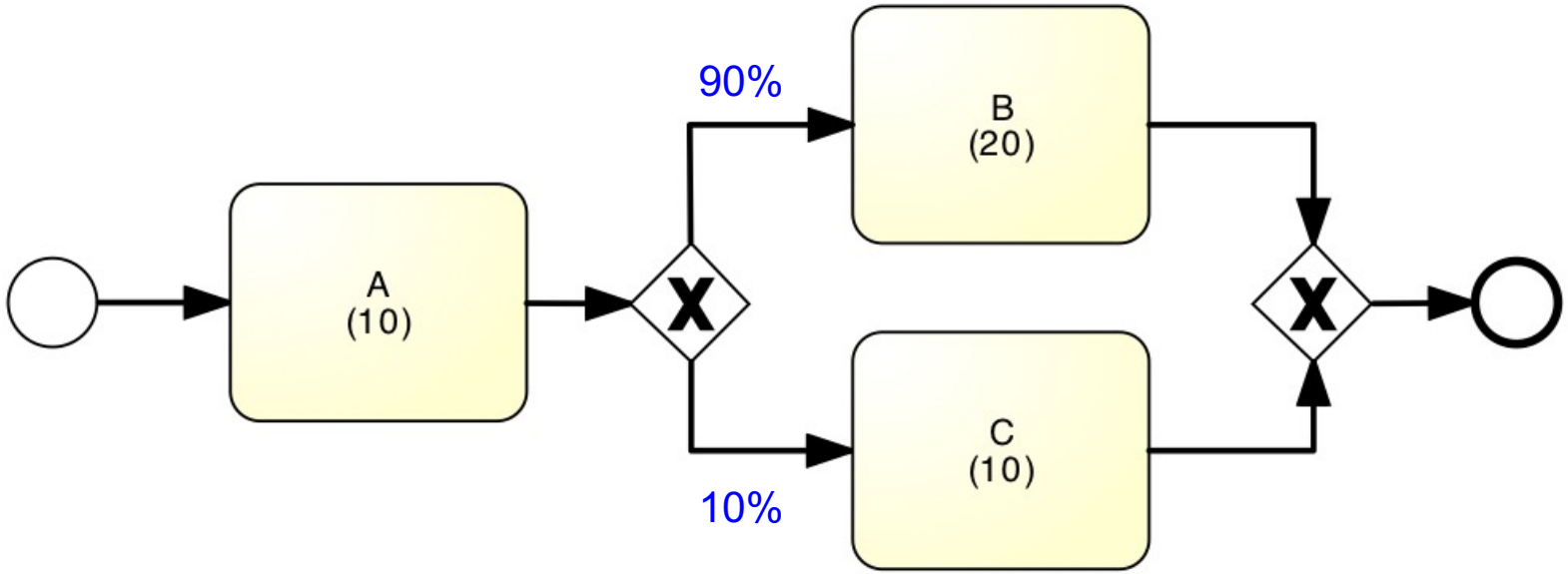
- What is the average cycle time?



Cycle time = 10 + (20+10)/2 = 25

Example: Alternative Paths

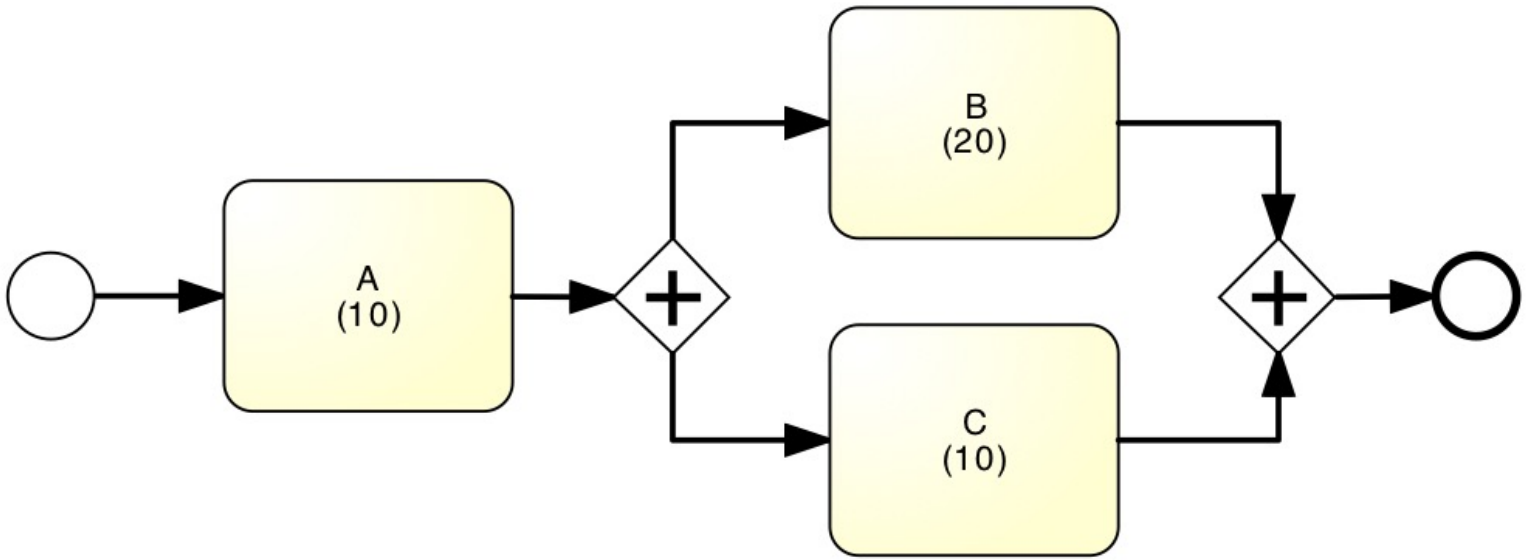
- What is the average cycle time?



Cycle time = 10 + 0.9*20+0.1*10 = 29

Example: Parallel paths

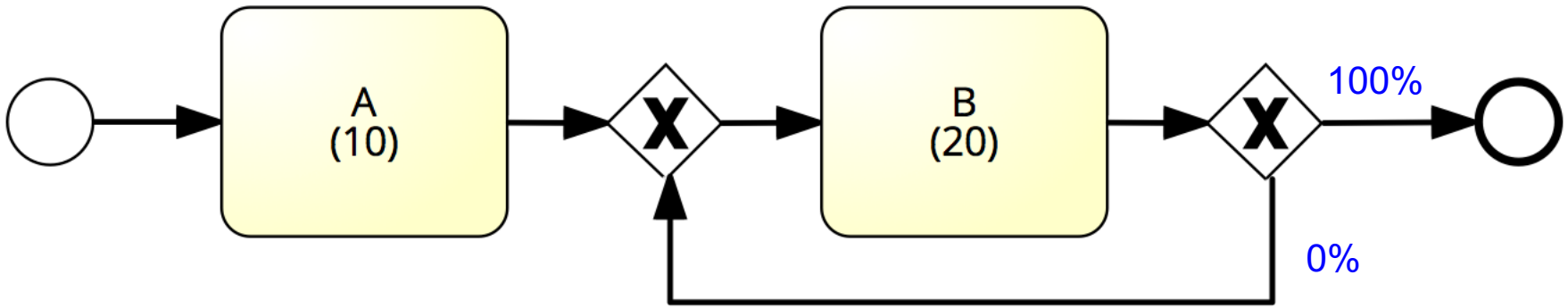
- What is the average cycle time?



Cycle time = 10 + 20 = 30

Example: Rework loop

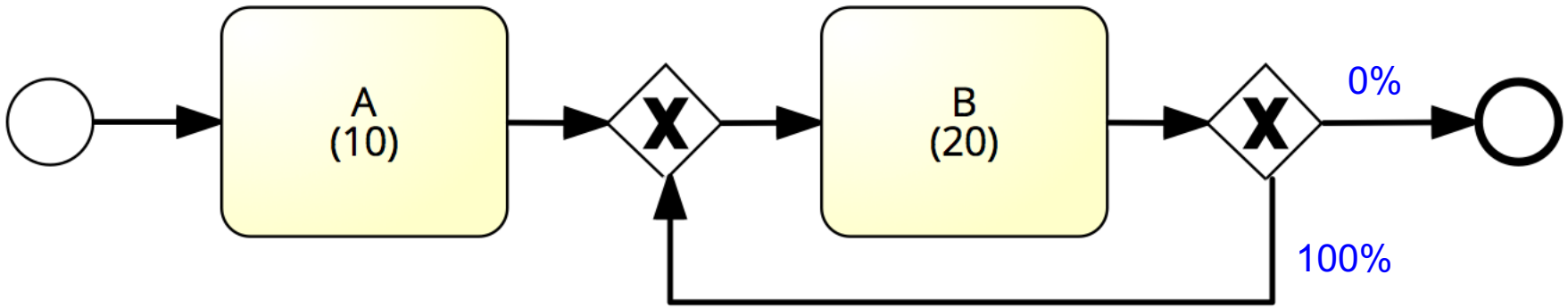
- What is the average cycle time?



Cycle time = 10 + 20 = 30

Example: Rework loop

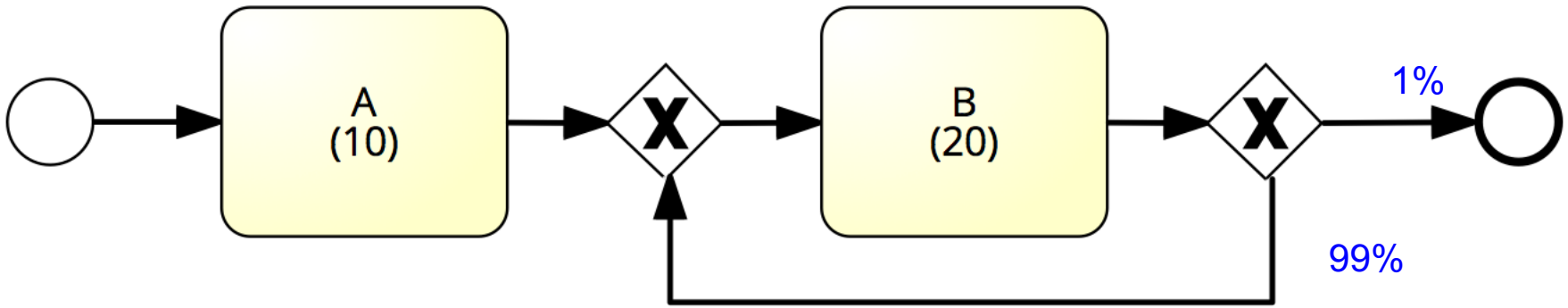
- What is the average cycle time?



Cycle time = ∞

Example: Rework loop

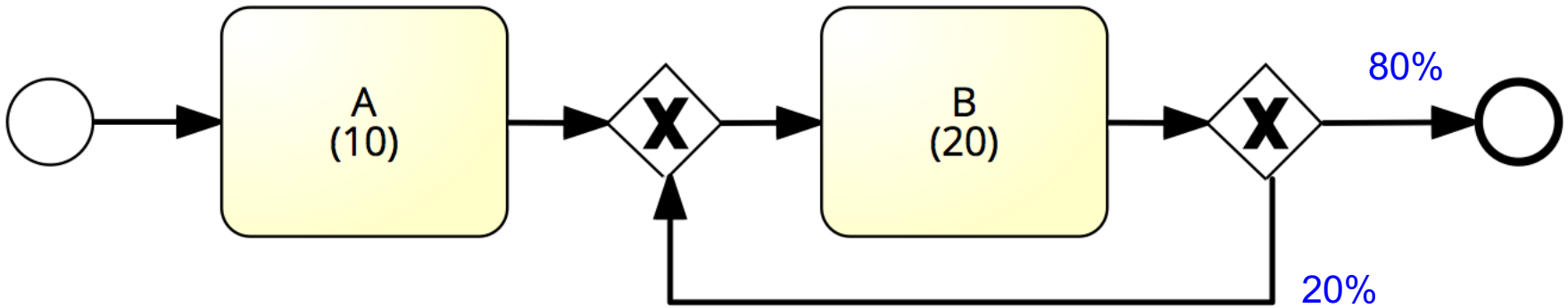
- What is the average cycle time?



Cycle time = $10 + 20*(1+0.99+0.99^2+0.99^3+\ldots) =$
$= 10 + 20/(1-0.99) = 2010$

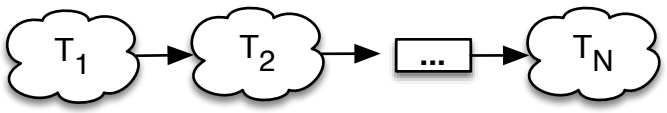
Example: Rework loop

- What is the average cycle time?

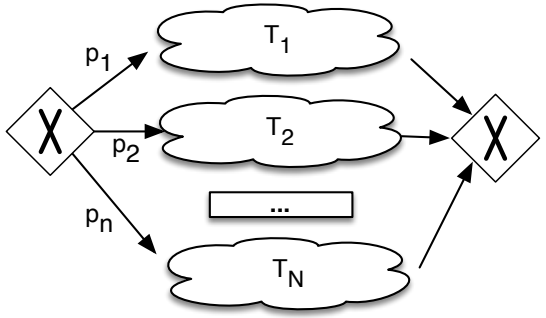


Cycle time = 10 + 20/0.8 = 35

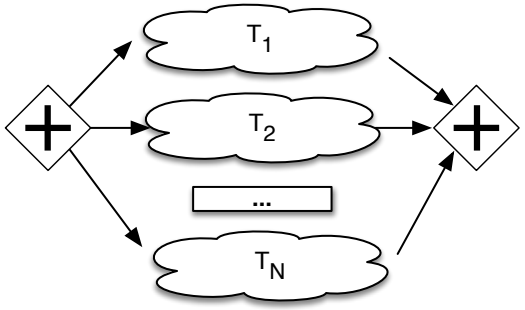
Flow analysis equations for cycle time



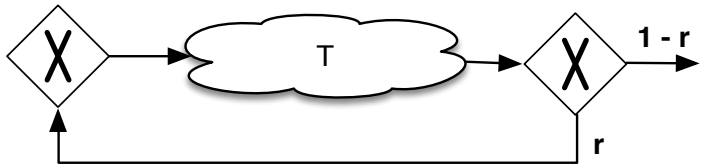
$$CT = T_1 + T_2 + \ldots + T_N$$



$$CT = p_1 * T_1 + p_2 * T_2 + \ldots + p_n * T_N$$

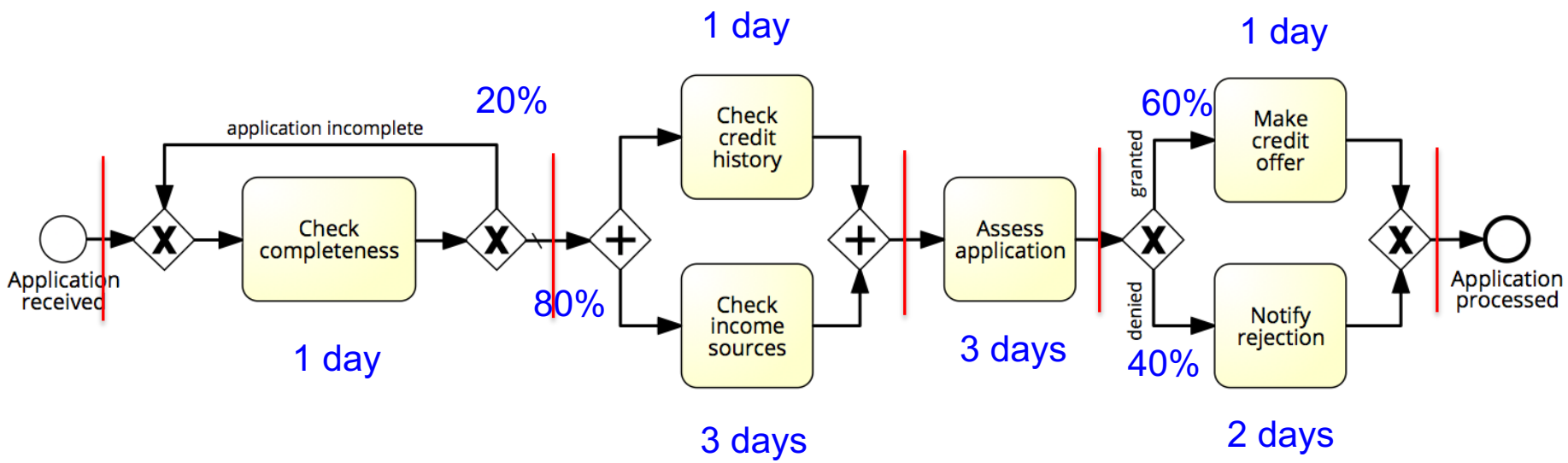


$$CT = \max(T_1, T_2, \ldots, T_N)$$



$$CT = T / (1-r)$$

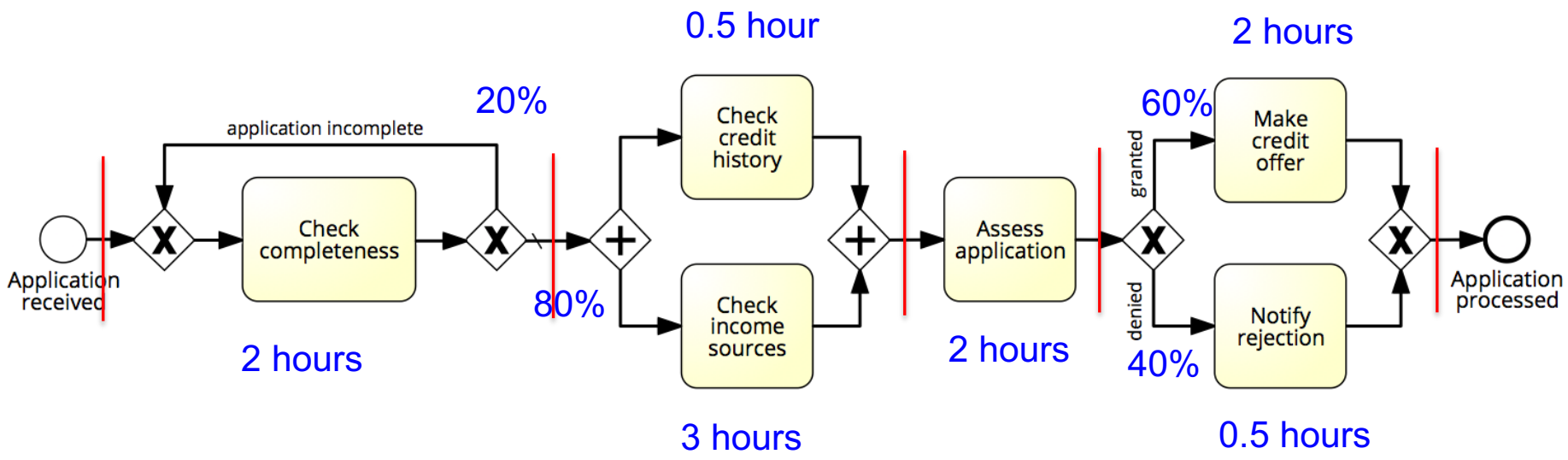
Flow analysis of cycle time



1/0.8 max(1,3) 3 0.6*1+0.4*2

Cycle time = 1.25 + 3 + 3 + 1.4 = 8.65 days

Flow analysis of processing time

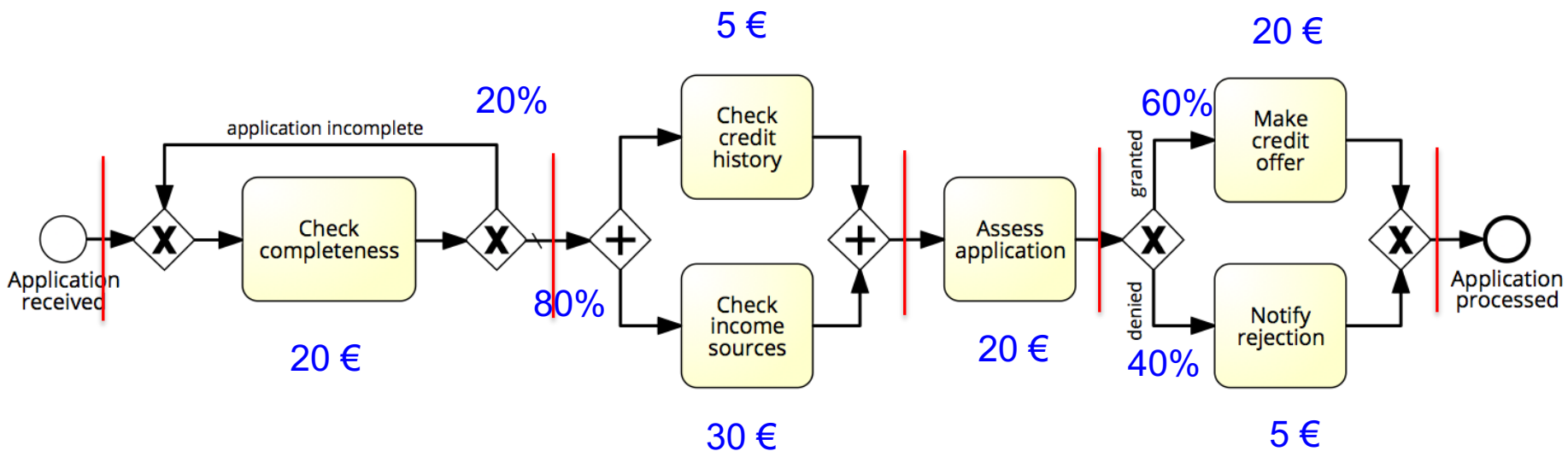


2/0.8 max(0.5,3) 2 0.6*2+0.4*0.5

Processing time = 2.5 + 3 + 2 + 1.4 = 8.9 hours

Cycle time efficiency = 8.9 hours / 8.65 days = 12.9%

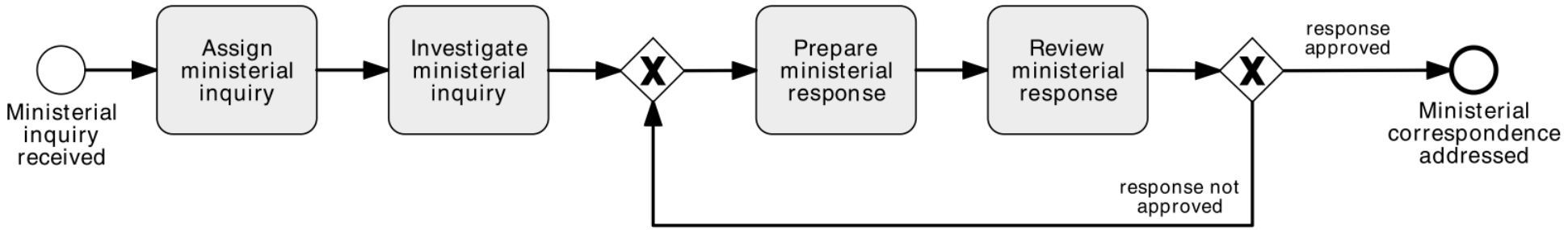
Flow analysis of cost



20/0.8 $\boxed{5 + 30}$ 20 0.6*20+0.4*5

Cost = 25 + 35 + 20 + 14 = 94 €

Exercise: Calculate the Cycle Time Efficiency

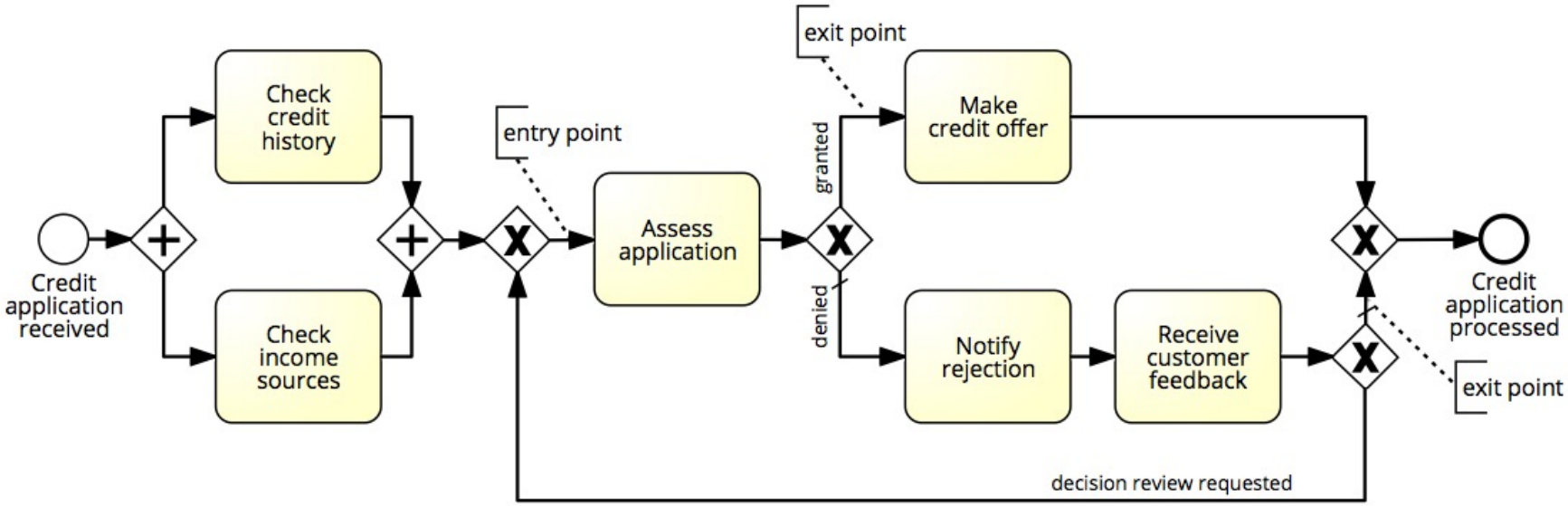


Activity	Cycle time	Processing time
Register ministerial enquiry	2 days	30 mins
Investigate ministerial enquiry	8 days	12 hours
Prepare ministerial response	4 days	4 hours
Review ministerial response	4 days	2 hour

Flow analysis: scope and limitations

- Flow analysis for cycle time calculation
- Other applications:
  - Calculating cost-per-process-instance
  - Calculating error rates at the process level
  - Estimating capacity requirements
- But it has its limitations…

Limitation 1: Not all Models are Structured



Limitation 2: Fixed arrival rate capacity

- Cycle time analysis does not consider:
  - The rate at which new process instances are created (arrival rate)
  - The number of available resources
- Higher arrival rate at fixed resource capacity
  ➔ high resource contention
  ➔ higher activity waiting times (longer queues)
  ➔ higher activity cycle time
  ➔ higher overall cycle time
- The slower you are, the more people have to queue up…
  - and vice-versa

Process Analysis Techniques

Qualitative analysis

- Value-Added & Waste Analysis
- Root-Cause Analysis
- Pareto Analysis
- Issue Register

Quantitative Analysis

- Flow analysis
- Queuing analysis
- Simulation

Why flow analysis is not enough?

Flow analysis does not consider waiting times due to resource contention

Queuing analysis and simulation address these limitations and have a broader applicability

Exercise

A fast-food restaurant receives on average 1200 customers per day (between 10:00 and 22:00). During peak times (12:00-15:00 and 18:00-21:00), the restaurant receives around 900 customers in total, and 90 customers can be found in the restaurant (on average) at a given point in time. At non-peak times, the restaurant receives 300 customers in total, and 30 customers can be found in the restaurant (on average) at a given point in time.

1. What is the average time that a customer spends in the restaurant during <u>peak</u> times?

   Batch= (900/90) = 10
   6 hrs/10 = 36 min

2. What is the average time that a customer spends in the restaurant during <u>non-peak</u> times?
3. The restaurant plans to launch a marketing campaign to attract more customers. However, the restaurant's capacity is limited and becomes too full during peak times. What can the restaurant do to address this issue without investing in extending its building?

   add more resources

Cycle Time & Work-In-Progress

- WIP = (average) Work-In-Process
  - Number of cases that are running – in other words: cases that have started but not yet completed.
  - Example: the number of active orders in an order-to-cash process.
- WIP is a form of waste

- **Little's Formula:** $L = \lambda \cdot W$
  - L = WIP
  - λ = arrival rate (number of new cases per time unit)
  - W = cycle time

Queuing Analysis

- Capacity problems are common and a key driver of process redesign
  - Need to balance the cost of increased capacity against the gains of increased productivity and service
- Queuing and waiting time analysis is particularly important in service systems
  - Large costs of waiting and/or lost sales due to waiting

Prototype Example – ER at a Hospital

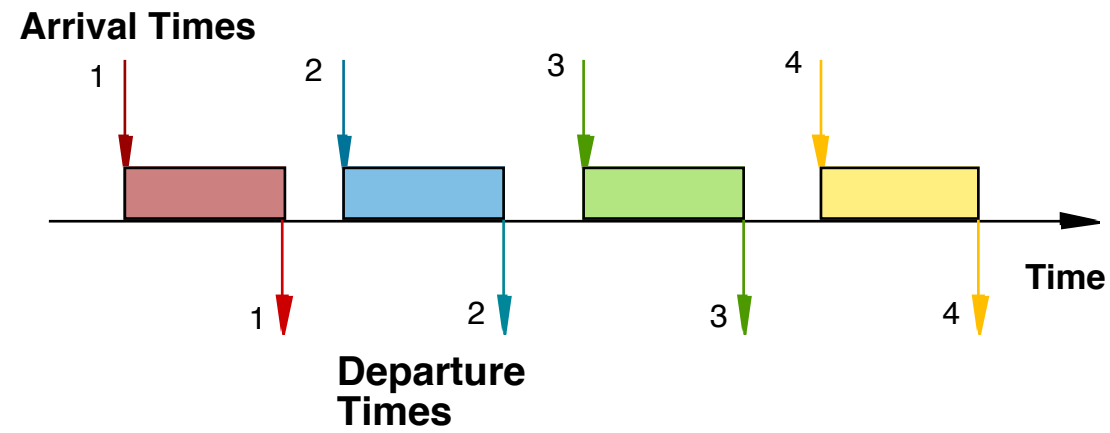
- Patients arrive by ambulance or by their own accord
- One doctor is always on duty
- More patients seeks help ⇒ longer waiting times

➢ ***Question***: ***Should another MD position be instated?***

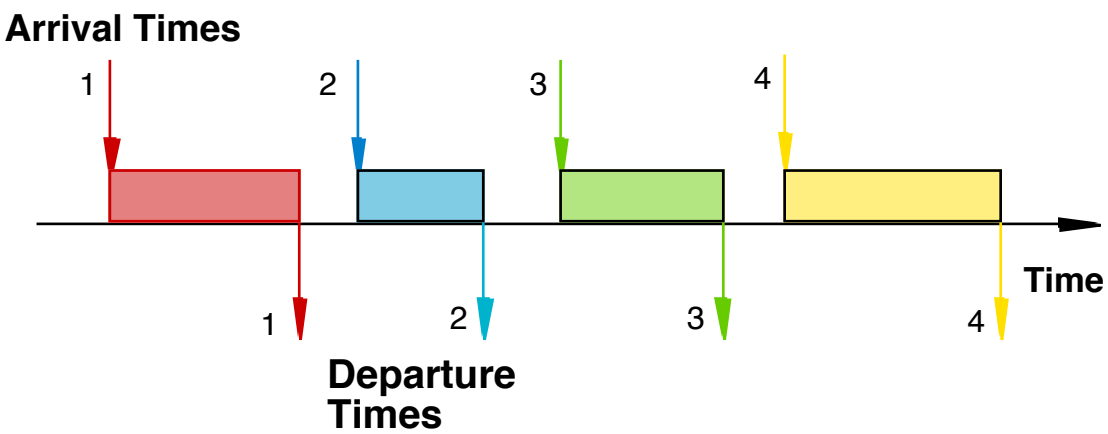
© Laguna & Marklund

Delay is Caused by Job Interference

If arrivals are regular or sufficiently spaced apart, no queuing delay occurs



Deterministic traffic

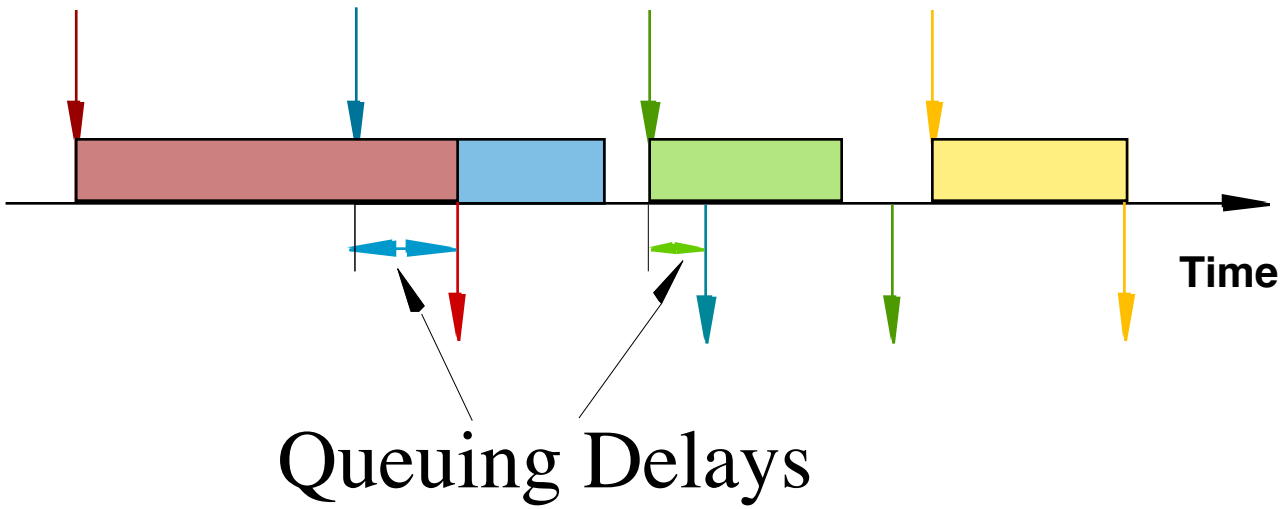


Variable but spaced apart traffic

© Dimitri P. Bertsekas

Job Size Variation Causes Interference

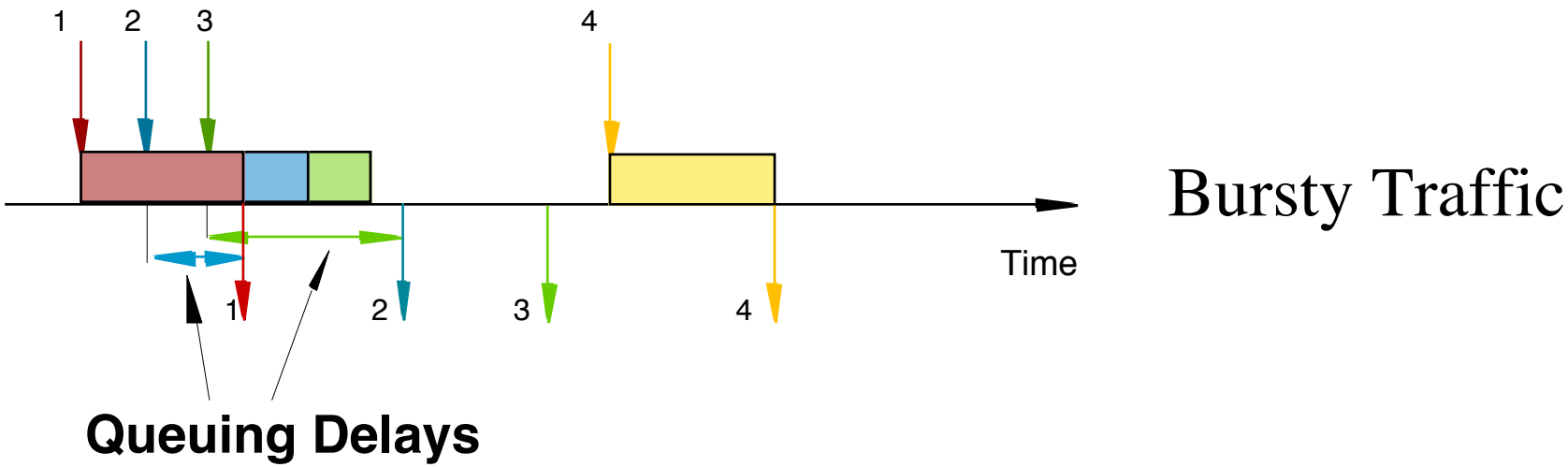
- Deterministic arrivals, variable job sizes



© Dimitri P. Bertsekas

Burstiness Causes Interference

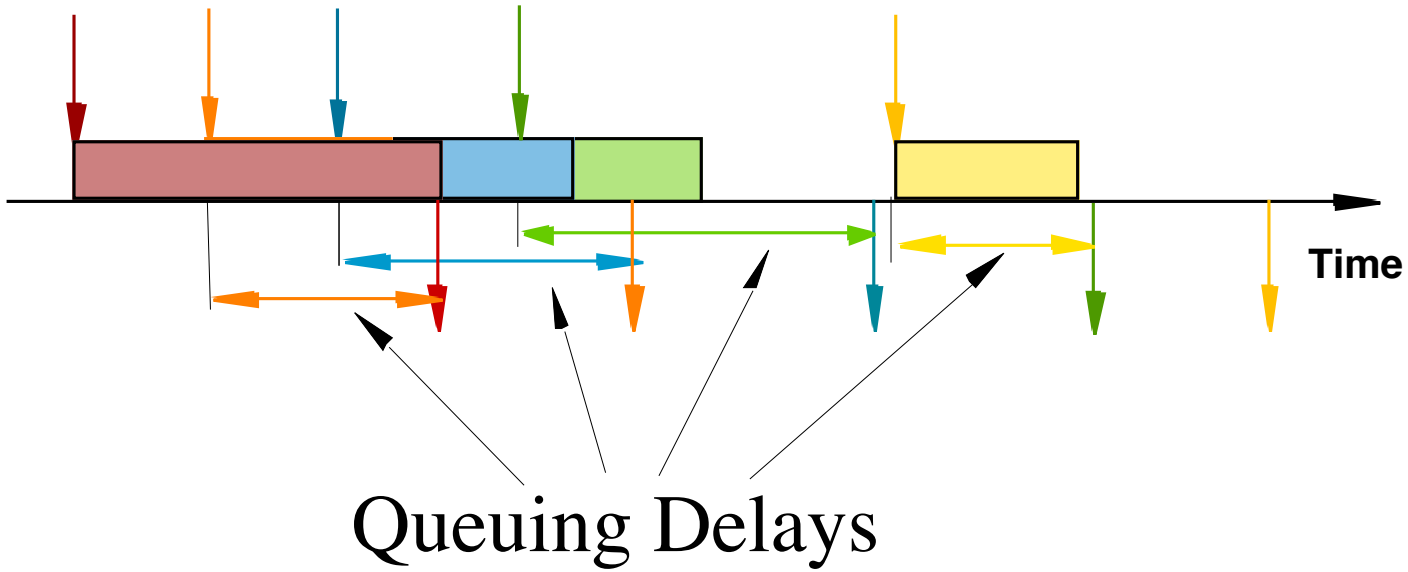
- Deterministic job size, variable arrivals



© Dimitri P. Bertsekas

High Utilization Exacerbates Interference

- The queuing probability increases as the load increases
- Utilization close to 100% is unsustainable → too long queuing times



© Dimitri P. Bertsekas

The Poisson Process

- A Poisson Process is a model for a series of discrete events where the *average time* between events is known, but the exact timing of events is random.
- The arrival of an event is independent of the event before

Example Poisson Process with average time between events of 60 days

https://towardsdatascience.com/the-poisson-distribution-and-poisson-process-explained-4e2cb17d459

The Poisson Process

- Common arrival assumption in many queuing and simulation models

- has a single parameter:
- λ is the **arrival rate** (number of events per time unit)

- Expected number of events in a time interval of length T is λT

- if the average time between events is 60 minutes, what is λ?
  - $\lambda = 1/60$

The Poisson Process

Key property: **no memory**

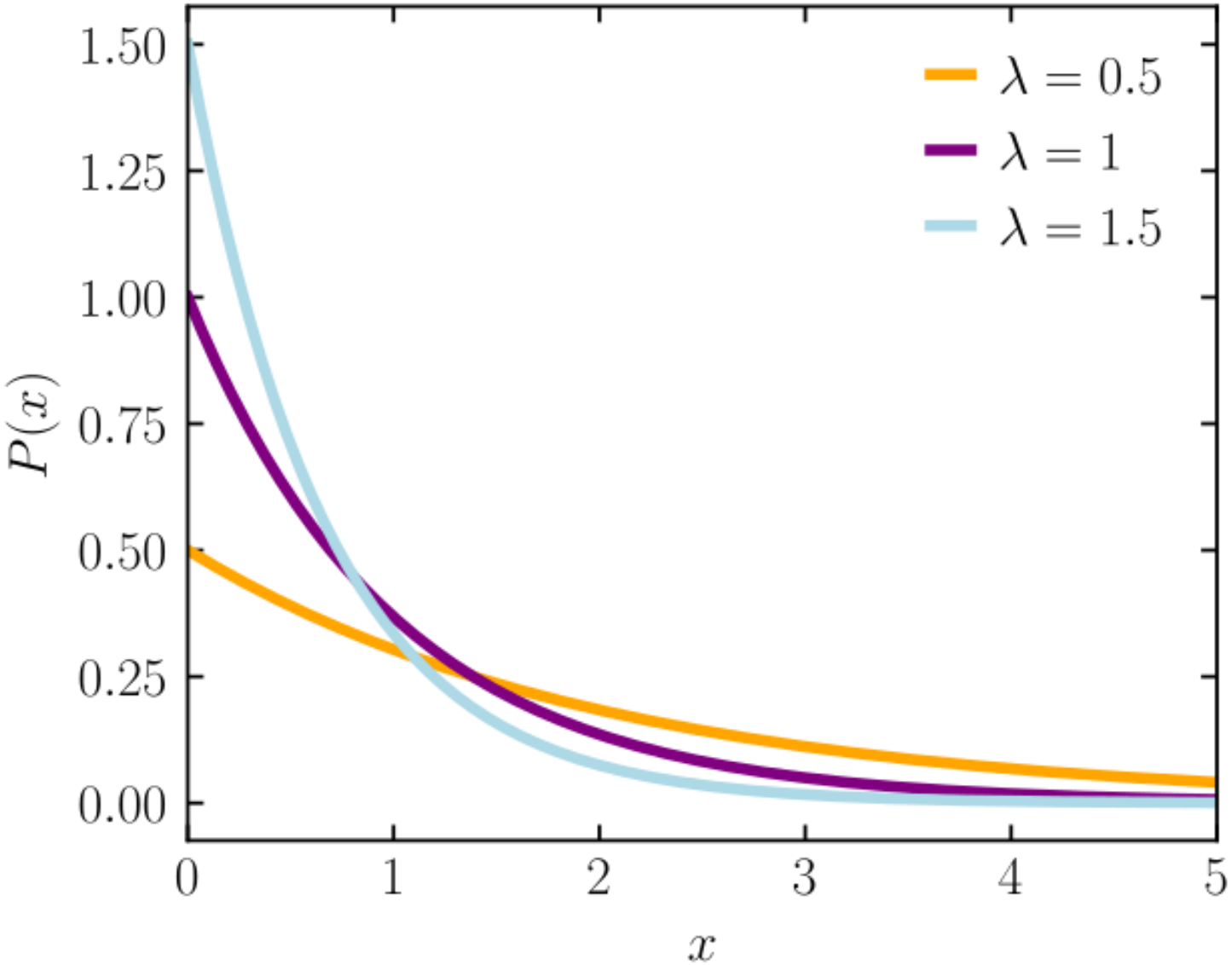
- The fact that a certain event has not happened tells us nothing about how long it will take before it happens
- Let T represent the random variable denoting the time elapsed since the last event (=inter-arrival time)

$$P(T>t+s \mid T>s) = P(T>t)$$

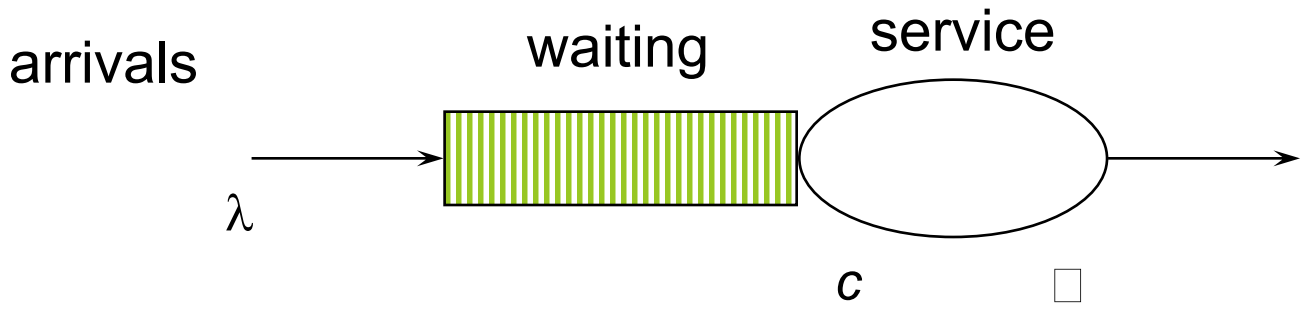
  - If we've already waited s minutes without seeing an event, the probability that an event won't occur in the next t minutes is the same as if we hadn't already waited s minutes

- The times between arrivals are independent, identically distributed and follow the **exponential distribution**
  - $P (T > t) = e^{-\lambda t}$

Probability Density Function of Exponential Distribution



Queuing theory: basic concepts

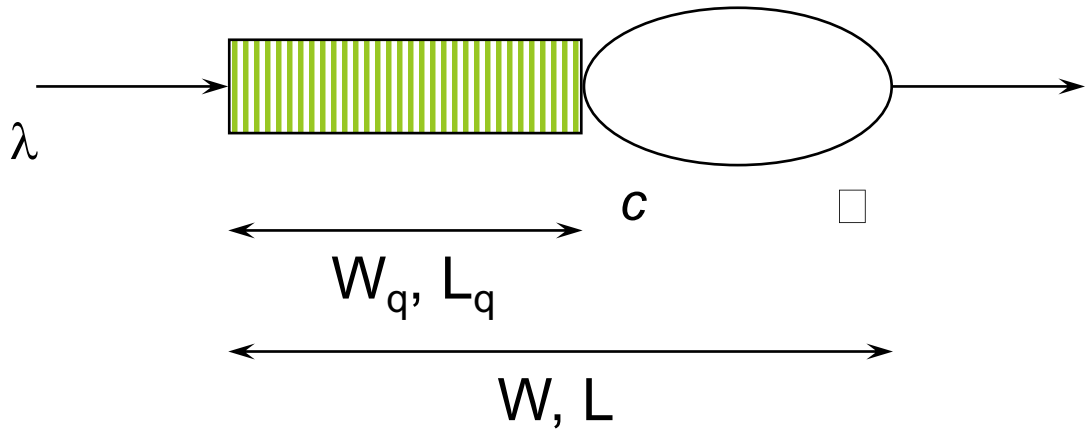


Basic characteristics:

- λ (mean arrival rate) = average number of arrivals per time unit
- μ (mean service rate) = average number of jobs that can be handled by one server per time unit:
- *c* = number of servers

© Wil van der Aalst

Queuing theory concepts (cont.)

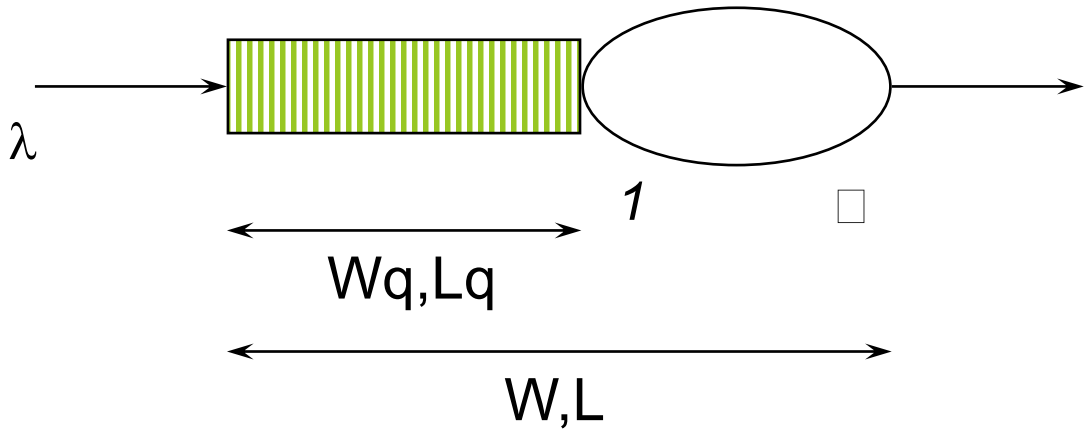


Given λ, μ, and c, we can calculate :

- ρ = resource utilization
- W_q = average time a job spends in queue (i.e., waiting time)
- W = average time in the "system" (i.e., *cycle time*)
- L_q = average number of jobs in queue (i.e., length of queue)
- L = average number of jobs in system (i.e., *Work-in-Progress*)

© Wil van der Aalst

M/M/1 queue



Assumptions:

- time between arrivals and processing time follow an exponential distribution
- 1 server (c = 1)
- FIFO

$$\rho = \frac{\text{Capacity Demand}}{\text{Available Capacity}} = \frac{\lambda}{\mu}$$

$L=\rho/(1-\rho)$

$W=L/\lambda=1/(\mu-\lambda)$

$L_q=\rho^2/(1-\rho) = L-\rho$

$W_q=L_q/\lambda=\lambda/(\mu(\mu-\lambda))$

© Laguna & Marklund

M/M/c queue

- Now there are c servers in parallel, so the expected capacity per time unit is then c*□

$$\rho \square \frac{\text{Capacity Demand}}{\text{Available Capacity}} \square \frac{\lambda}{c * \square}$$

Little's Formula $\Rightarrow$ $W_q = L_q/\lambda$

$W = W_q + (1/\square)$

Little's Formula $\Rightarrow$ $L = \lambda W$

© Laguna & Marklund

Tool Support

- For M/M/c systems, the exact computation of L_q is rather complex, but…

- **these calculations can be done by a queuing theory calculator:**
  - http://www.supositorio.com/rcalc/rcalclite.htm
  - https://qsa.inf.unideb.hu/prod/frontend/schemes/

Example – ER at County Hospital

- Situation
  - Patients arrive according to a Poisson process with intensity λ ($\Leftrightarrow$ the time between arrivals is exp(λ) distributed)
  - The service time (the doctor's examination and treatment time of a patient) follows an exponential distribution with mean 1/□ (=exp(□) distributed)

  $\Rightarrow$ *The ER can be modeled as an M/M/c system where c = the number of doctors*
- Data gathering

  $\Rightarrow$ λ = 2 patients per hour

  $\Rightarrow$ □ = 3 patients per hour

- Question
  - Should the capacity be increased from 1 to 2 doctors?

© Laguna & Marklund

Queuing Analysis – Hospital Scenario

- Interpretation
  - To be in the queue = to be in the waiting room
  - To be in the system = to be in the ER (waiting or under treatment)

Characteristic	One doctor (c=1)	Two Doctors (c=2)
ρ	2/3	1/3
L_q	4/3 patients	1/12 patients
L	2 patients	3/4 patients
W_q	2/3 h = 40 minutes	1/24 h = 2.5 minutes
W	1 h	3/8 h = 22.5 minutes

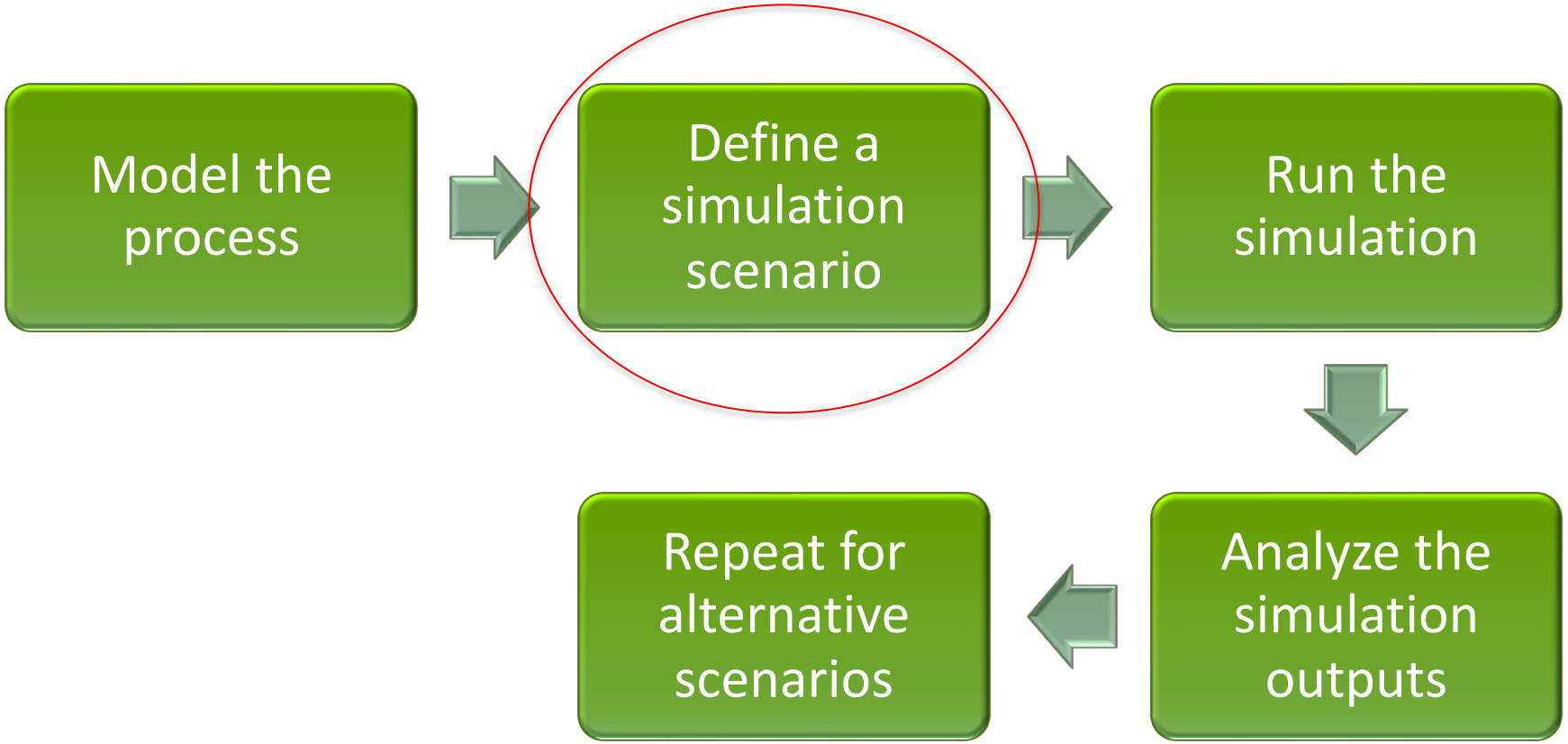
- Is it warranted to hire a second doctor ?

© Laguna & Marklund

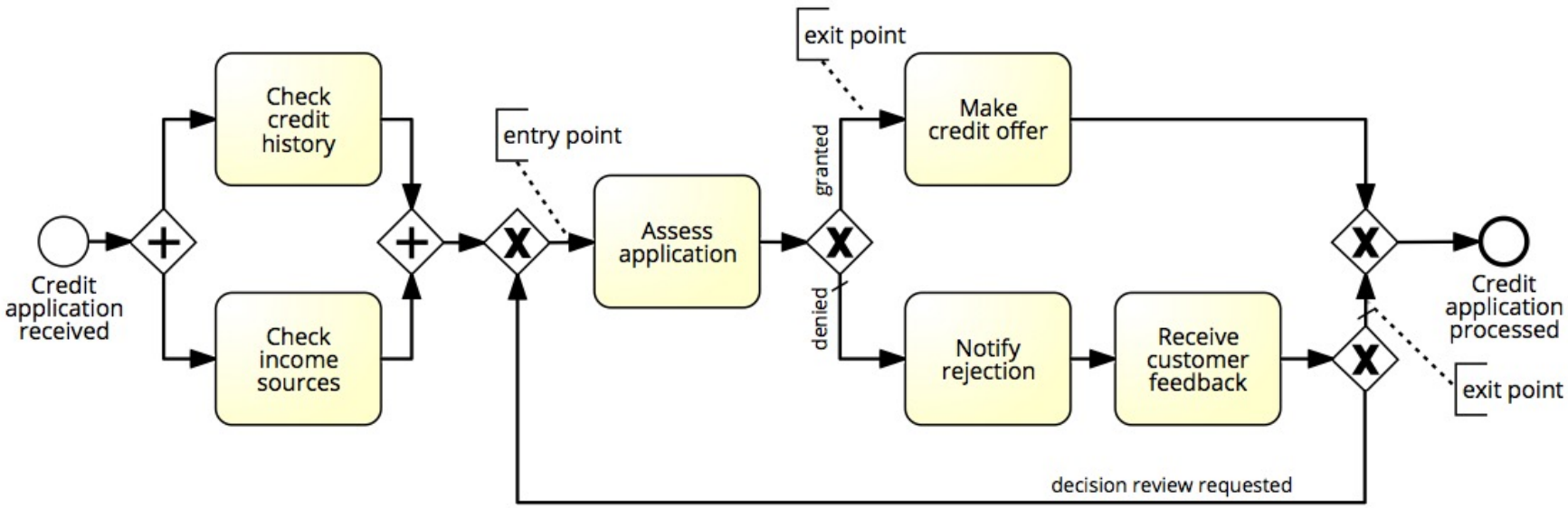
Process Simulation

- Versatile quantitative analysis method for
  - As-is analysis
  - What-if analysis
- In a nutshell:
  - Run a large number of process instances
  - Gather performance data (cost, time, resource usage)
  - Calculate statistics from the collected data

Process Simulation



Example


Credit application received
Check credit history
Check income sources
entry point
Assess application
exit point
granted
denied
Make credit offer
Notify rejection
Receive customer feedback
Credit application processed
exit point
decision review requested

Elements of a simulation scenario

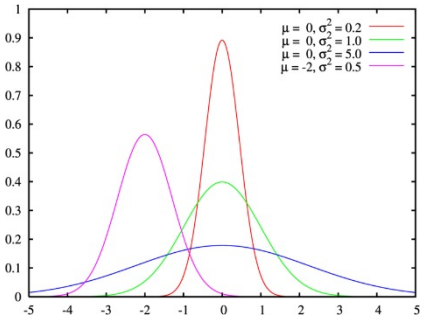
1. Processing times of activities
   - Fixed value
   - Probability distribution

Choice of probability distribution

Fixed

- Can be used to approximate cases where the activity processing time varies very little
- Example: a task performed by a software application

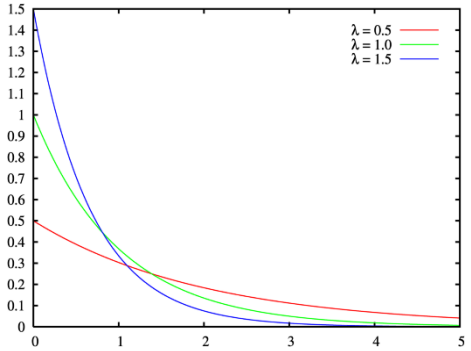
Normal



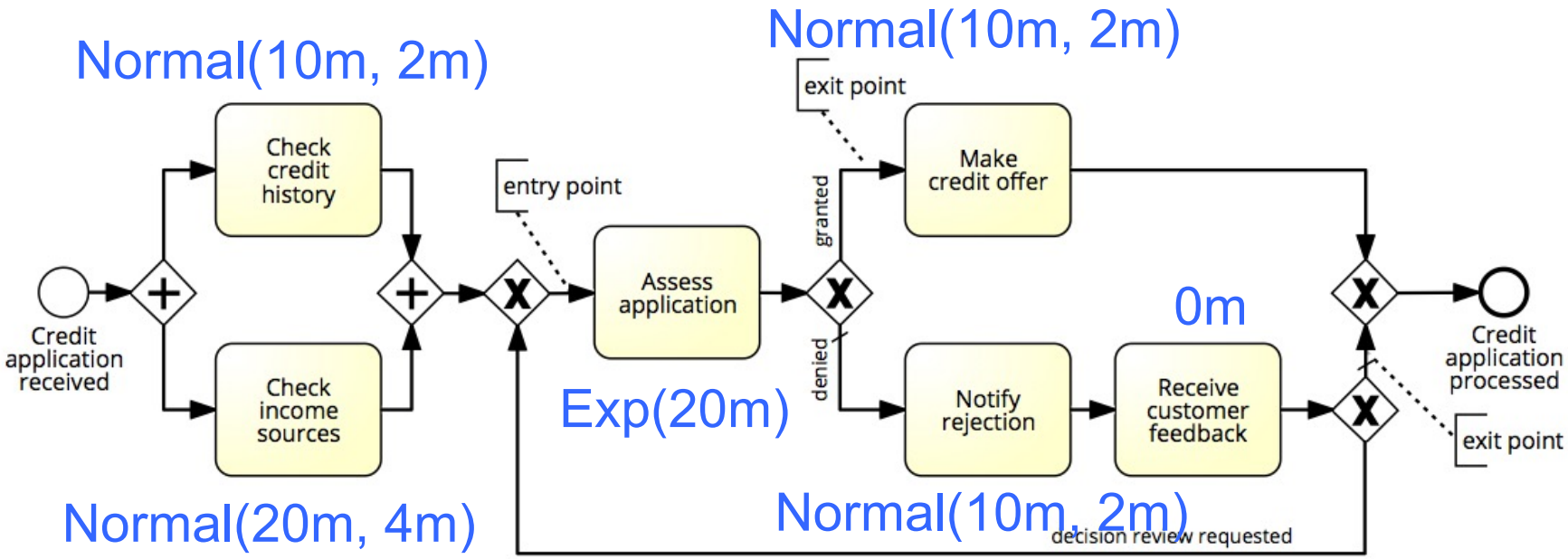
- Repetitive activities
- Example: "Check completeness of an application"
- Requires us to specify the mean and the std deviation

Exponential

- Complex activities that may involve detailed analyses or decisions
- Example: "Assess an application"
- Requires us to specify the mean only



Simulation Example

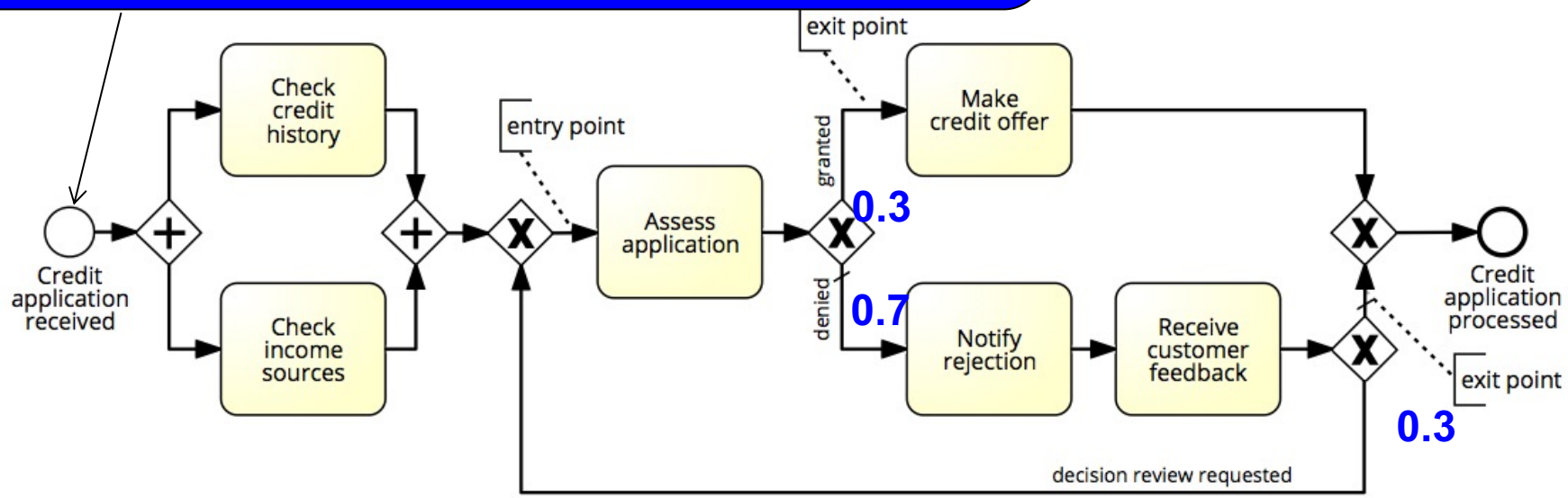


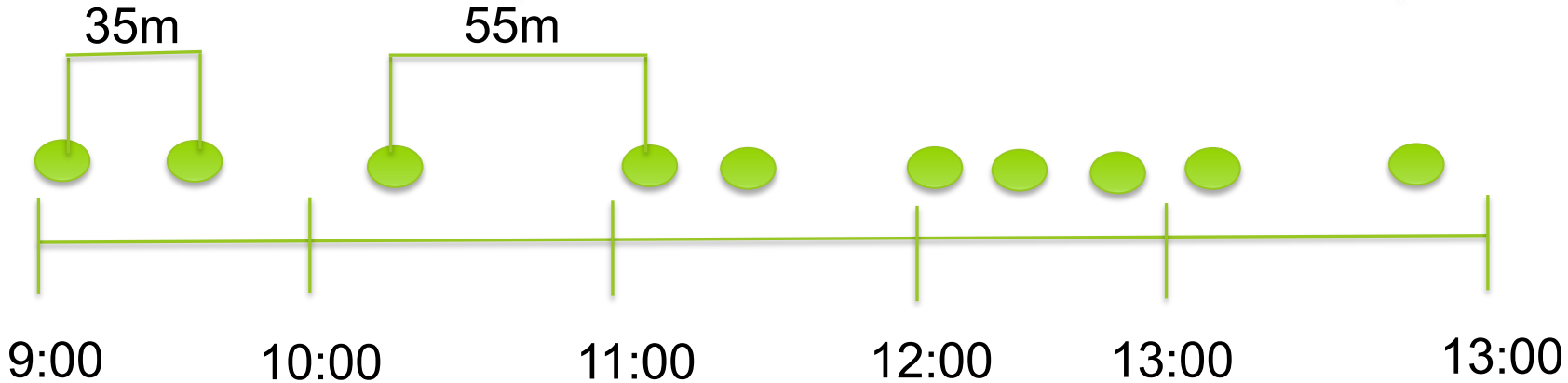
Elements of a simulation model

1. Processing times of activities
   - Fixed value
   - Probability distribution
2. Conditional branching probabilities
3. Arrival rate of process instances and probability distribution
   - Typically, exponential distribution with a given mean inter-arrival time
   - Arrival calendar, e.g., Monday-Friday, 9am-5pm, or 24/7

Branching probability and arrival rate

Arrival rate = 2 applications per hour
Inter-arrival time = 0.5 hour
Exponential distribution
From Monday-Friday, 9am-5pm





Elements of a simulation model

1. Processing times of activities
   - Fixed value
   - Probability distribution
2. Conditional branching probabilities
3. Arrival rate of process instances and probability distribution
   - Typically, exponential distribution with a given mean inter-arrival time
   - Arrival calendar, e.g., Monday-Friday, 9am-5pm, or 24/7
4. Resource pools

Resource pools

- Name
- Size of the resource pool
- Cost per time unit of a resource in the pool
- Availability of the pool (working calendar)
- Examples:

<u>Clerk</u>

€ 25 per hour

Mon-Fri, 9am-5pm

<u>Credit Officer</u>

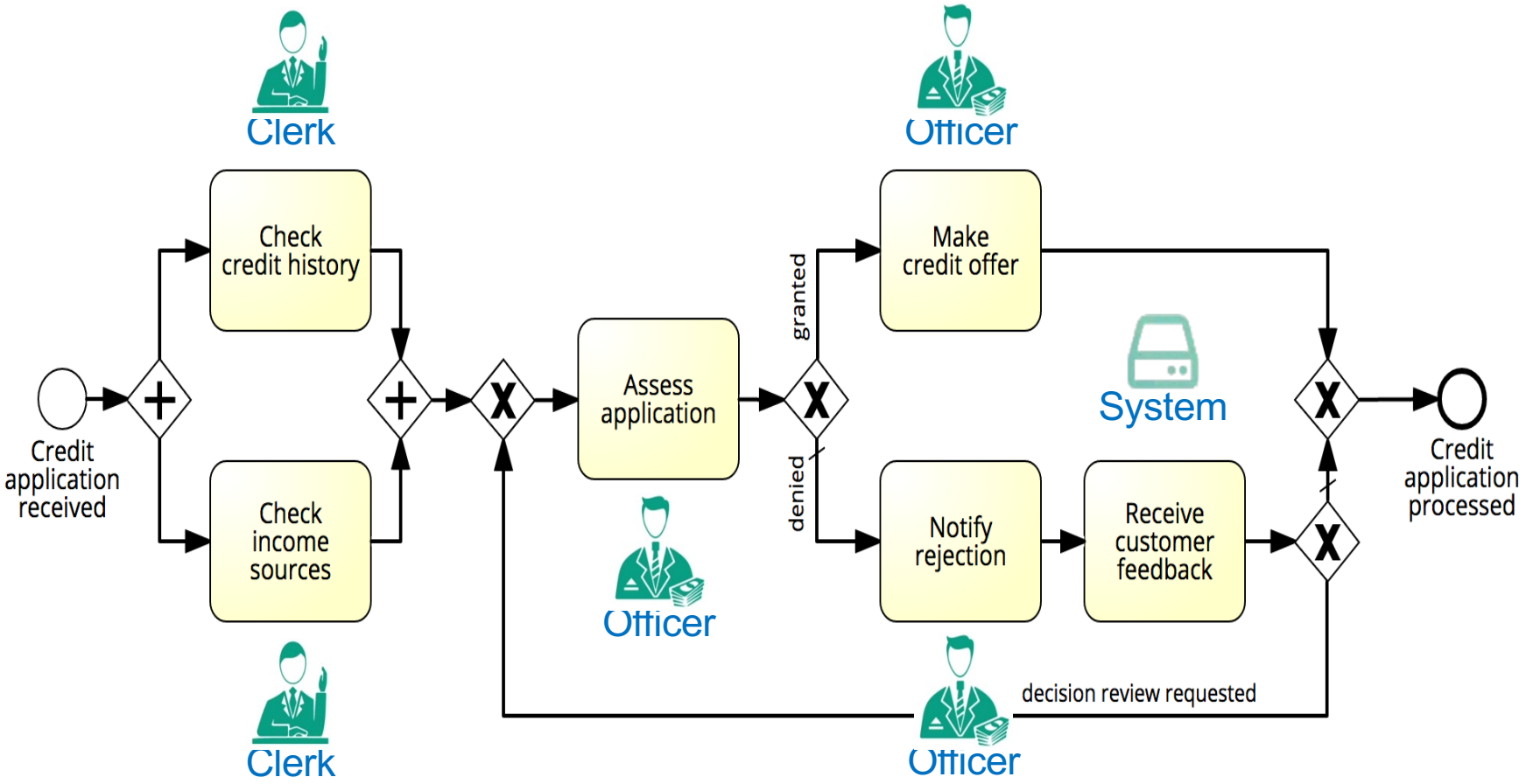
€ 35 per hour

Mon-Fri, 9am-4pm

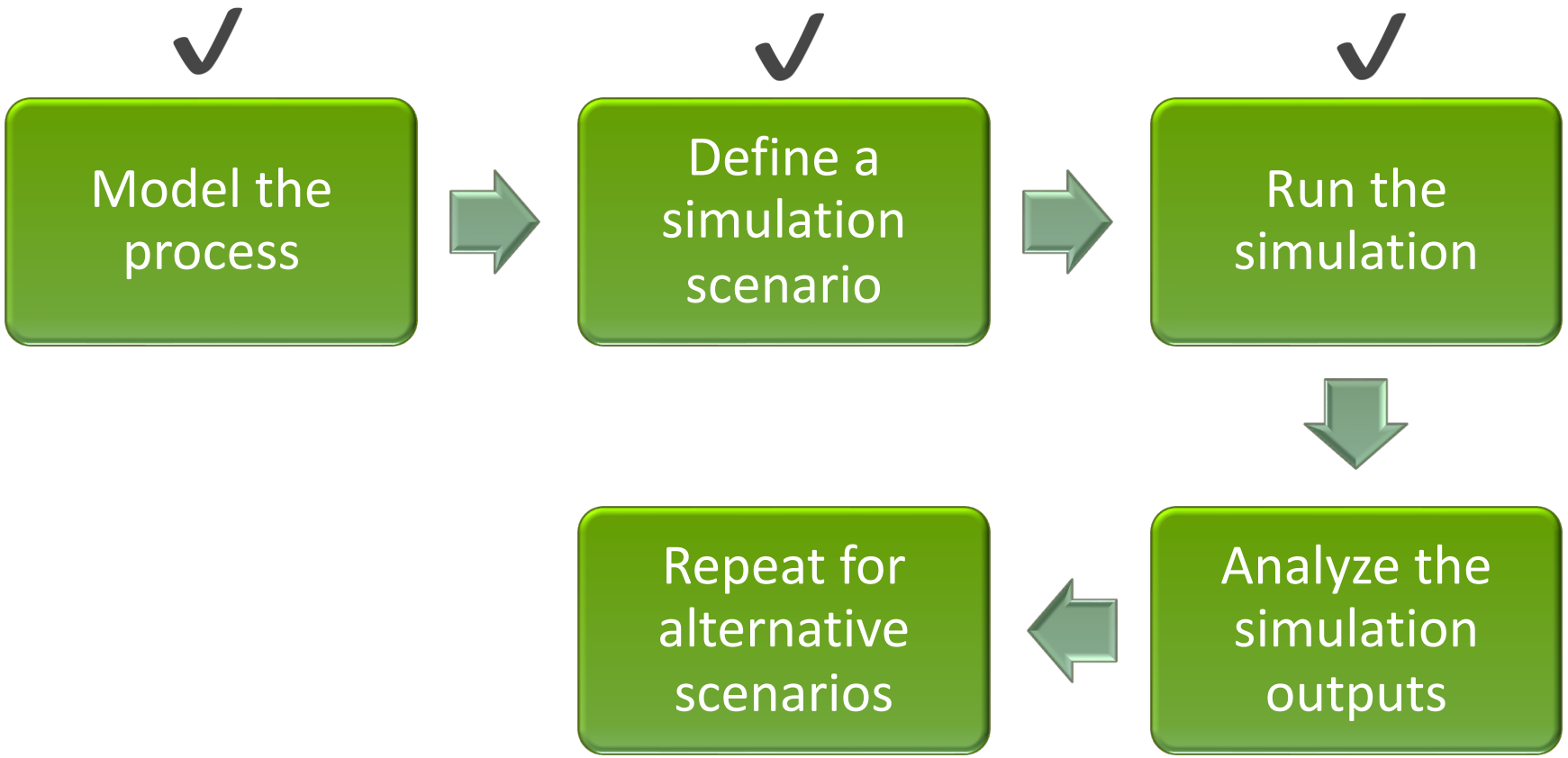
Elements of a simulation model

1. Processing times of activities
   - Fixed value
   - Probability distribution
2. Conditional branching probabilities
3. Arrival rate of process instances and probability distribution
   - Typically, exponential distribution with a given mean inter-arrival time
   - Arrival calendar, e.g., Monday-Friday, 9am-5pm, or 24/7
4. Resource pools
5. Assignment of tasks to resource pools

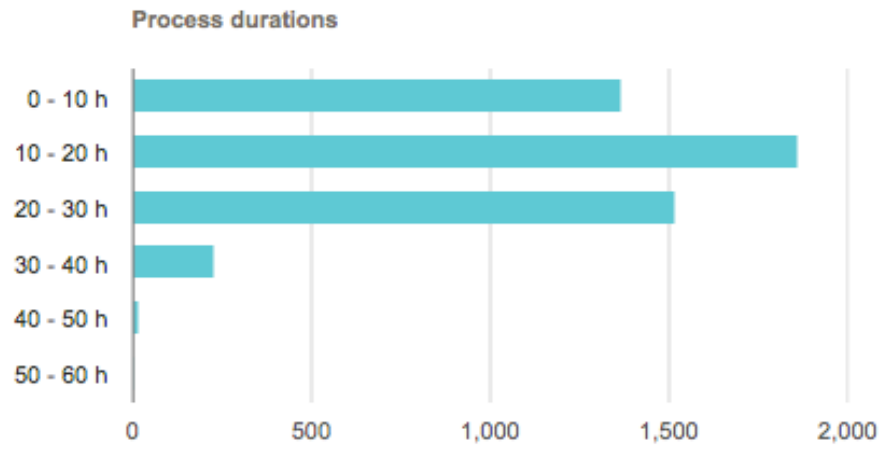
Resource pool assignment

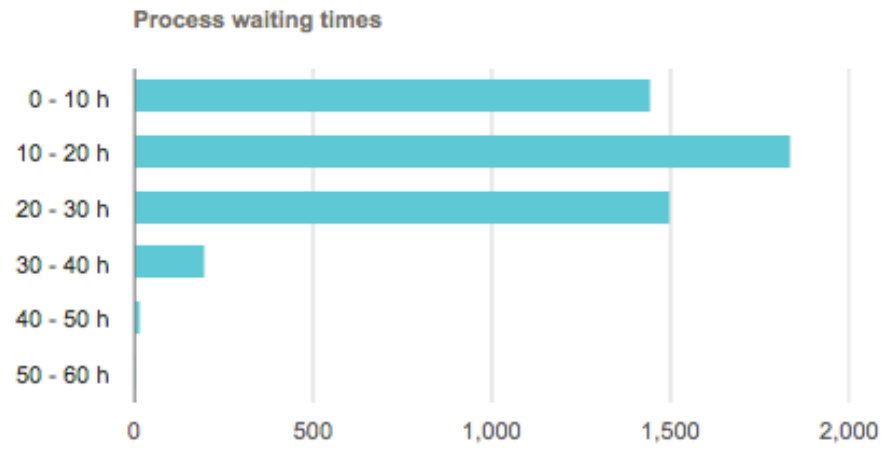


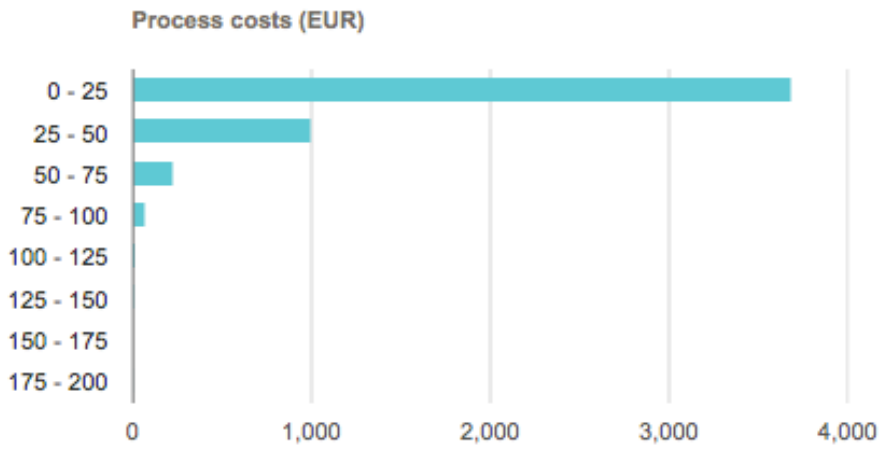
Process Simulation

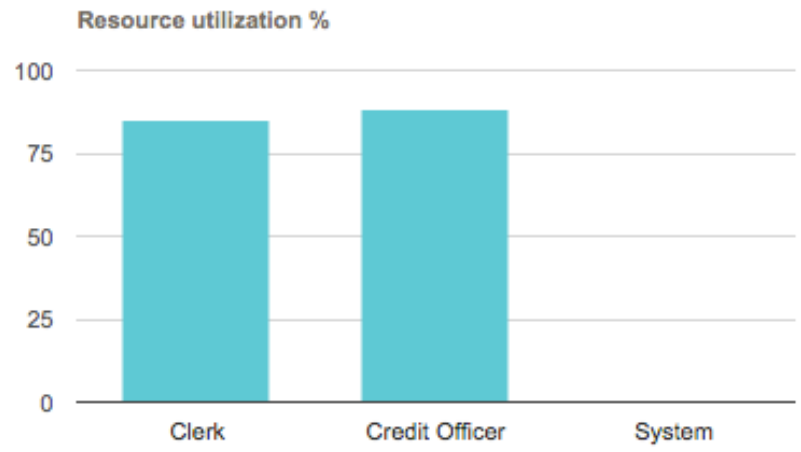


Output: Performance measures & histograms

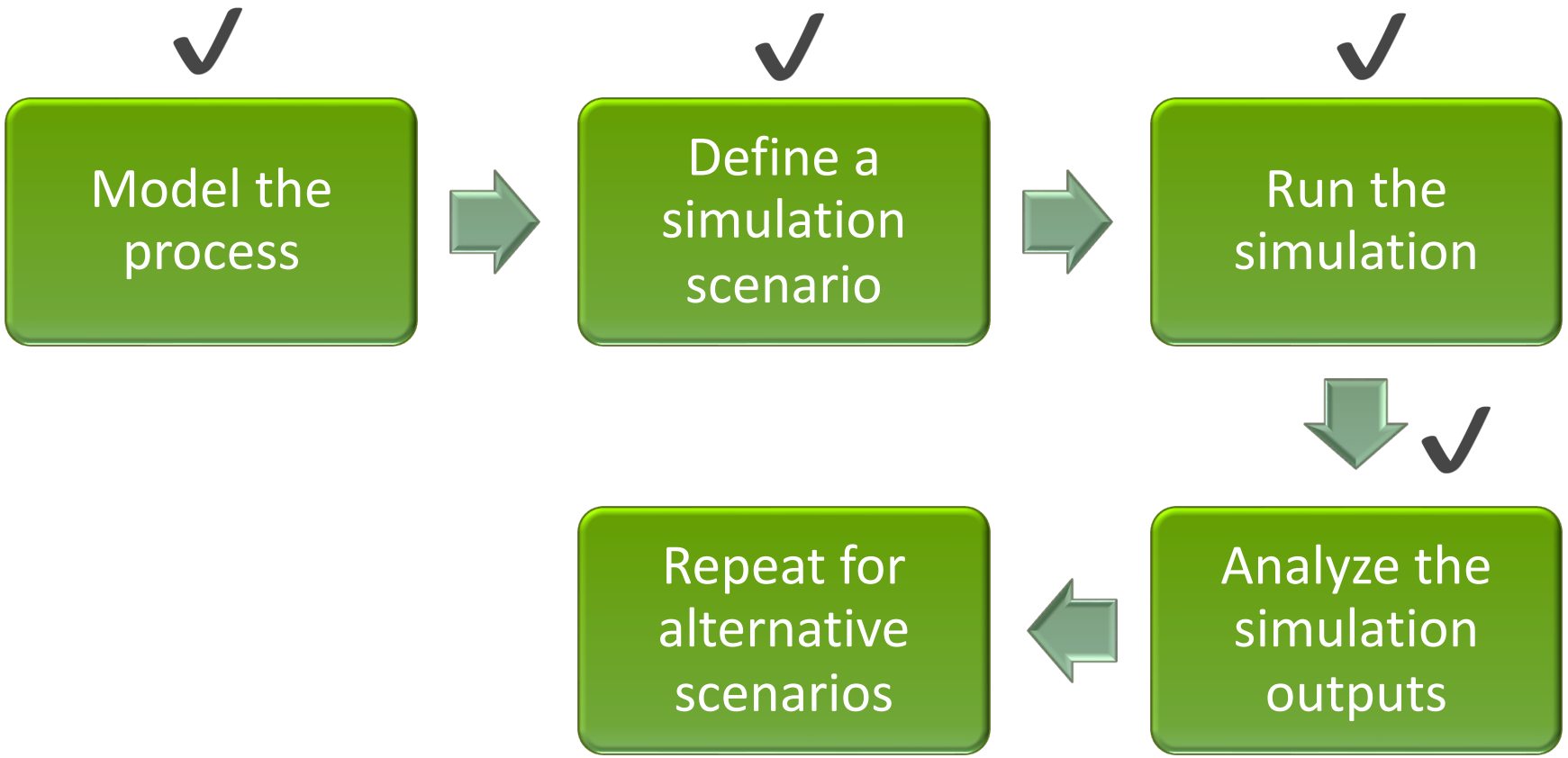




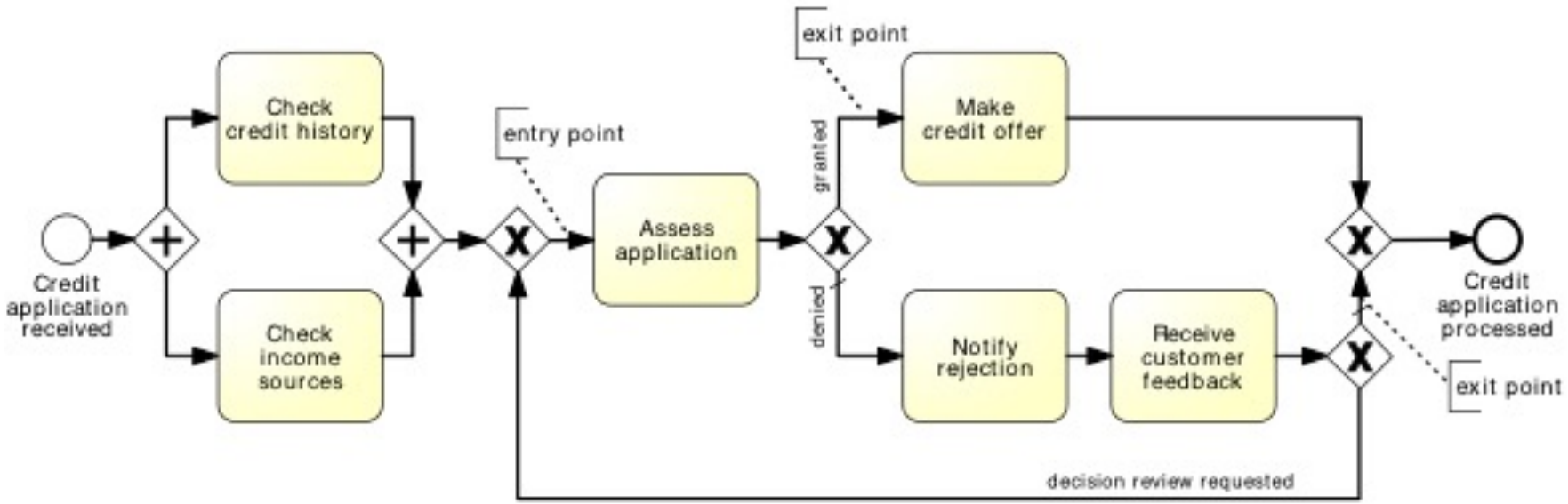




Process Simulation



Demo: Simulation in BIMP



https://bimp.cs.ut.ee/simulator/trial?sample=credit_card_application

Pitfalls of simulation

- Stochasticity
- Data quality
- Simplifying assumptions

Stochasticity

- Problem
  - Simulation results may differ from one run to another
- Solutions
  1. Make the simulation timeframe long enough to cover weekly and seasonal variability, where applicable
  2. Use multiple simulation runs, average results of multiple runs, compute confidence intervals

Data quality

- Problem
  - Simulation results are only as trustworthy as the input data
- Solutions:
  1. Rely as little as possible on “guesstimates”. Use input analysis where possible:
     - Derive simulation scenario parameters from numbers in the scenario
     - Use statistical tools to check fit the probability distributions
  2. Simulate the “as is” scenario and cross-check results against actual observations

Simulation simplifying assumptions

- That the process model is always followed to the letter
  - No deviations
  - No workarounds
- That a resource only works on one task
  - No multitasking
- That if a resource becomes available and a work item (task) is enabled, the resource will start it right away
  - No batching
- That resources work constantly (no interruptions)
  - Every day is the same!
  - No tiredness effects
  - No distractions beyond “stochastic” ones

Acknowledgements

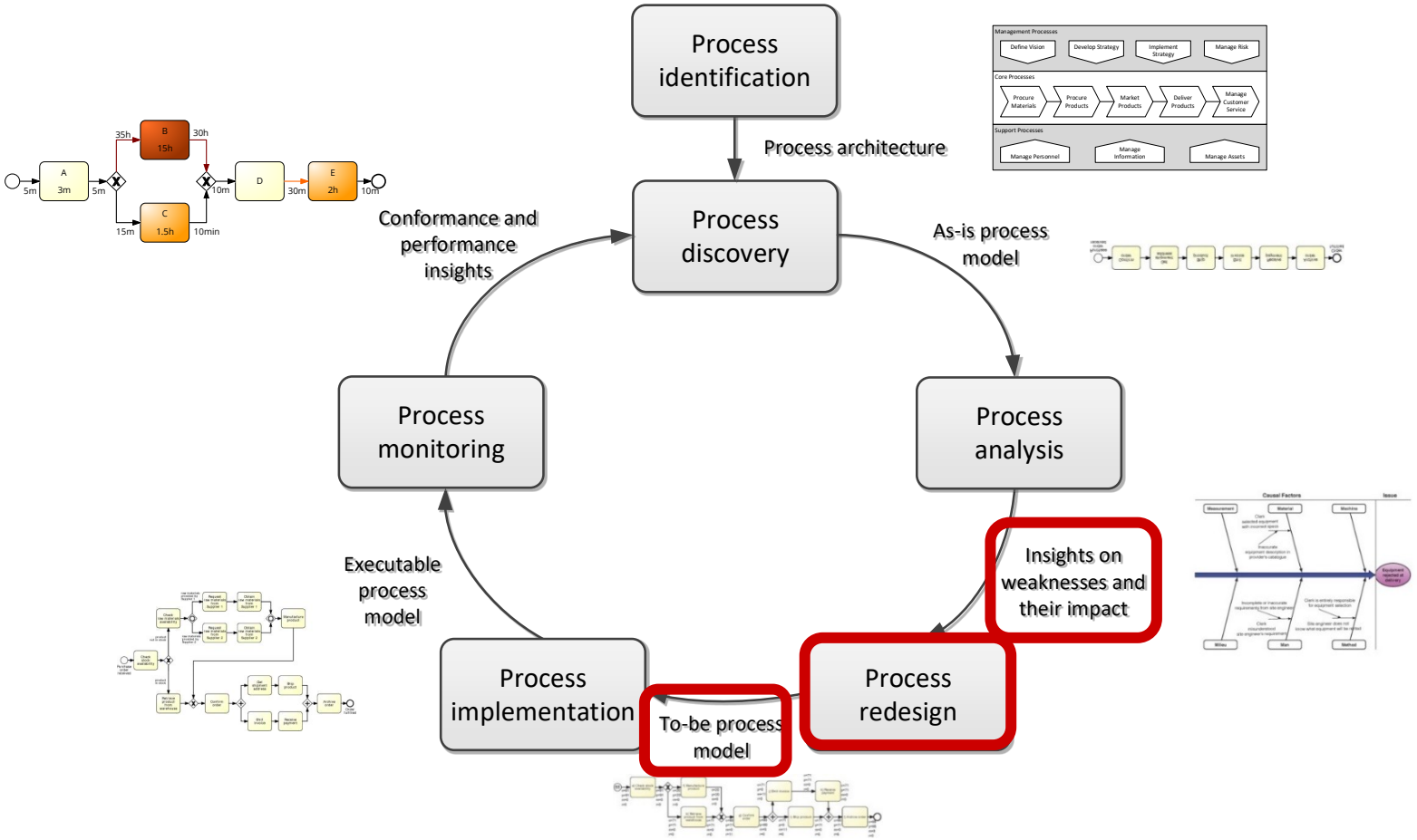
- All material comes from Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A. Reijers, authors of the “Fundamentals of Business Process Management” book.

INFO-H420
Management of Data Science and Business Workflows
Part I
5. Process Redesign

Dimitris SACHARIDIS

2023-2024

Process Redesign

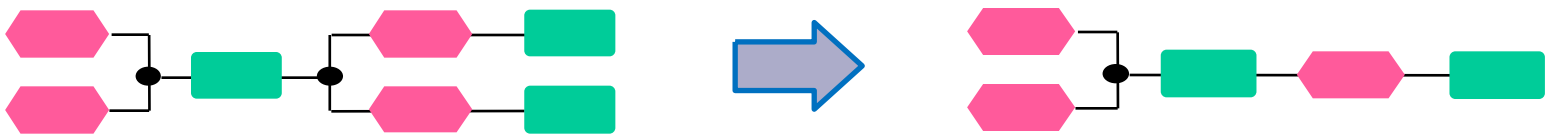


Process Redesign

Identify possibilities for improving the design of a process

AS-IS: **Descriprive** modelling of the real world

TO-BE: **Prescriptive** modelling of the real world



- No silver-bullet: requires **creativity**
- *Redesign heuristics* can be used to generate ideas

Process redesign approaches

Transformational Redesign

- Puts into question the fundamental assumptions and principles of the existing process structure
- Aims to achieve breakthrough innovation
- **Example: Business Process Reengineering (BPR)**

Transactional Redesign

- Doesn't put into question the current process structure
- Seeks to identify problems and resolve them incrementally, one step at a time
- Example: Heuristic redesign

Business Process Reengineering (BPR)

- **Transformative:** Puts into question the fundamental assumptions of the "as is" process
- **Analytical:** Based on a set of principles that foster:
  - Outcome-driven processes
  - Integration of information gathering, work and decisions

The Ford Case Study

Ford needed to review its procurement process to:

- Do it cheaper (cut costs)
- Do it faster (reduce turnaround times)
- Do it better (reduce error rates)

Accounts payable in North America alone employed > 500 people and turnaround times for processing POs and invoices was in the order of weeks

https://hbr.org/1990/07/reengineering-work-dont-automate-obliterate

The Ford Case Study

Automation would bring some improvement (20% improvement)

But Ford decided not to do it… Why?

a) Because at the time, the technology needed to automate the process was not yet available.
b) Because nobody at Ford knew how to develop the technology needed to automate the process.
c) Because there were not enough computers and computer-literate employees at Ford.

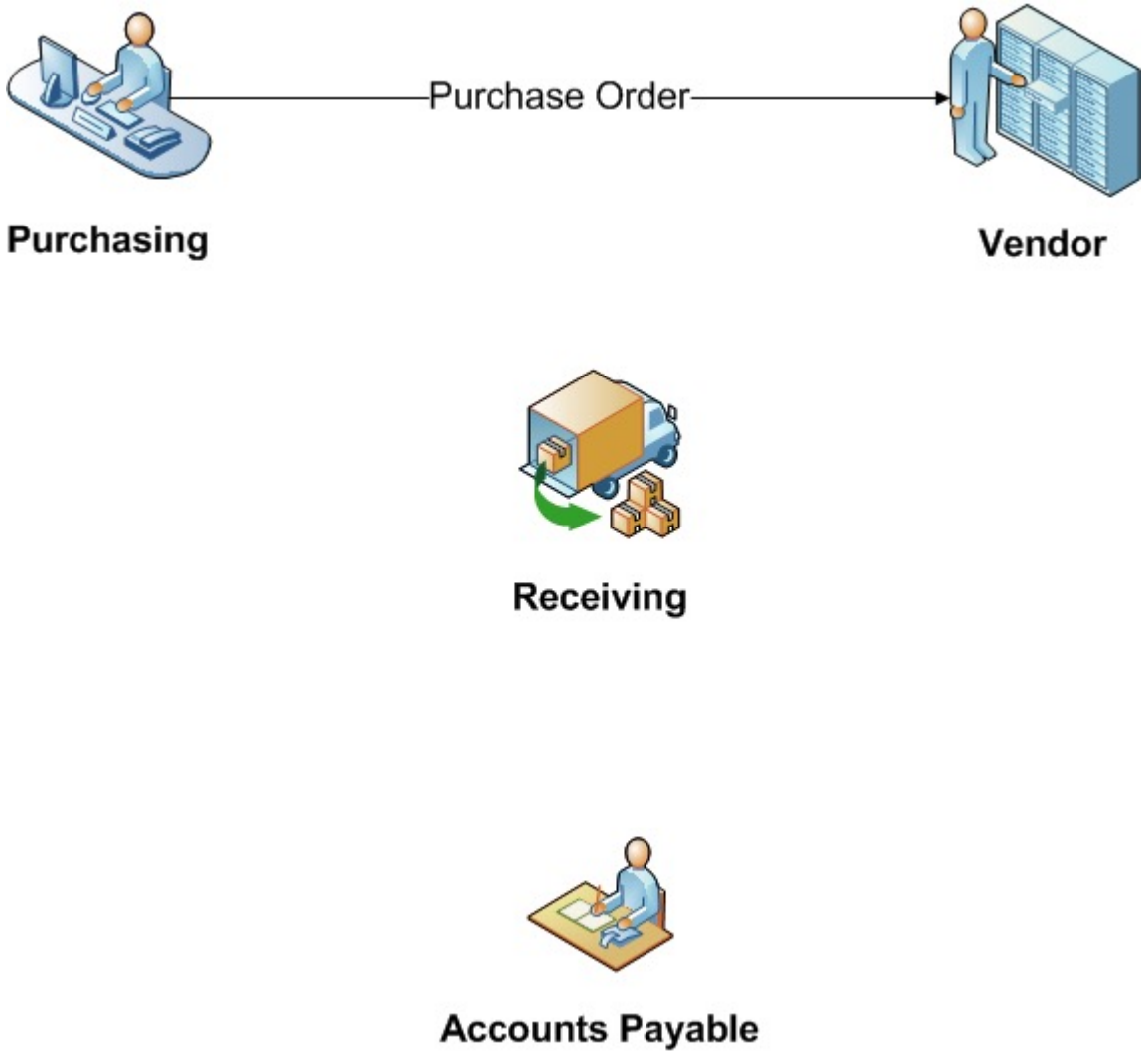
The correct answer is …

Mazda's Accounts Payable Department

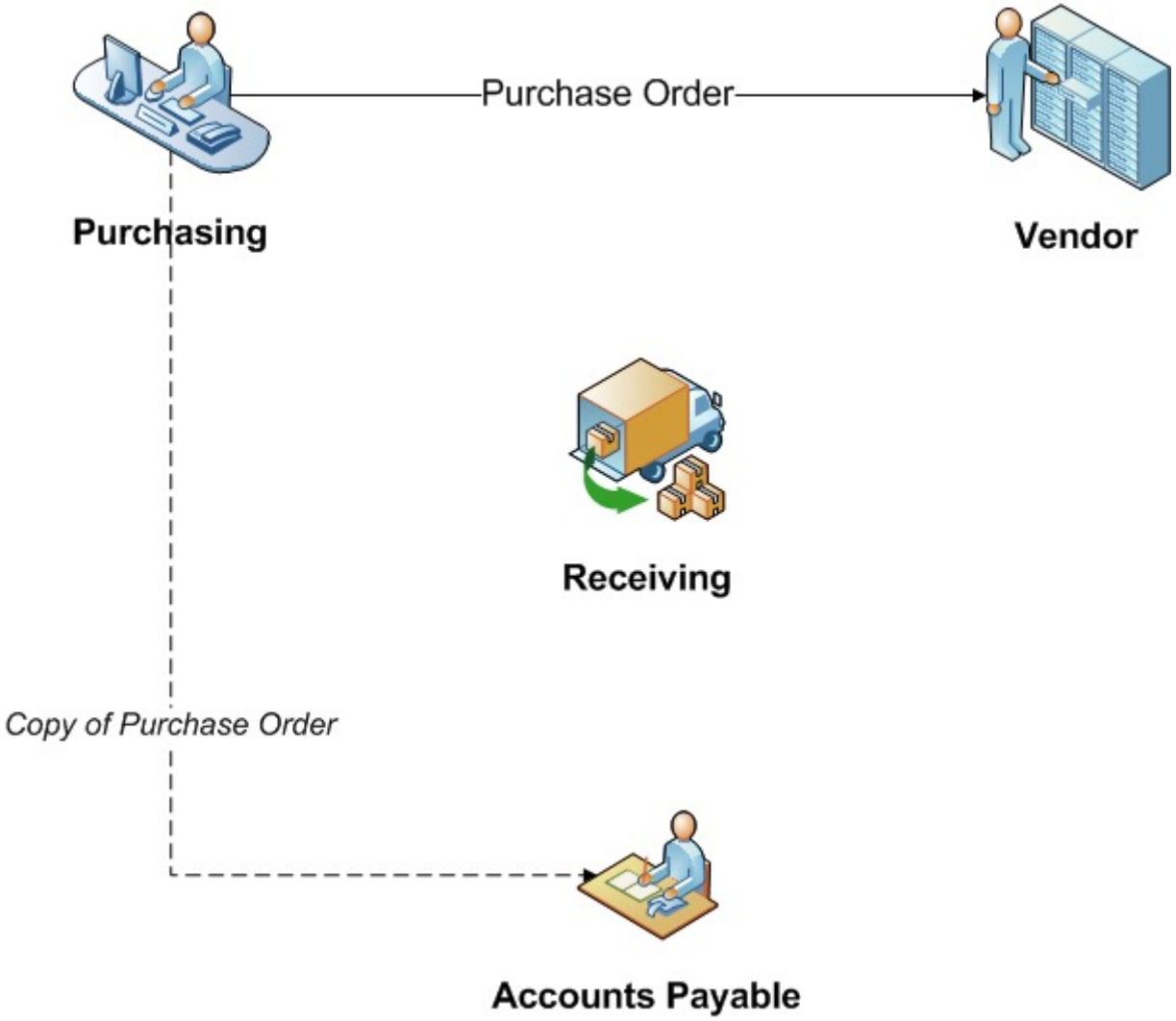
Mazda's accounts payable team was about 5 people, versus a department of over 500 in Ford.

Even after considering differences of size, this was **5 times smaller** than Ford.

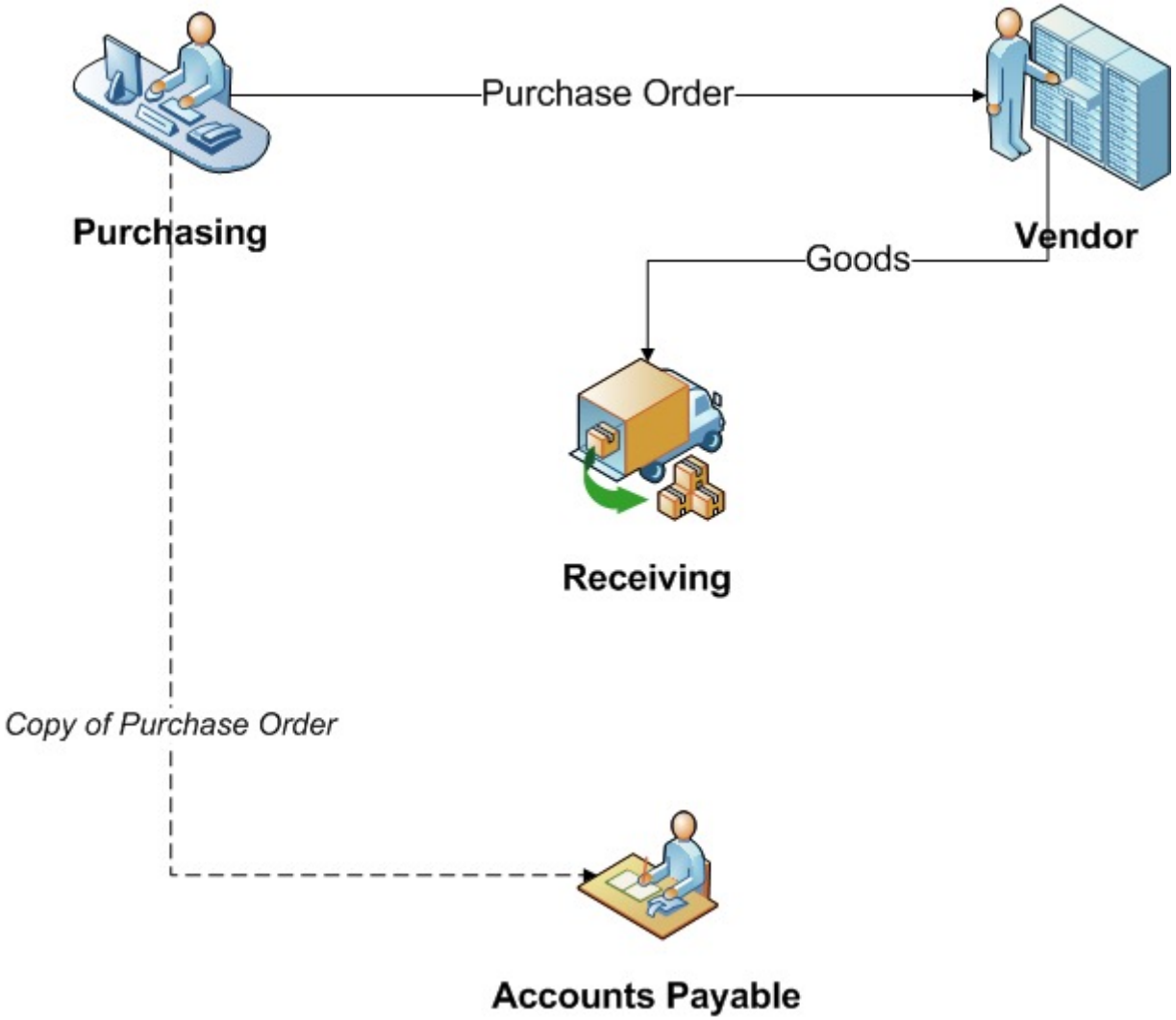
How the process worked? ("as is")



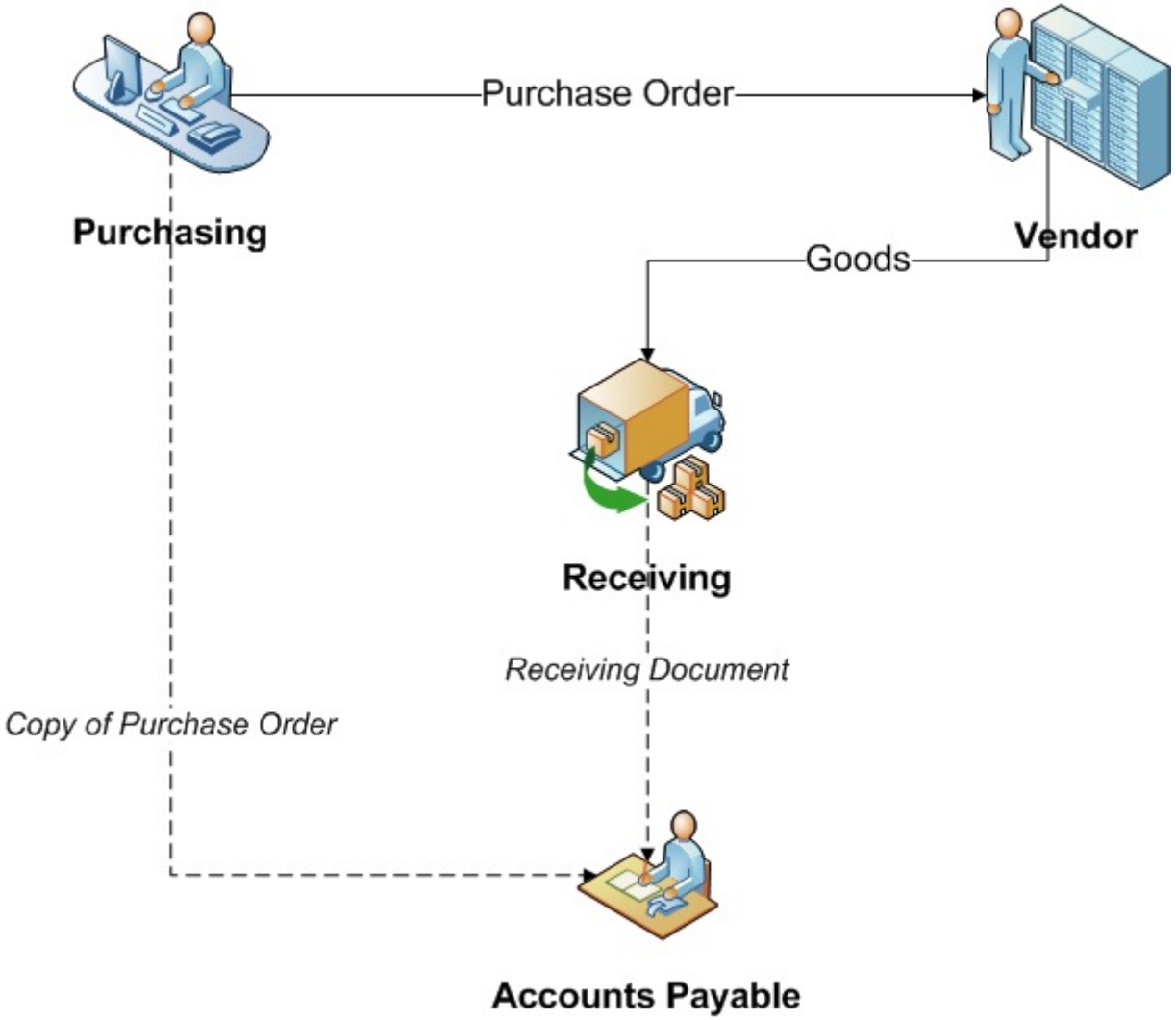
How the process worked? ("as is")



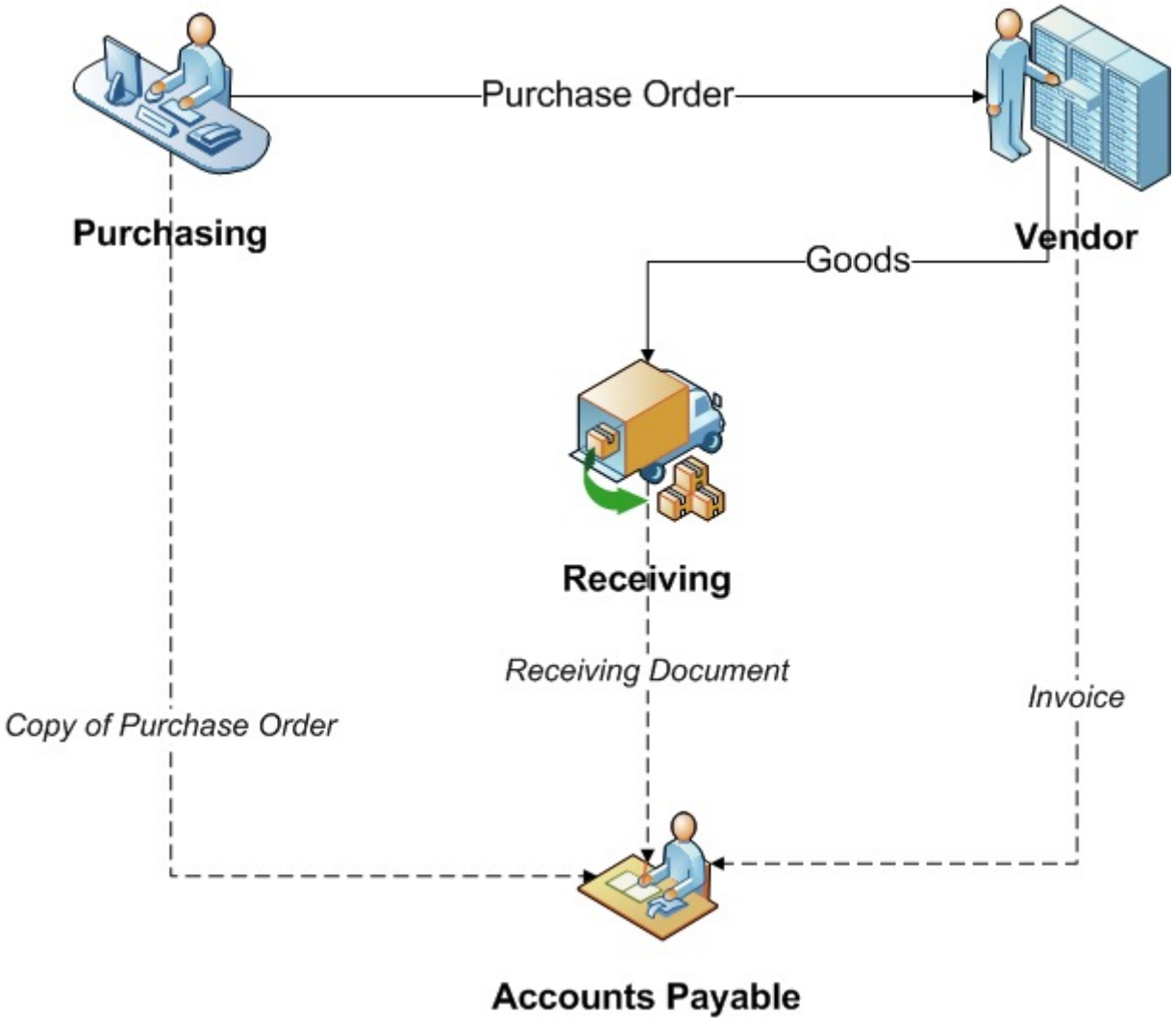
How the process worked? ("as is")



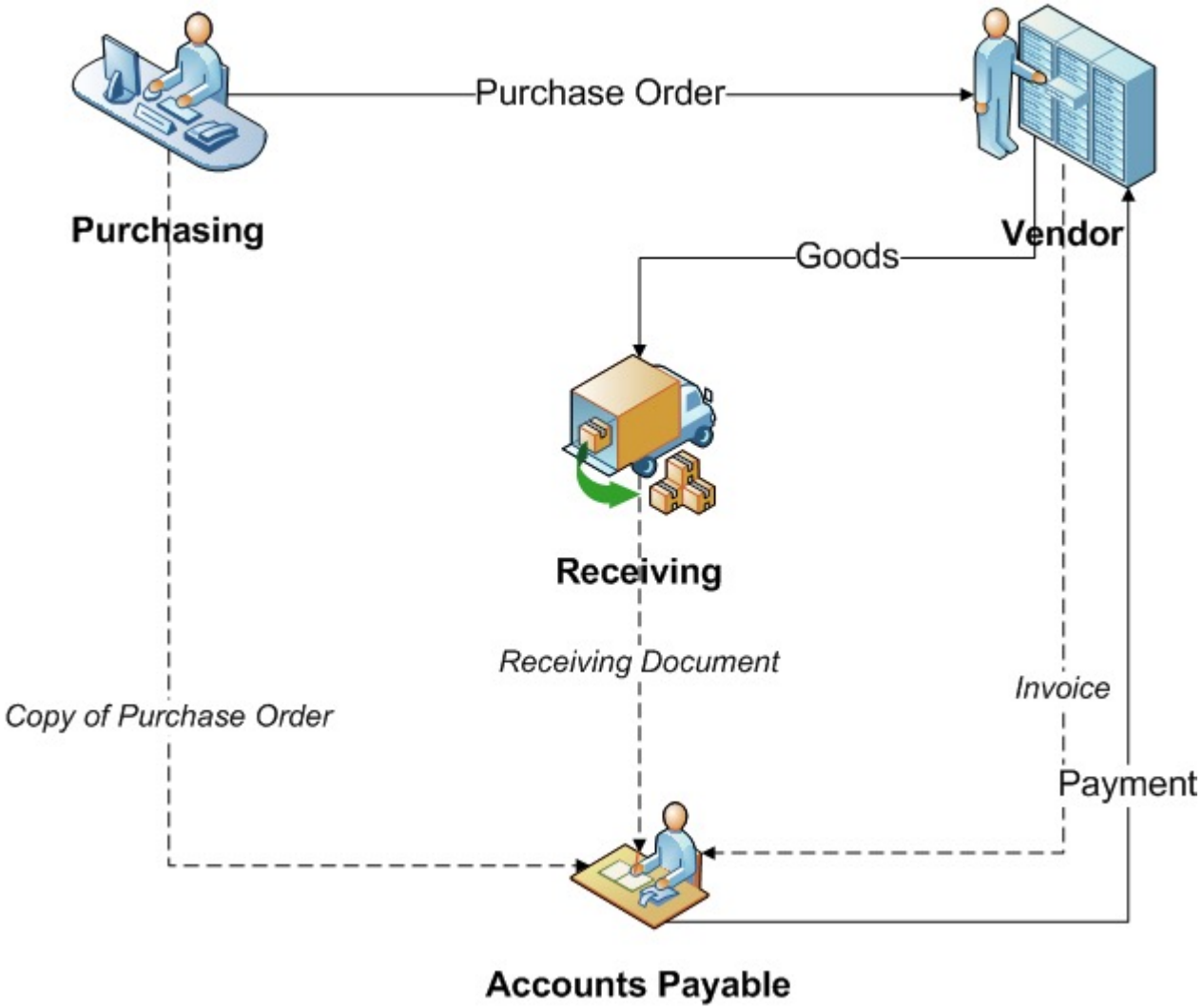
How the process worked? ("as is")



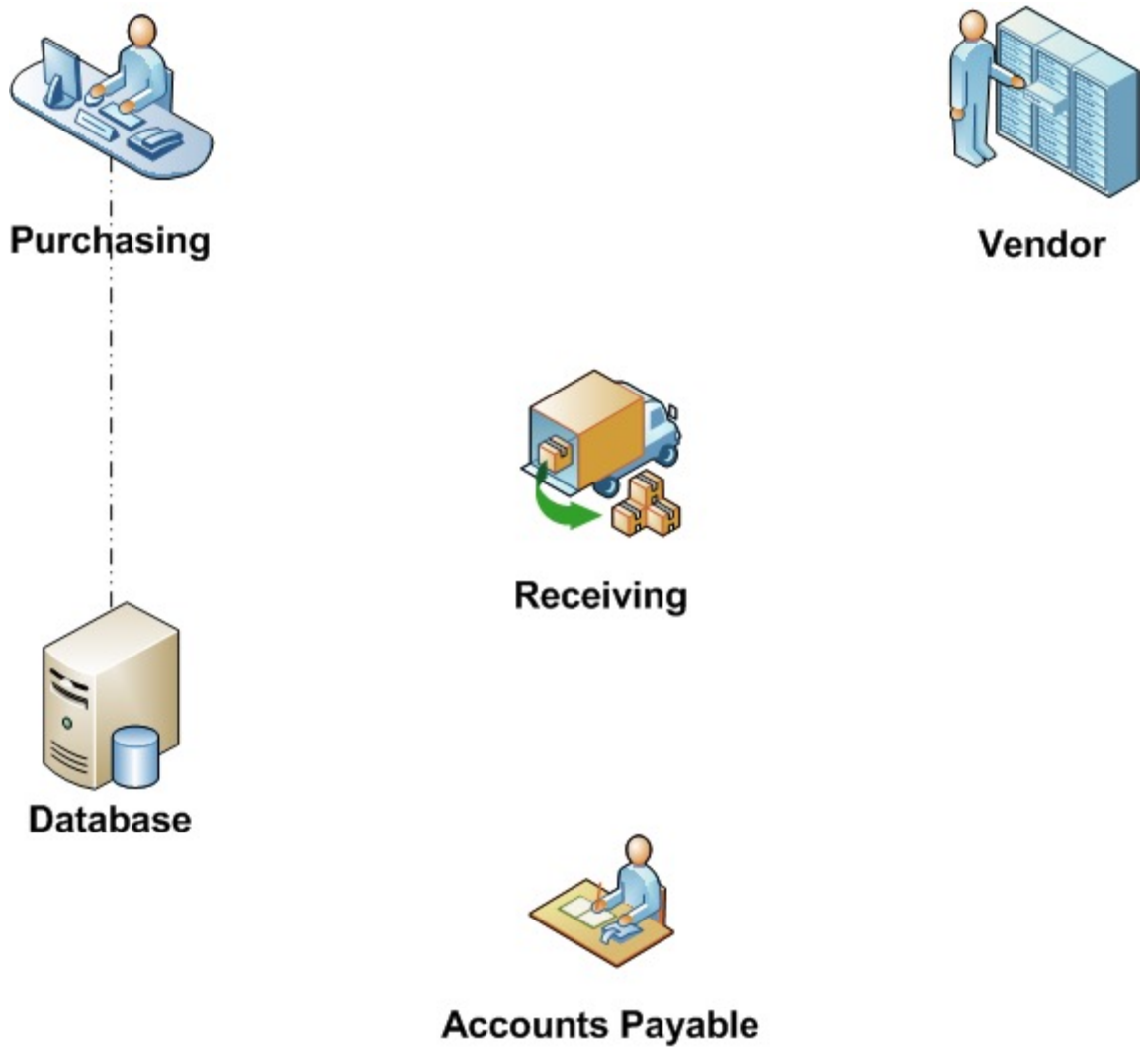
How the process worked? ("as is")



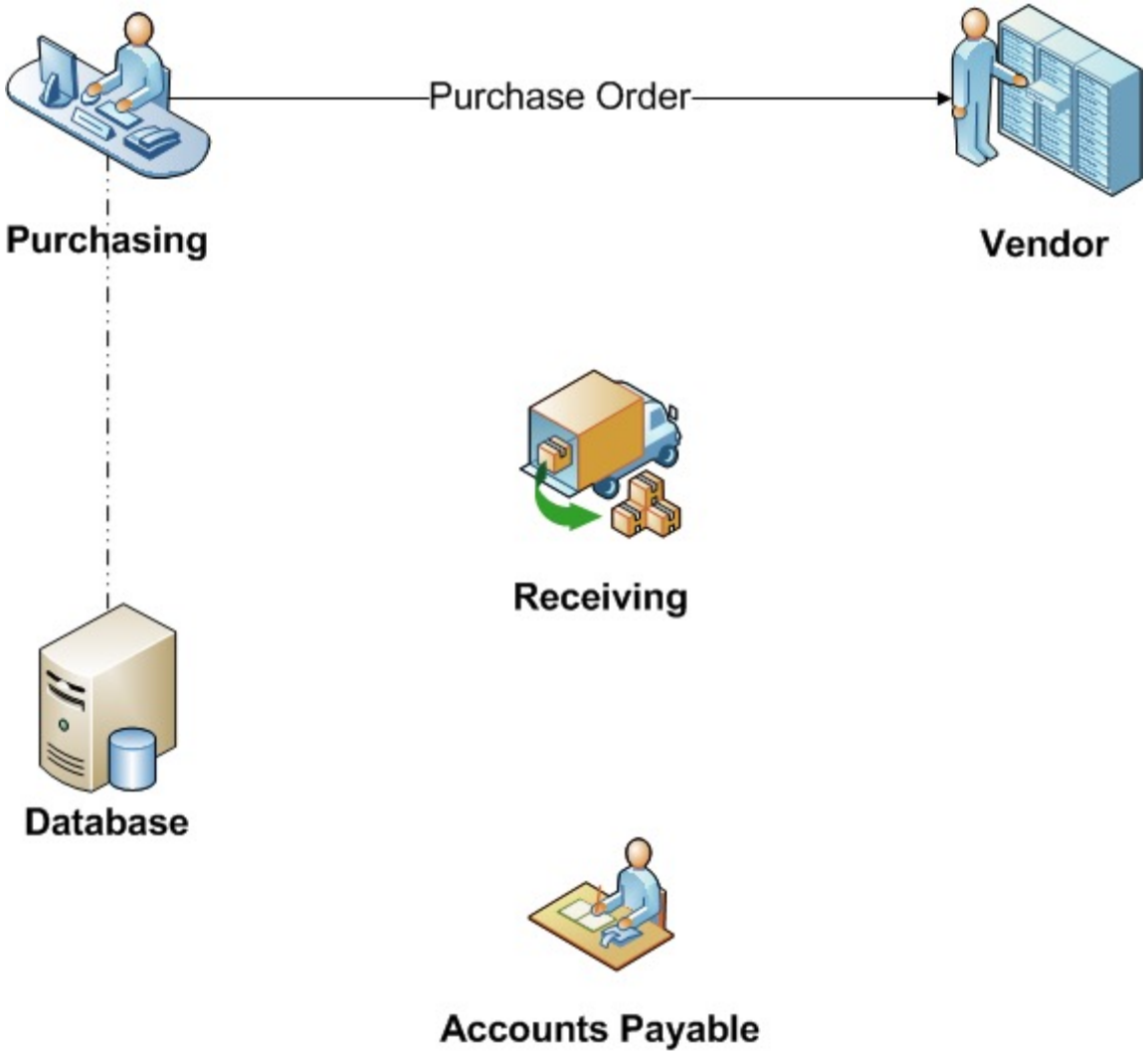
How the process worked? ("as is")



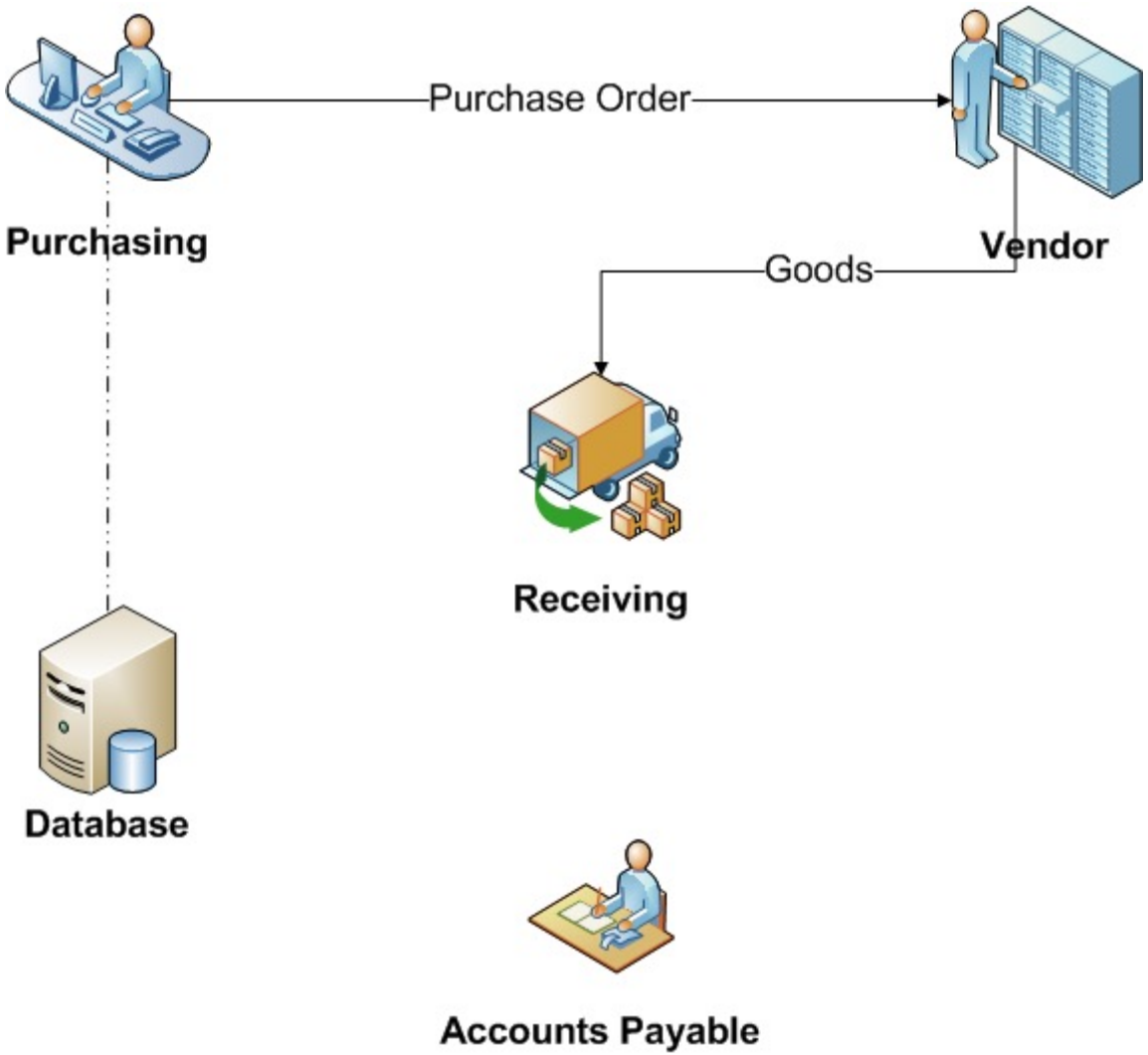
Reengineered Process ("to be")



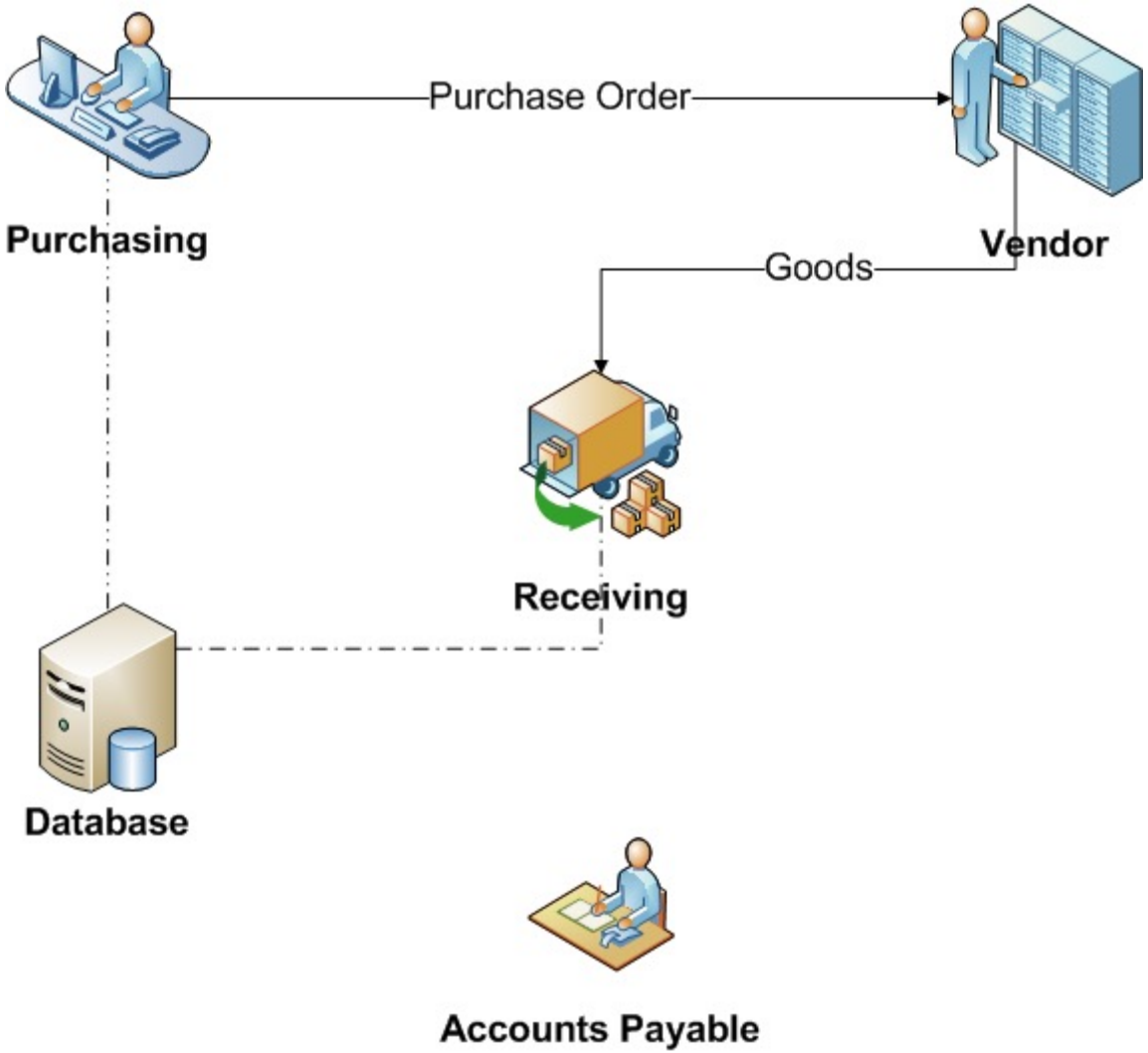
Reengineered Process ("to be")



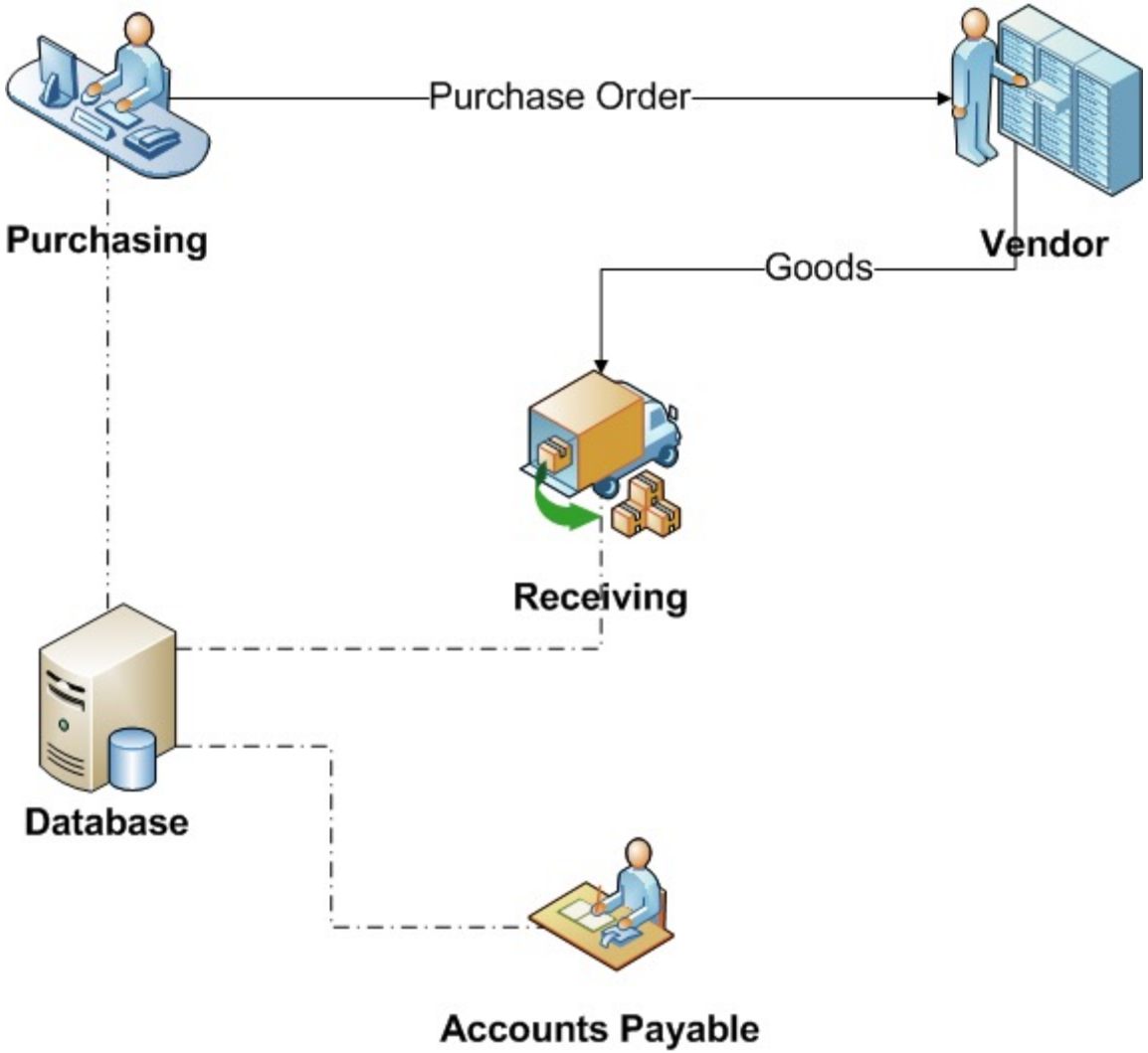
Reengineered Process ("to be")



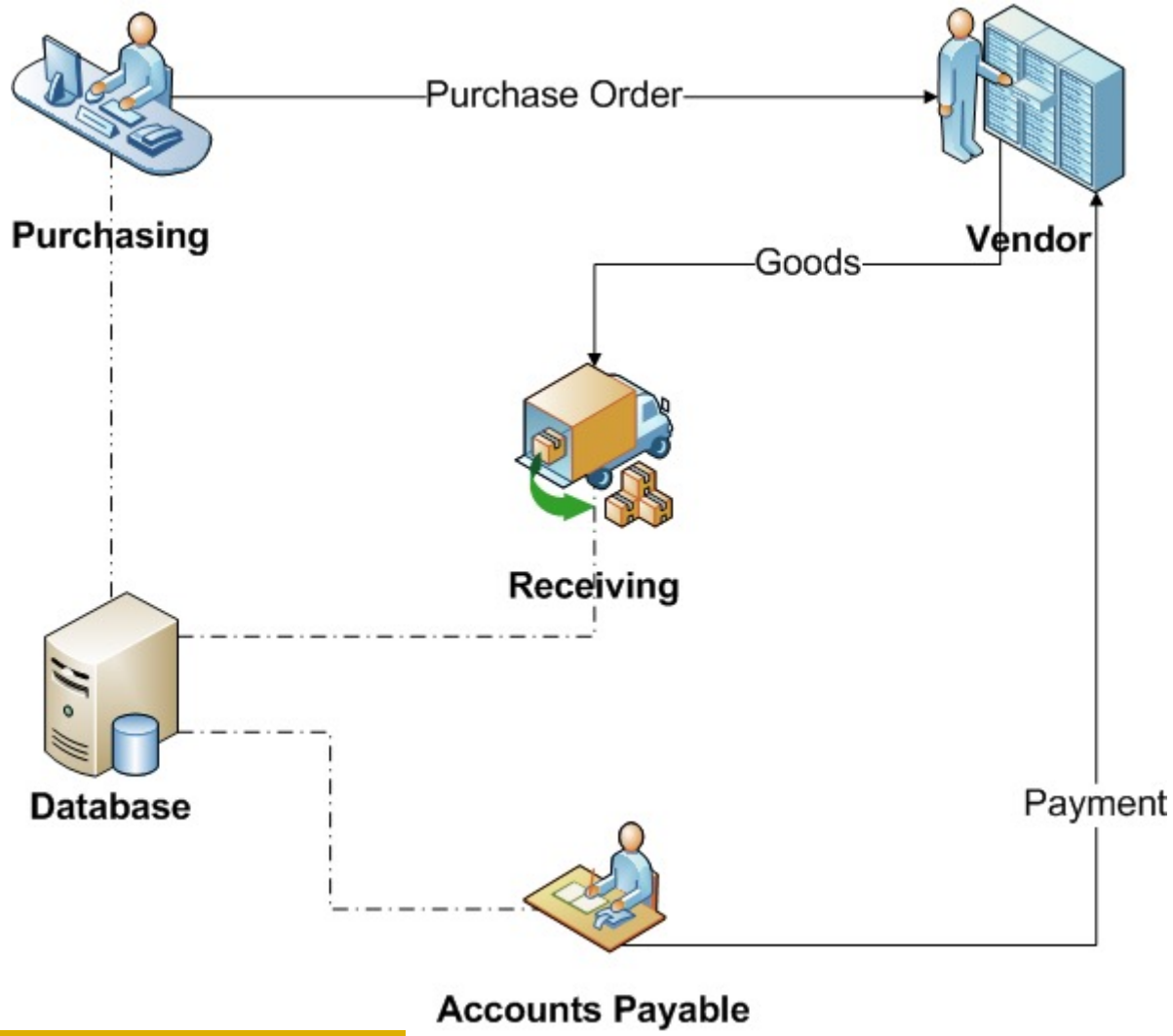
Reengineered Process ("to be")



Reengineered Process ("to be")



Reengineered Process ("to be")



Evaluated Receipts Settlement

Outcome…

- 75% reduction in head count
- Simpler material control
- More accurate financial information
- Faster purchase requisition
- Less overdue payments

Lessons:

- Why automate something we don't need to do at all?
- Automate things that need to be done.
- **Digitization** vs. **Digitalization**

"Don't Automate, Obliterate!" (Hammer, 1990)

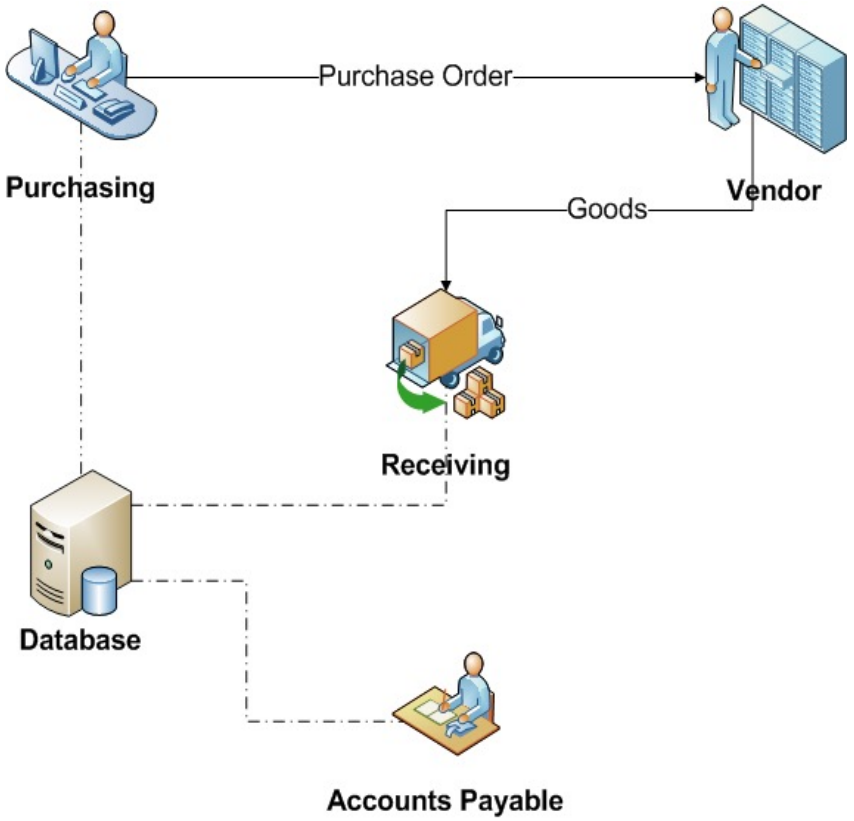
Some principles of BPR

1. Capture information once and at the source
2. Subsume information-processing work into the real work that produces the information
3. Have those who use the output of the process drive the process
4. Put the decision point where the work is performed, empower workers to decide, and build control into the process
5. Treat geographically dispersed resources as though they were centralized.

Principle 1

Capture information once and at the source

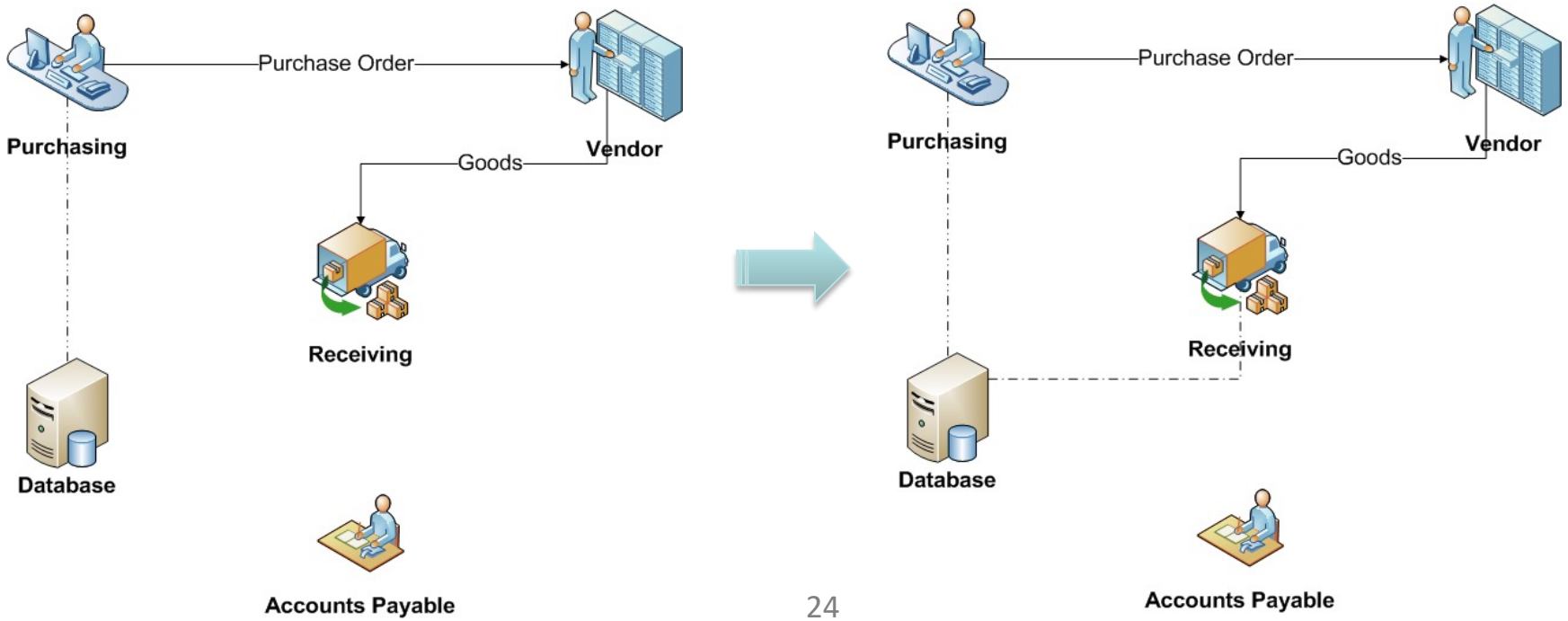
- Shared data store
  - All process workers access the same data
  - Don't send around data, share it!
- Self-service
  - Customers capture data themselves
  - Customers perform tasks themselves (e.g. collect documents)



Principle 2

Subsume information-processing work into the real work

- Evaluated receipt settlement: when receiving the products, record the fulfillment of the PO, which triggers payment



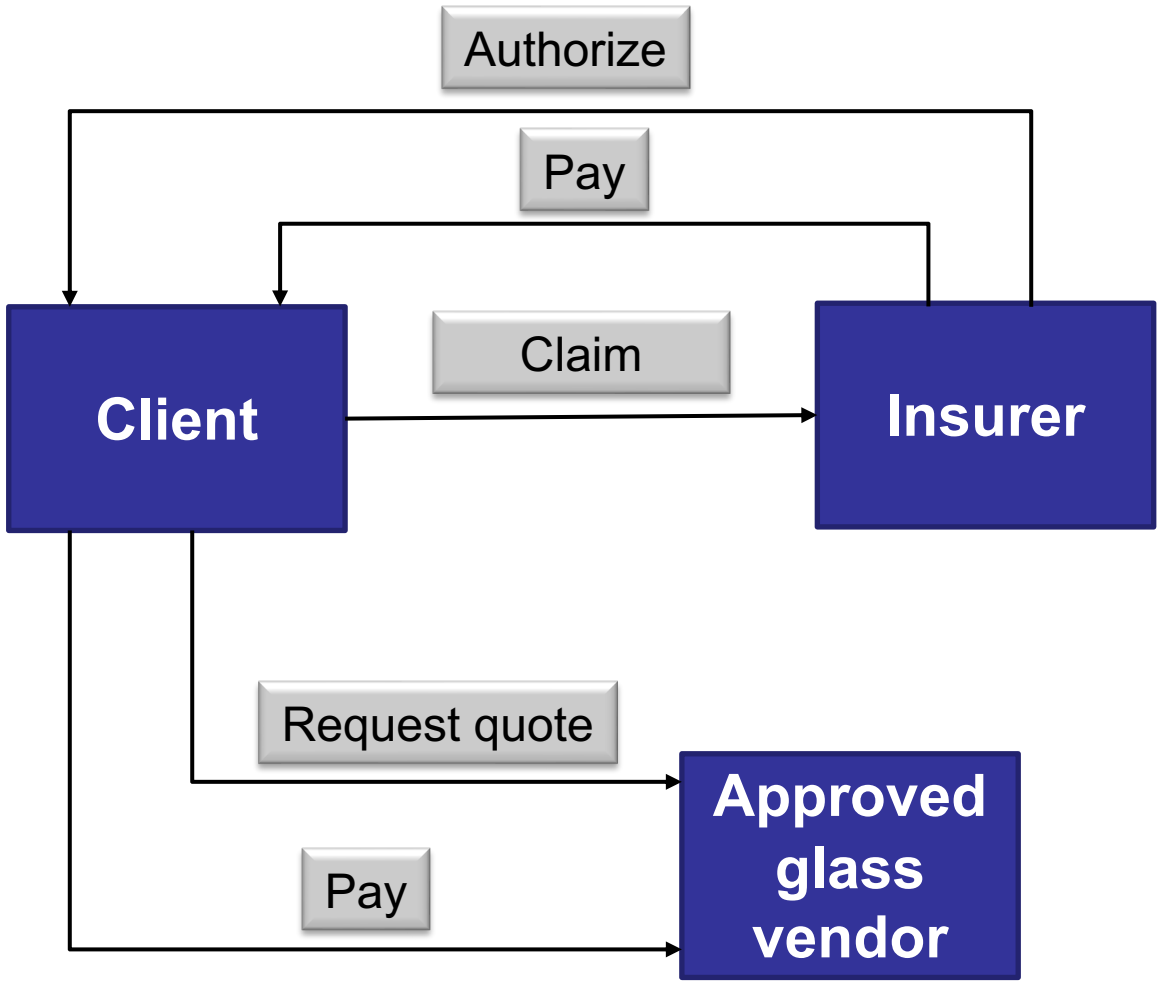
Principle 3

Have those who use the output of the process drive the process

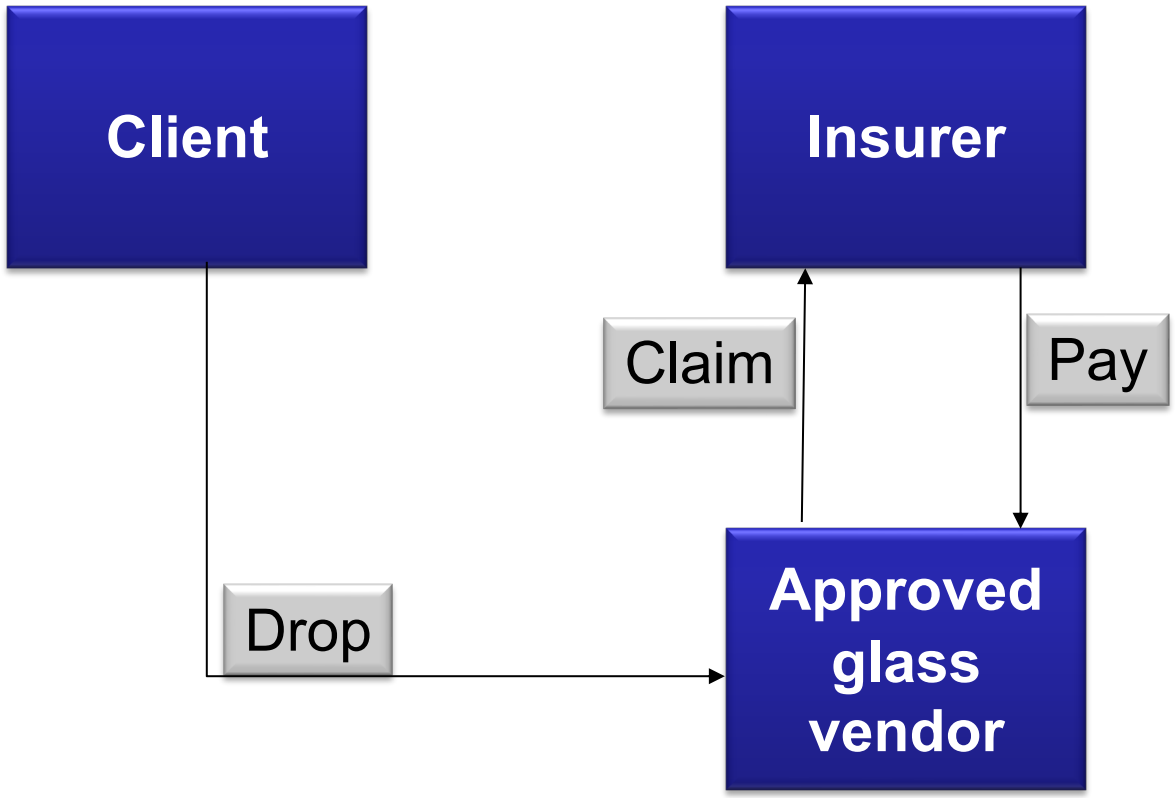
- Vendor-managed inventory
- Scan-based trading

Push work to the actor that has the incentive to do it

Example: problematic claims process



Redesigned claims process



Principle 4

Put the decision point where the work is performed, empower workers to decide, and build control into the process

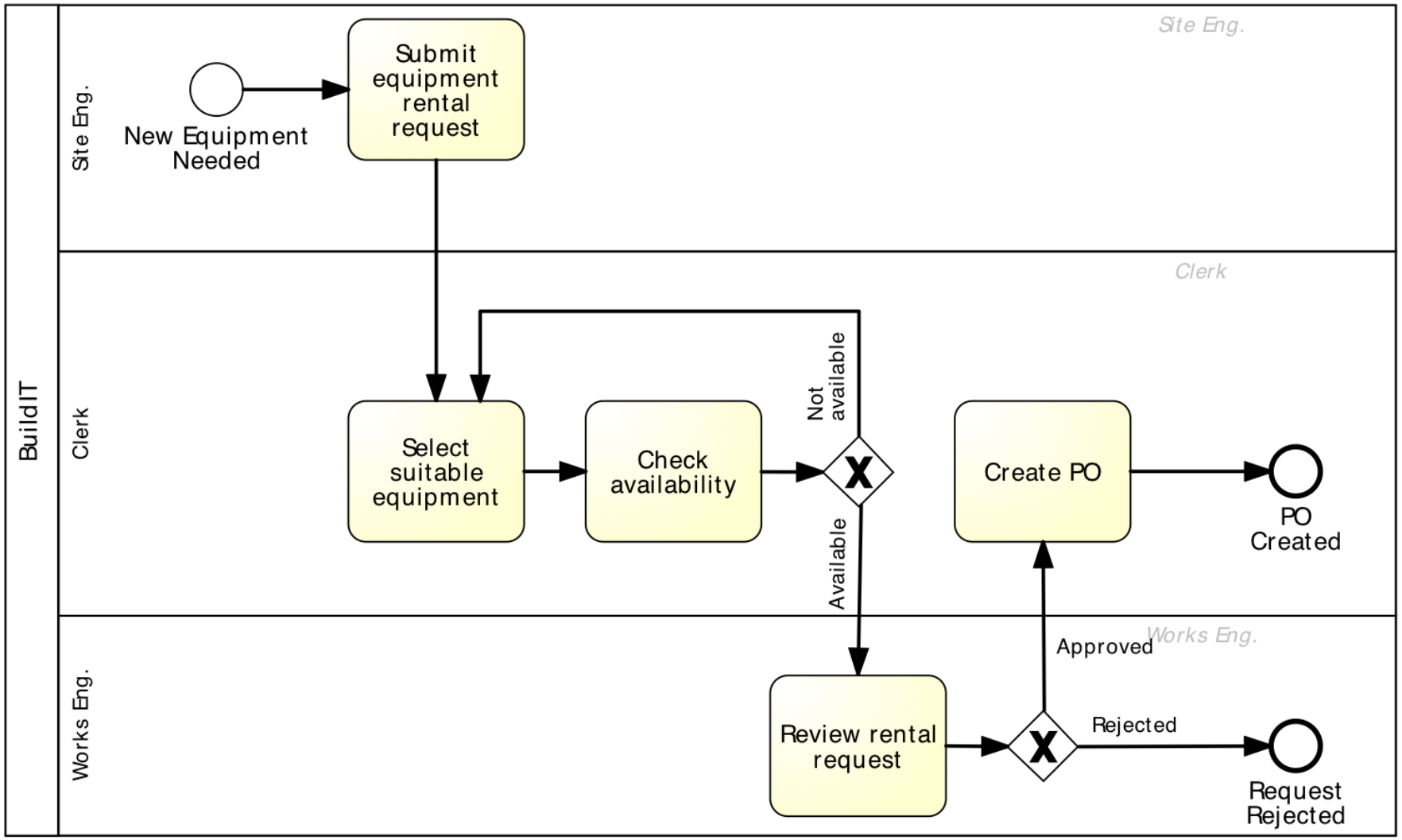
- Empower the process workers
- Provide process workers with information needed to make decisions themselves
- Replace back-and-forth handovers between workers and managers (transportation waste) with well-designed controls

Principle 5

Treat geographically dispersed resources as though they were centralized.

- If same people perform the same function in different locations, integrate and share their work wherever possible
- Larger resource pools → less waiting times even with relatively high resource utilization

Example: Equipment rental process



Example: Self-service-based redesign

Principles 1 & 2

- When equipment is needed, site engineer queries the suppliers' catalogue, selects equipment and triggers PO

Principle 3

- Supplier stocks frequently used equipment at construction site, site engineers scan to put them into use

Principle 4

- Site engineer is empowered with the authority to rent the equipment; works engineer performs statistical controls

Process redesign approaches

Transformational Redesign

- Puts into question the fundamental assumptions and principles of the existing process structure
- Aims to achieve breakthrough innovation
- Example: Business Process Reengineering (BPR)

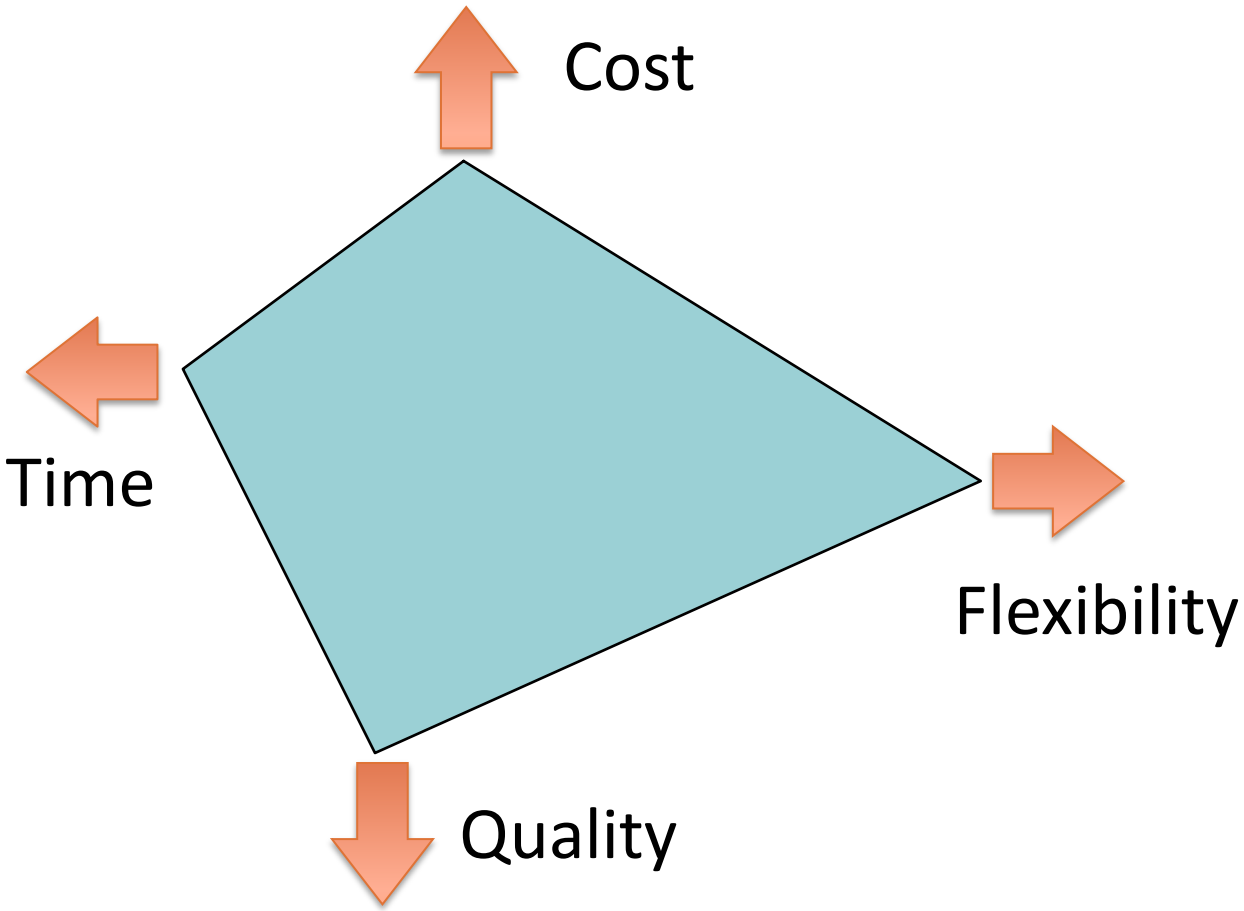
Transactional Redesign

- Doesn't put into question the current process structure
- Seeks to identify problems and resolve them <u>incrementally</u>, one step at a time
- **Example: Heuristic redesign**

Heuristic process redesign

- A method to identify changes to an “as is” process based on a collection of heuristics that strike tradeoffs between:
  - Cost
  - Time
  - Quality
  - Flexibility

Performance measures: the Devil's Quadrangle



Flexibility

- Ability to react to changes in:
  - Workload
  - Customer demands and expectations
  - Resource and business partner availability and performance
- Example: Following natural disasters (e.g. storms), the number of home insurance claims increases by tenfold
- To address this surge, flexibility is required at:
  - Resource level: Staff redeployment, faster performance
  - Process level: Performing tasks differently to speed up the front-end
  - Management: Relaxing business rules and controls where possible

Redesign heuristics

Task-level

- Task elimination
- Task composition/decomposition
- Triage

Flow-level

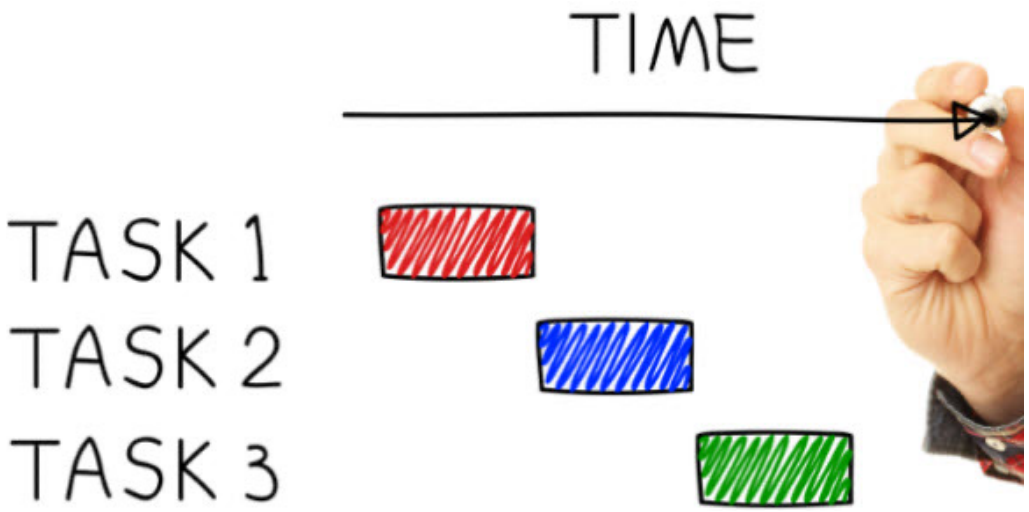
- Re-sequencing
- Parallelism enhancement

Process-level

- Specialization & standardization
- Resource optimization
- Communication optimization
- Automation

Task-level redesign heuristics

1. Task elimination
2. Task composition/decomposition
3. Triage



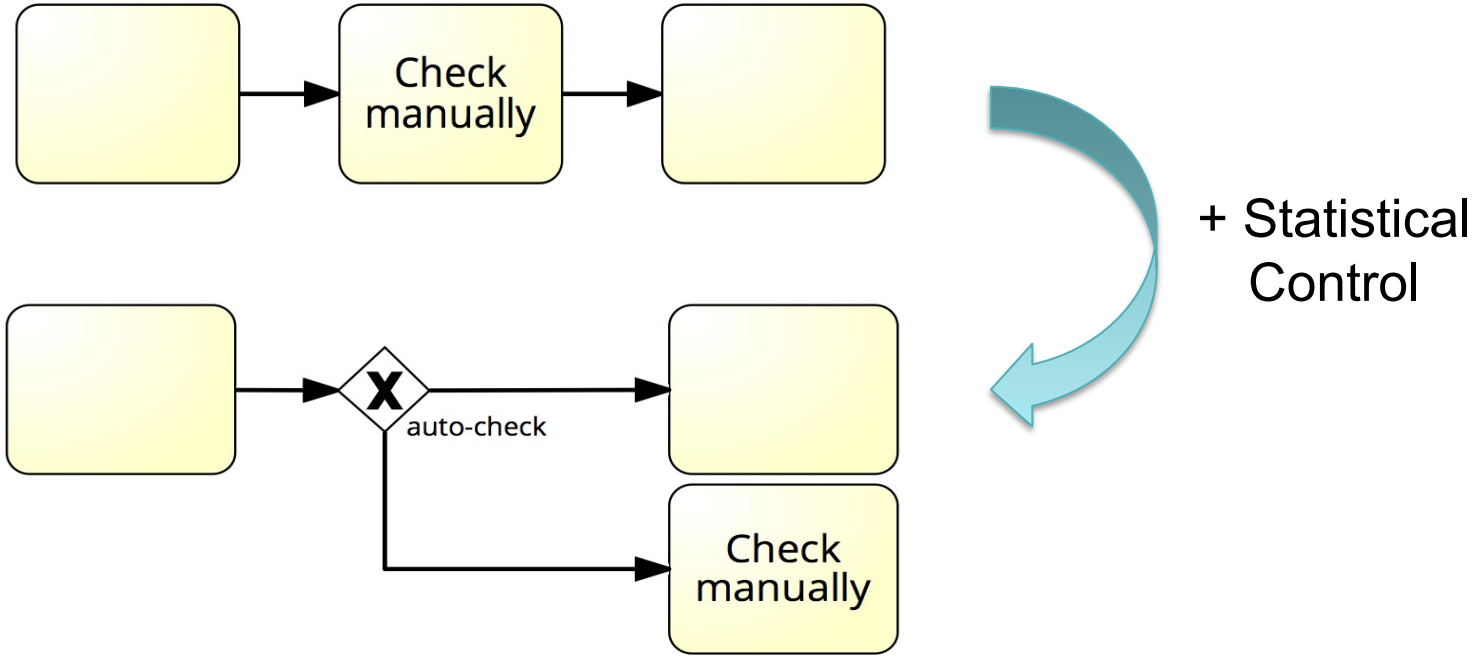
H1. Task elimination

Eliminate non-value-adding steps wherever these can be isolated

- Forward, send, receive, …

Consider reducing manual control steps (checks & approvals) by:

- Skipping them where feasible
- Replacing them with statistical controls
- Skipping them selectively



H1. Task elimination

(T+, C+/-, Q-)

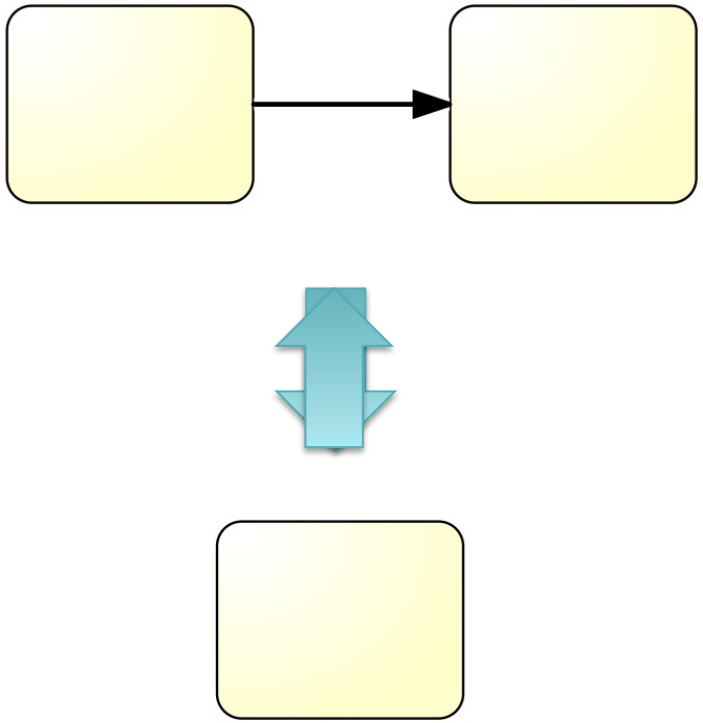
Consider trade-off between the cost of the check and the cost of not doing it

Examples:

- Procure-to-pay process: some types of employees are empowered to trigger isolated purchases below $500 without supervisor approval
- Order-to-cash process: invoices from trusted suppliers under $1000 are not checked on a one-by-one basis
- University admission process: authenticity check is very expensive, yet it leads to only 1% of applications being rejected

H2. Task composition/decomposition

- Consider composing two tasks to eliminate transportation and reduce “context switches”, OR
- Consider splitting a task into two and assign to separate, specialized resources



H2. Task composition and decomposition

Composition example:

- Procure-to-pay process: Merging two checks: “Check necessity of purchase” and “Check budget”

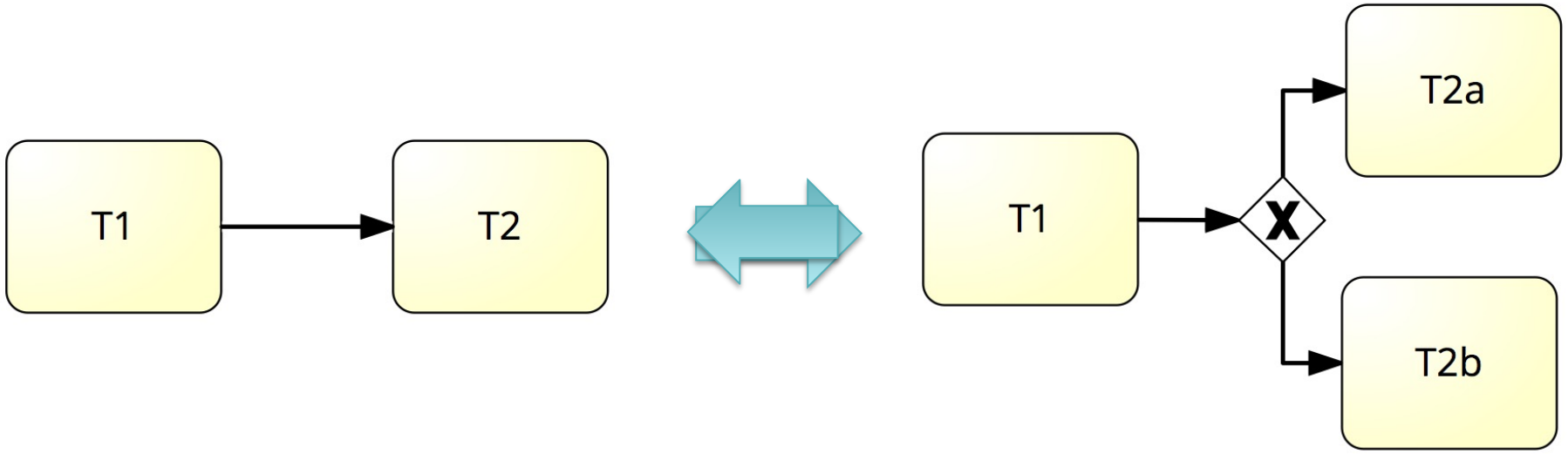
Decomposition example:

- Make-to-order process: Separate a single thick “prepare quote” task into “prepare bill of materials”, “prepare production plan” and “estimate costs and delivery time”

Composition: (T+, C+/-, F+)
Decomposition: (T-, C+, F-)

H3. Triage

- Specialize a task: divide a *general* task into two or more alternative tasks
- Generalize tasks: integrate two or more alternative tasks into one general task



H3. Triage

Specialization example:

- Procure-to-pay process: Separate approvals of *small* purchases, *medium* purchases and *large* purchases

Generalization example:

- Make-to-order process: Integrate quote preparation for two product lines into one single task

Specialization: (T+, C+/-, F-)
Generalization: (T-, C+/-, F+)

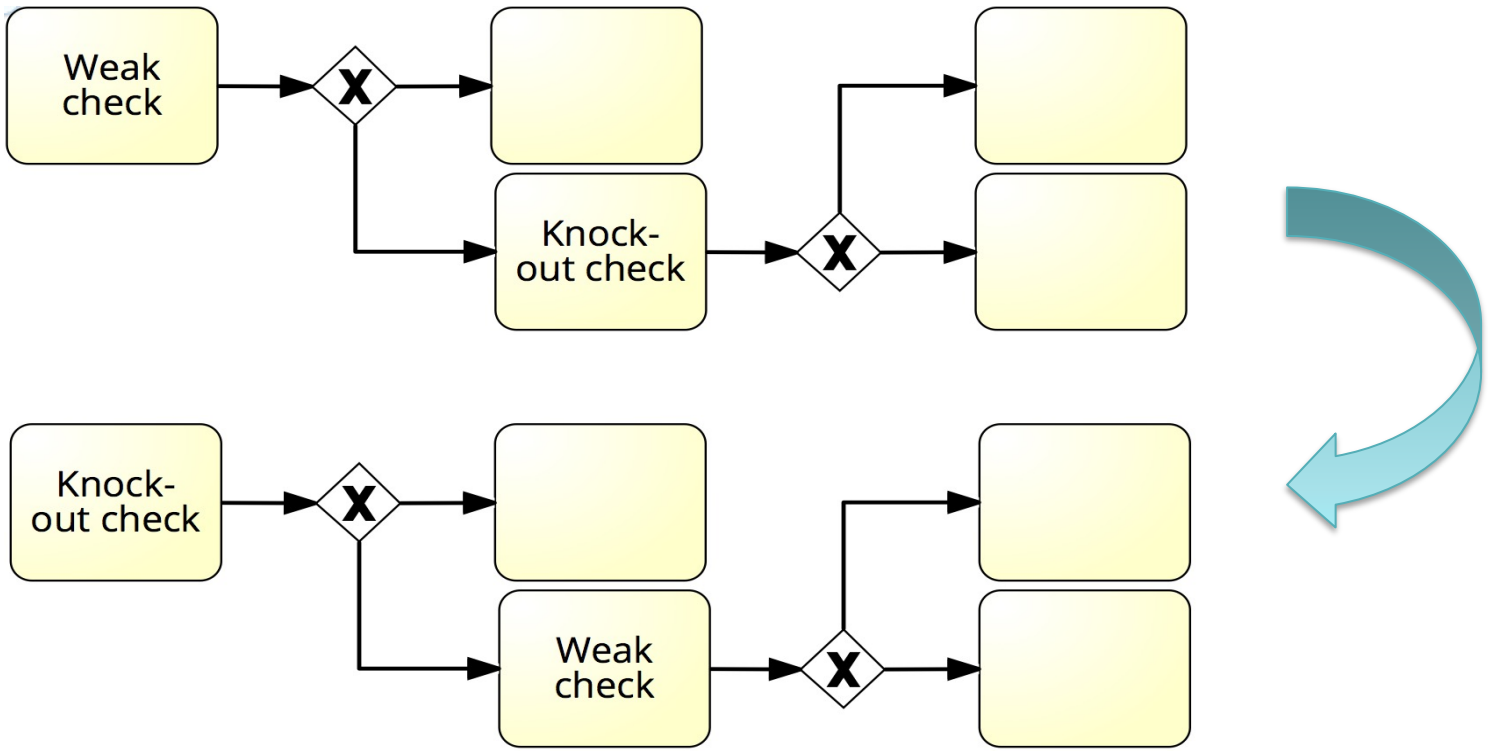
Flow-level redesign heuristics

4. Re-sequencing
5. Parallelism enhancement

H4. Re-sequencing

Re-order tasks according to their cost/effect ratio to minimize over-processing

- Postpone expensive tasks that may end up not being necessary until the end
- Put *knock-out* checks first in order to identify problems early



H4. Re-sequencing

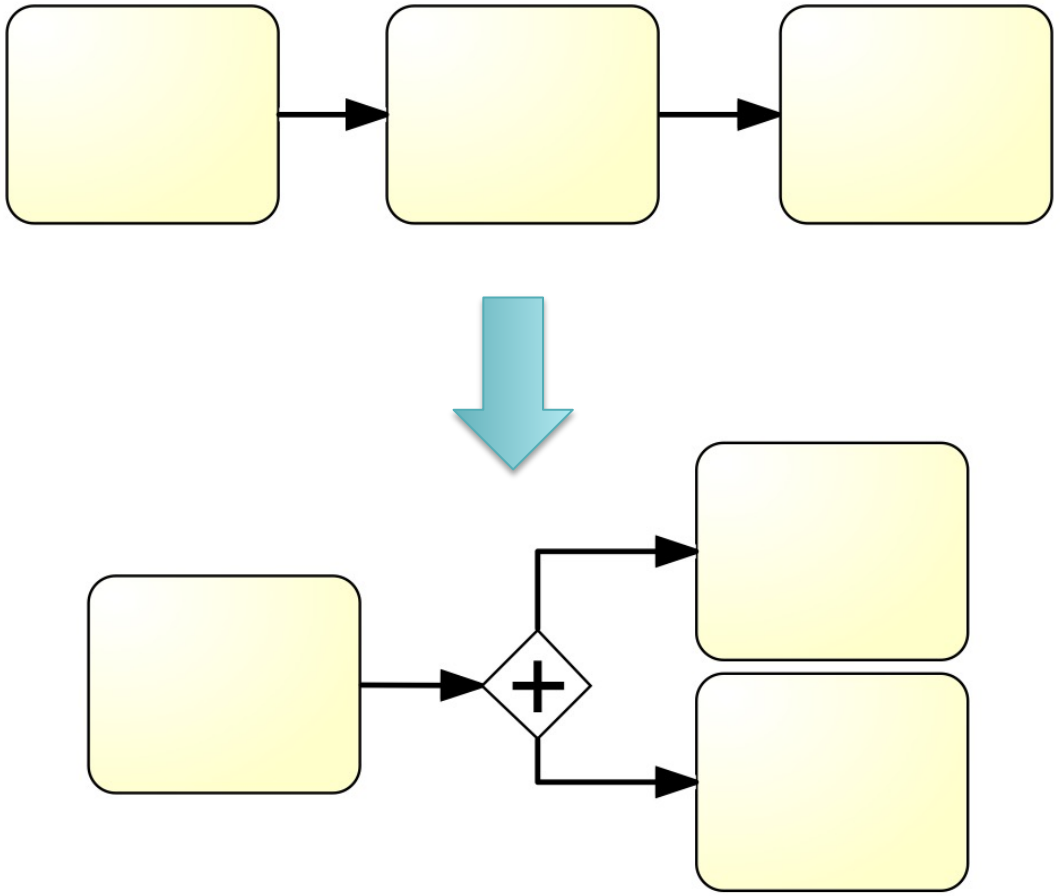
(T+,C+)

Examples:

- Make-to-order process: If “Prepare production plan” is time-consuming, postpone it until after the quote price has been tentatively accepted by the customer
- Procure-to-pay process: If “Check necessity of purchase” leads to 20% of knock-outs and “Check budget” leads to 2%, perform “Check necessity of purchase” first
- University admission process: authenticity check (very slow) leads to 1% of applications being rejected while committee’s check leads to 80% of applications being rejected. Put committee’s check first

H5. Parallelism enhancement

Parallelize tasks where possible in order to reduce cycle time

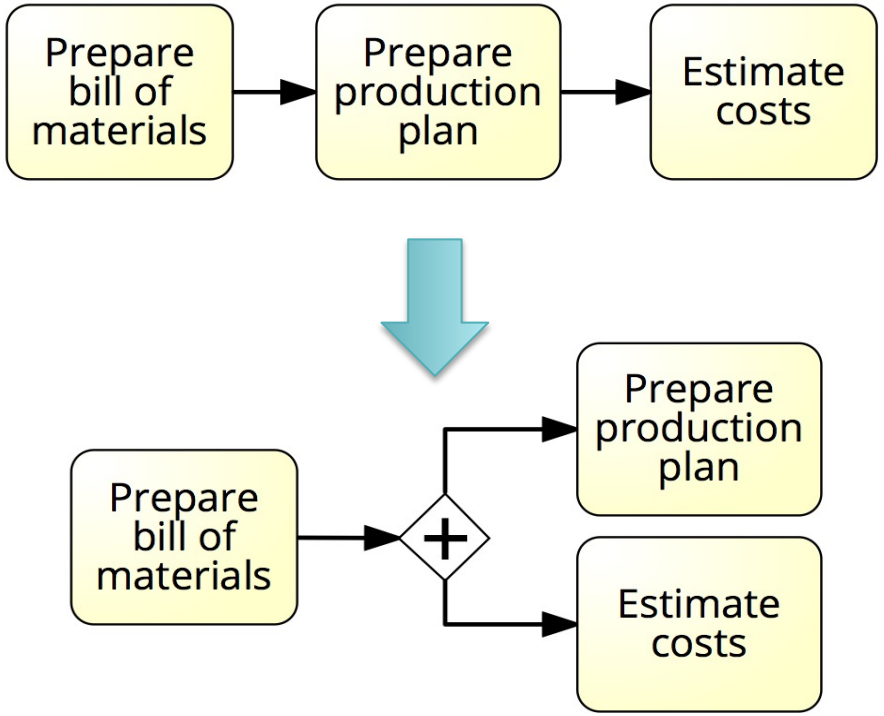


H5. Parallelism enhancement

(T+)

Examples:

- Procure-to-pay process: Parallelize “Approve budget” and “Approve necessity of purchase”
- Make-to-order process: After “Prepare bill of materials”, perform “Prepare production plan” and “Estimate costs” in parallel



Process-level redesign heuristics

6. Process specialization & standardization
7. Resource optimization
8. Communication optimization
9. Automation

H6. Process specialization/standardization

Process specialization

- One process is split into multiple ones: by customer class, by geographic location, by time period (winter, summer), etc.
- Resources are split accordingly

Process standardization

- Two processes are integrated
- Resources are pooled together

H6. Process specialization & standardization

Specialization example:

- Procure-to-pay process: One process for Direct procurement (e.g. raw materials) and one for Indirect procurement (MRO - Maintenance, Repair and Operations)
- Claims handling process: One claims handling process for the summer season (stormy season - peak) and one for the winter season (off-peak)

Standardization example:

- Claims handling process: Integrate claims handling for motor insurance across different brands of a group

Specialization: (C+/-, Q+/-, F-)
Standardization: (C+, Q+/-, F+)

H7. Resource optimization

Use resources of a given type as if they were in one room

- Avoid one group of people overloaded and another (similar) group idle

Let people do work that they are good at

- However, avoid inflexibility as a result of specialization

When allocating work to resources, consider the flexibility in the near future

- Allocate work to specialized resources first

Avoid setups as much as possible

- *Chain* multiple instances of the same task [sequential]
- *Batch* multiple instances of the same task [parallel]

H7. Resource optimization

Resource integration example:

- Claims handling process: Share resources across different types of claims (e.g. motor and personal insurance)

Batching example:

- Claims handling process: Batch all claims for a given geographic area and assign them to the same resources
- University admission process: Batch all applications and handle them to the assessment committee

(T+, C+, F+/-)

H8. Communication optimization

Automate handling, recording and organization of messages

Monitor customer interactions, record exceptions

Optimize

1. Number of interactions with customers and business partners
2. Type of interaction (synchronous vs. asynchronous)
3. Timing of interactions

(T+,Q+,C+/-,F-)

H8. Communication optimization

1. Optimize number of interactions
   - Gather sufficient information to get to the next milestone (reduce external interactions)

2. Optimize type of interaction
   - *Synchronous* interactions effective to resolve minor defects
   - *Asynchronous* to notify, inform, resolve major defects, request additional information to reach next milestone

H8. Communication optimization

3. Optimize timing of interactions:

- *Front-loaded process*: bulk of information exchange and processing happens upfront
  - Complete-kit concept
- *Back-loaded process*: bulk of information exchange and processing happens downstream
  - Example: Ford procurement process in the late 80s

H8. Communication optimization

Complete-Kit Concept: *"Work should not begin until all pieces necessary to complete the job are available"*

Boaz Ronen

Principles for complete-kit process design:

- Provide complete and easy-to-follow instructions for those who will initiate the process.
- If a process cannot start, the client should be notified of all defects that could be reasonably identified at the onset of the process
- Consider the tradeoff between "incomplete-kit" process initiation vs. roundtrip to revise and resubmit a request

H9. Automation

Use data sharing (Intranets, packaged enterprise systems) to:

- Increase availability of information to improve visibility and decision-making (subject to security/privacy requirements)
- Avoid duplicate data entry and transportation

Use network technology to:

- Replace physical flow (e.g. paper documents) with information flow
- Enable self-service via e.g. online forms and Web data services

H9. Automation

Use tracking technology to identify and locate materials and resources

- Identification: Bar/QR code, RFID
- Location: GPS, indoor positioning

Use business rules technology to automate information processing tasks (including decisions)

Automate end-to-end processes with a dedicated BPM system or system with process automation functionality

(T+,C+/-, Q+/-, F-)

Acknowledgements

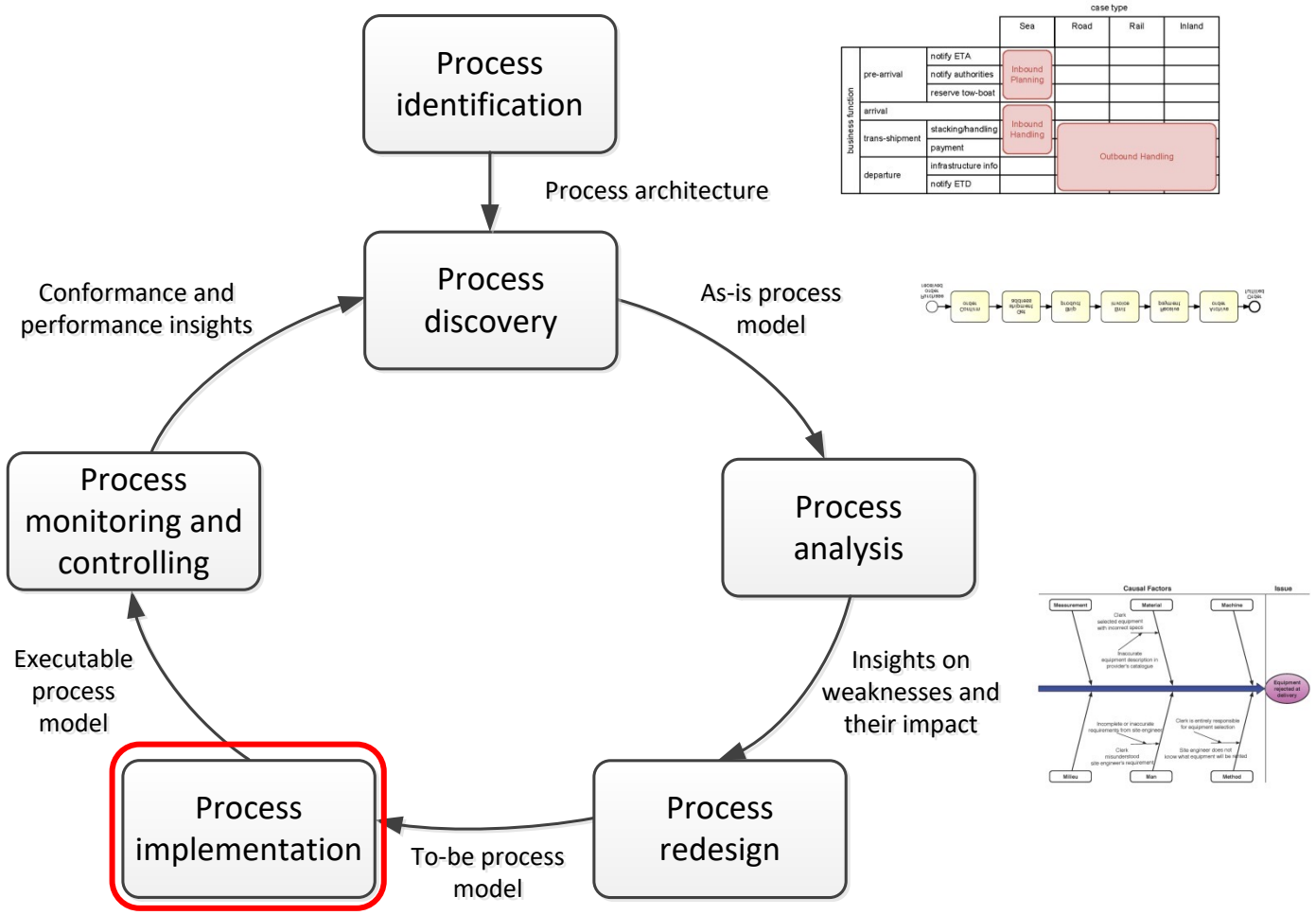
- All material comes from Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A. Reijers, authors of the “Fundamentals of Business Process Management” book.

INFO-H420
Management of Data Science and Business Workflows
Part I
6. Process Automation and Mining

Dimitris SACHARIDIS

2023-2024

BPM Lifecycle



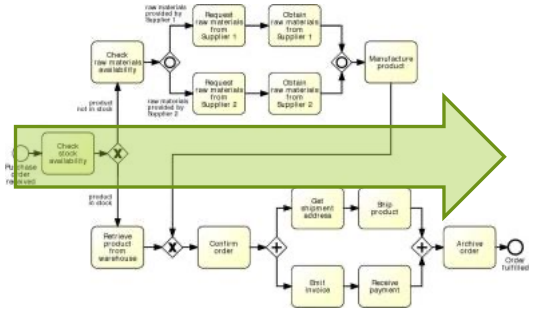
From “conceptual” to executable models

Conceptual “to-be” process models

- are made by domain experts
- provide a basis for communication amongst relevant stakeholders
- must be understandable
- must be intuitive and may leave room for interpretation
- contain purely a relevant set of process information

Executable process models

- are made by IT experts
- provide input to a process enactment system - BPMS
- must be machine readable
- must be unambiguous and should not contain any uncertainties
- contain further details that are only relevant to implementation



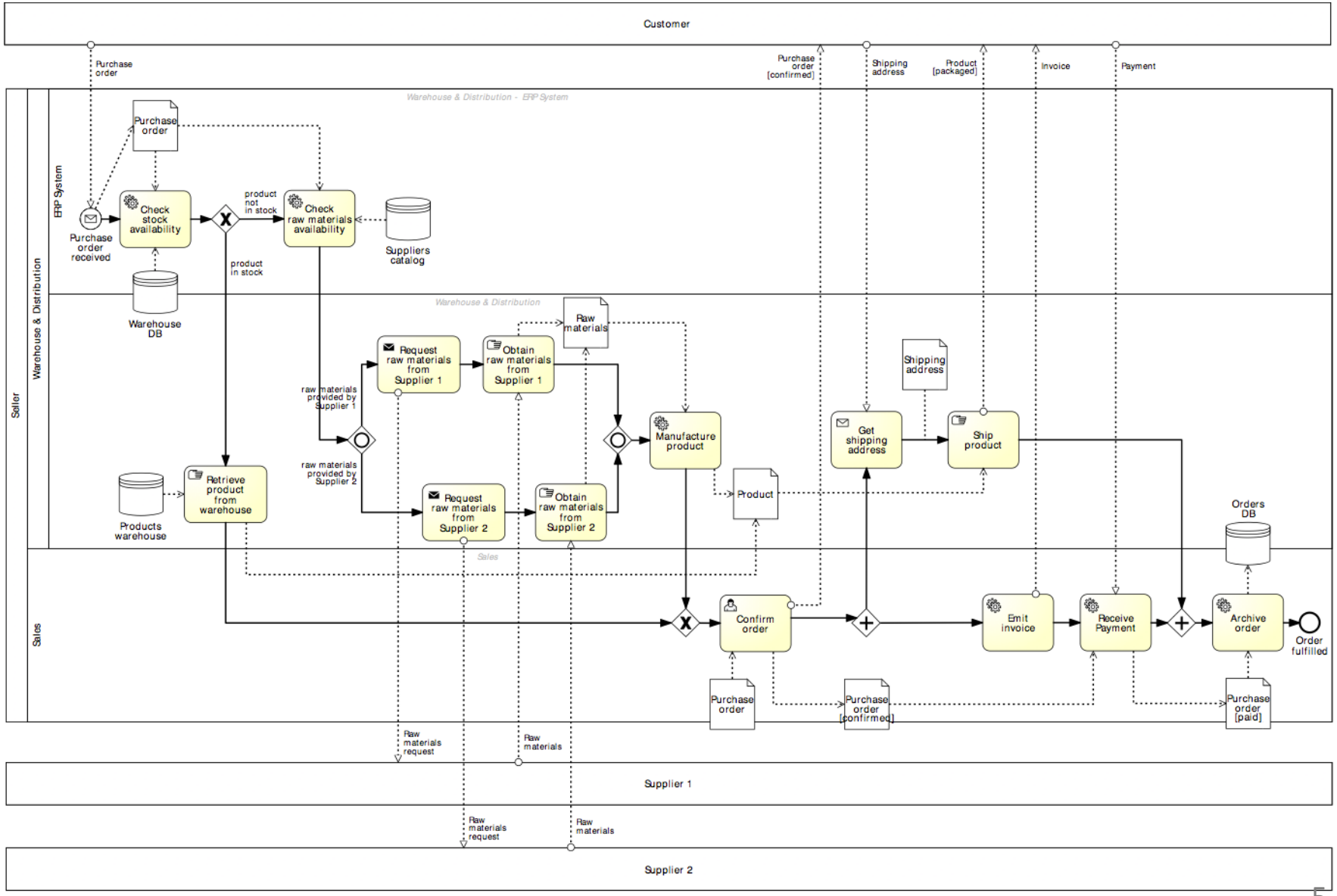
“to-be executed” process model

Bridging the gap: A five-step method

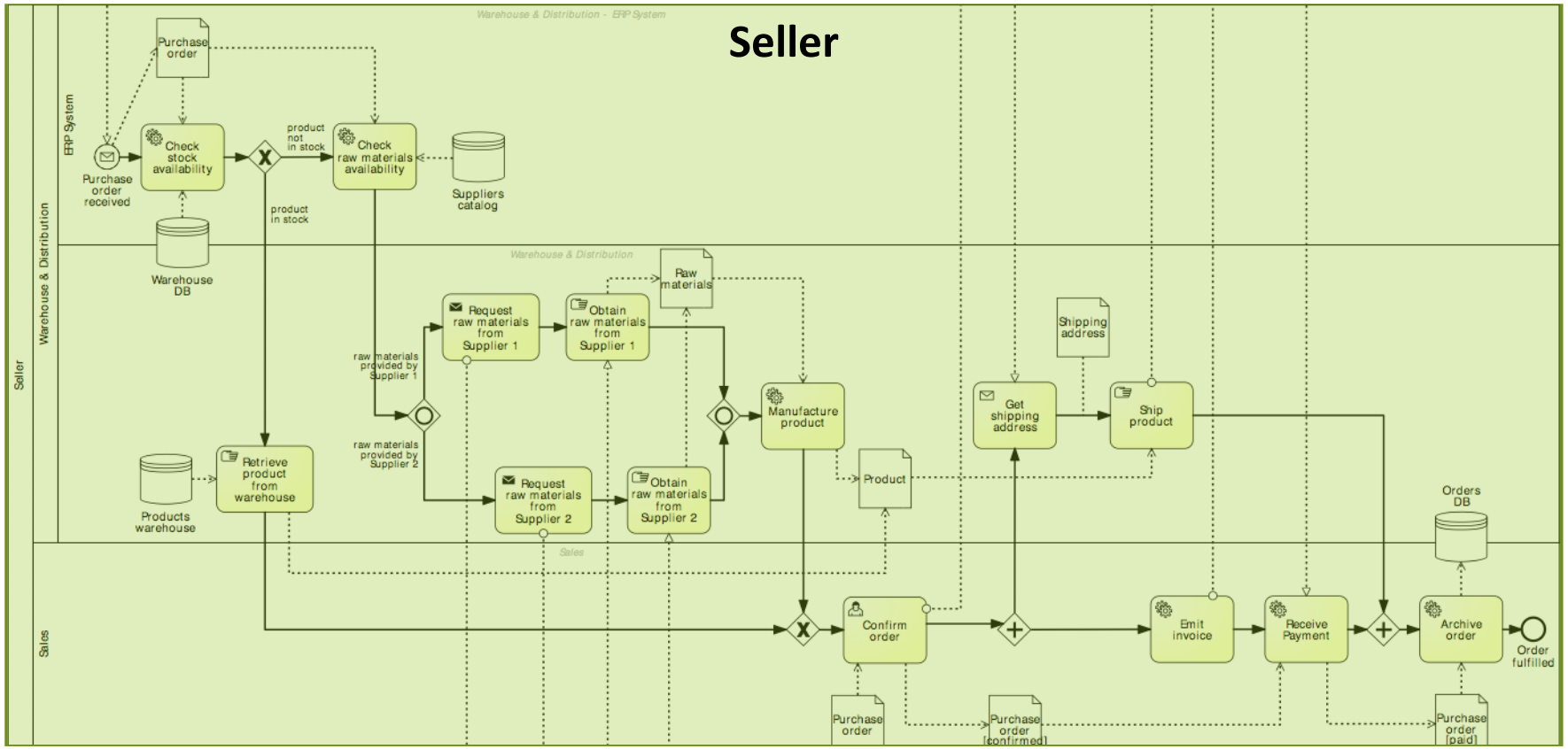
1. Identify the automation boundaries
2. Review manual tasks
3. Complete the process model
4. Adjust task granularity
5. Specify execution properties

Adapted from teaching material of Remco Dijkman, TU/e.

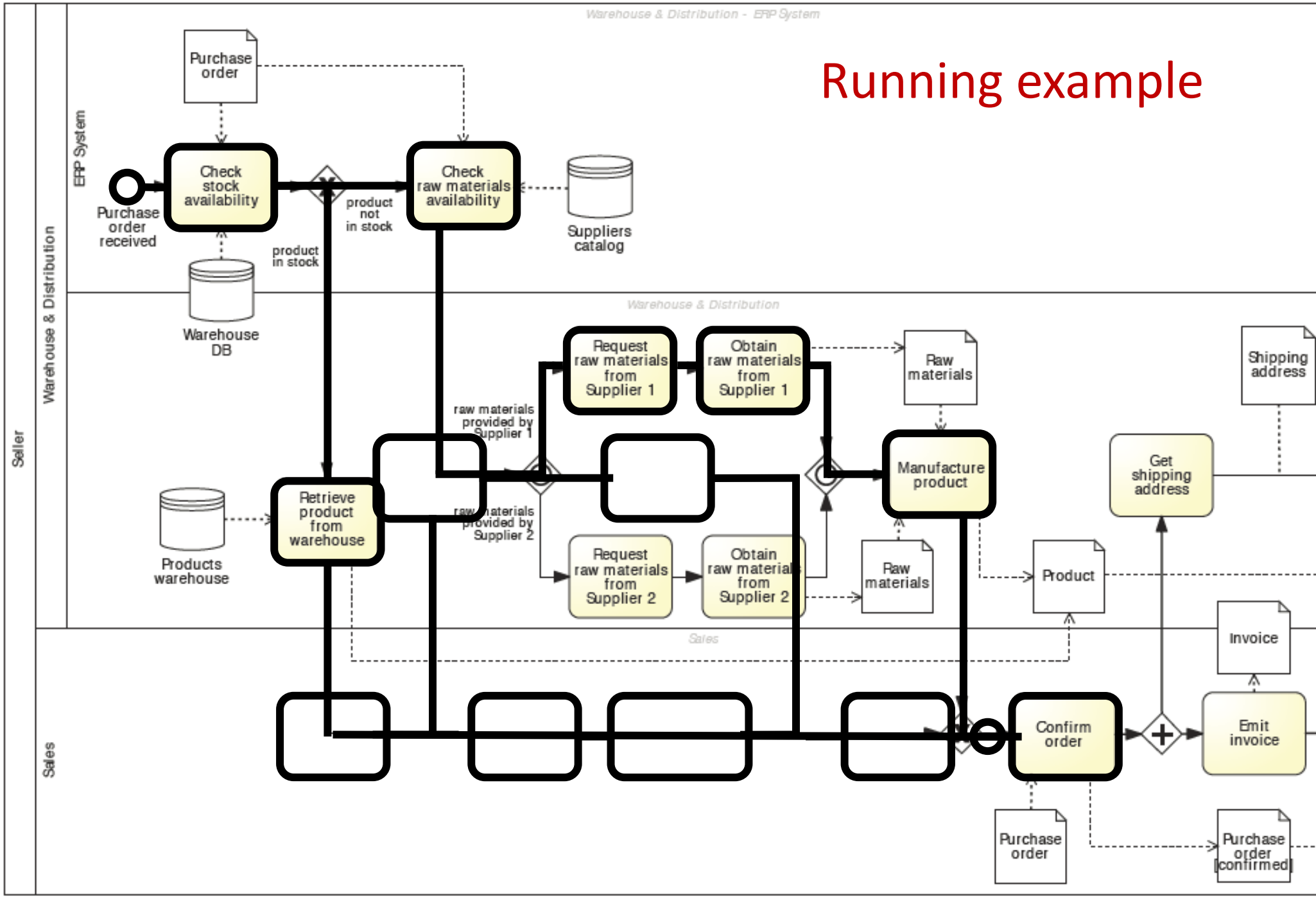
Running example



Running example



Running example



1. Identify the automation boundaries

Principle: not all parts of a process can be automated.

-> Start by identifying each task's type:

1

Automated tasks

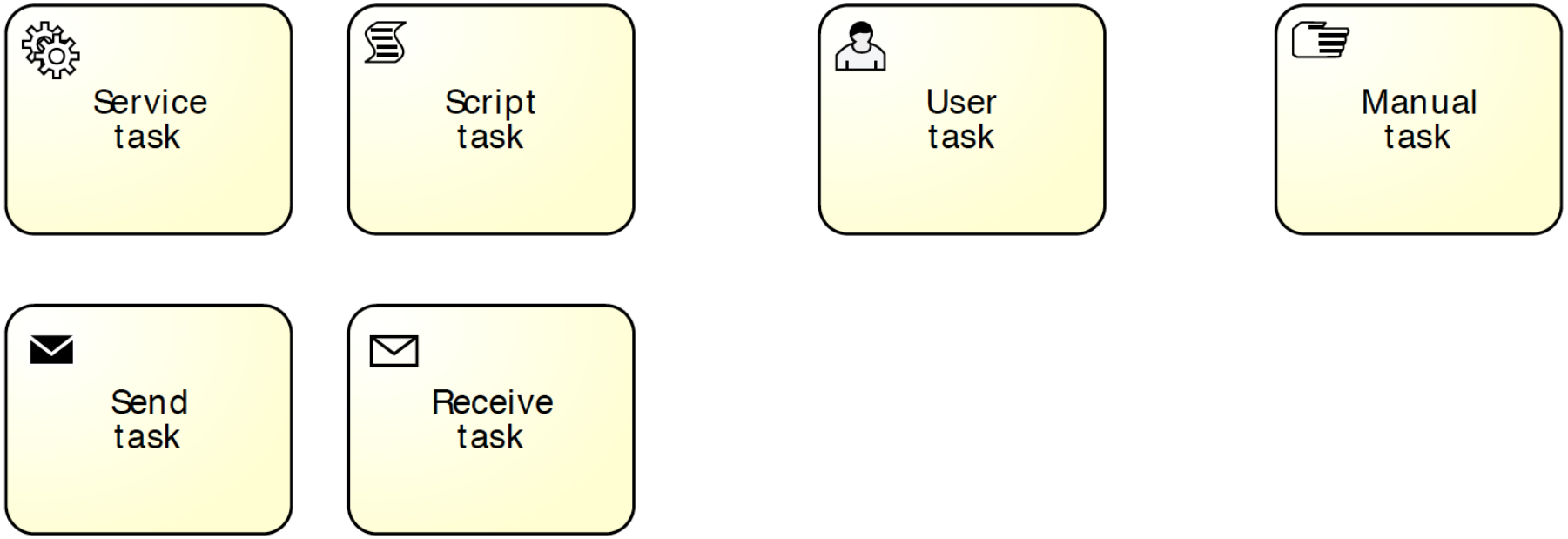
2

User tasks

3

Manual tasks

In BPMN: specify task markers



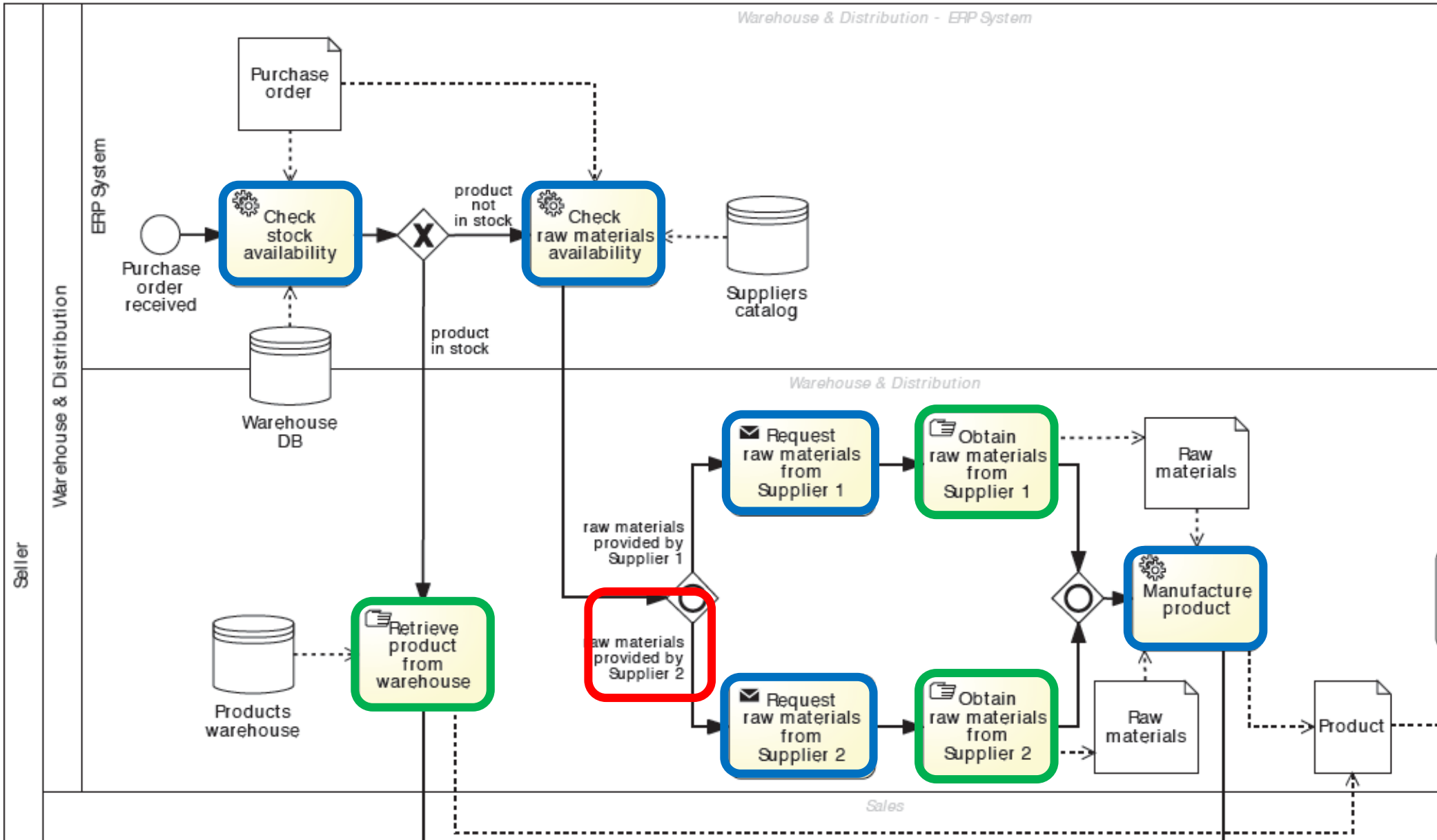
Automated tasks

User task

Manual task

In our example…

automated
user
manual



2. Review manual tasks

Principle: if it can't be seen by the BPMS, it doesn't exist.

-> Find ways to support manual tasks via IT:

- via user task
- via automated task

2. Review manual tasks

Principle: if it can't be seen by the BPMS, it doesn't exist.

-> Find ways to support manual tasks via IT:

- via user task
- via automated task

-> Leave them out of the automated process

3. Complete the process model

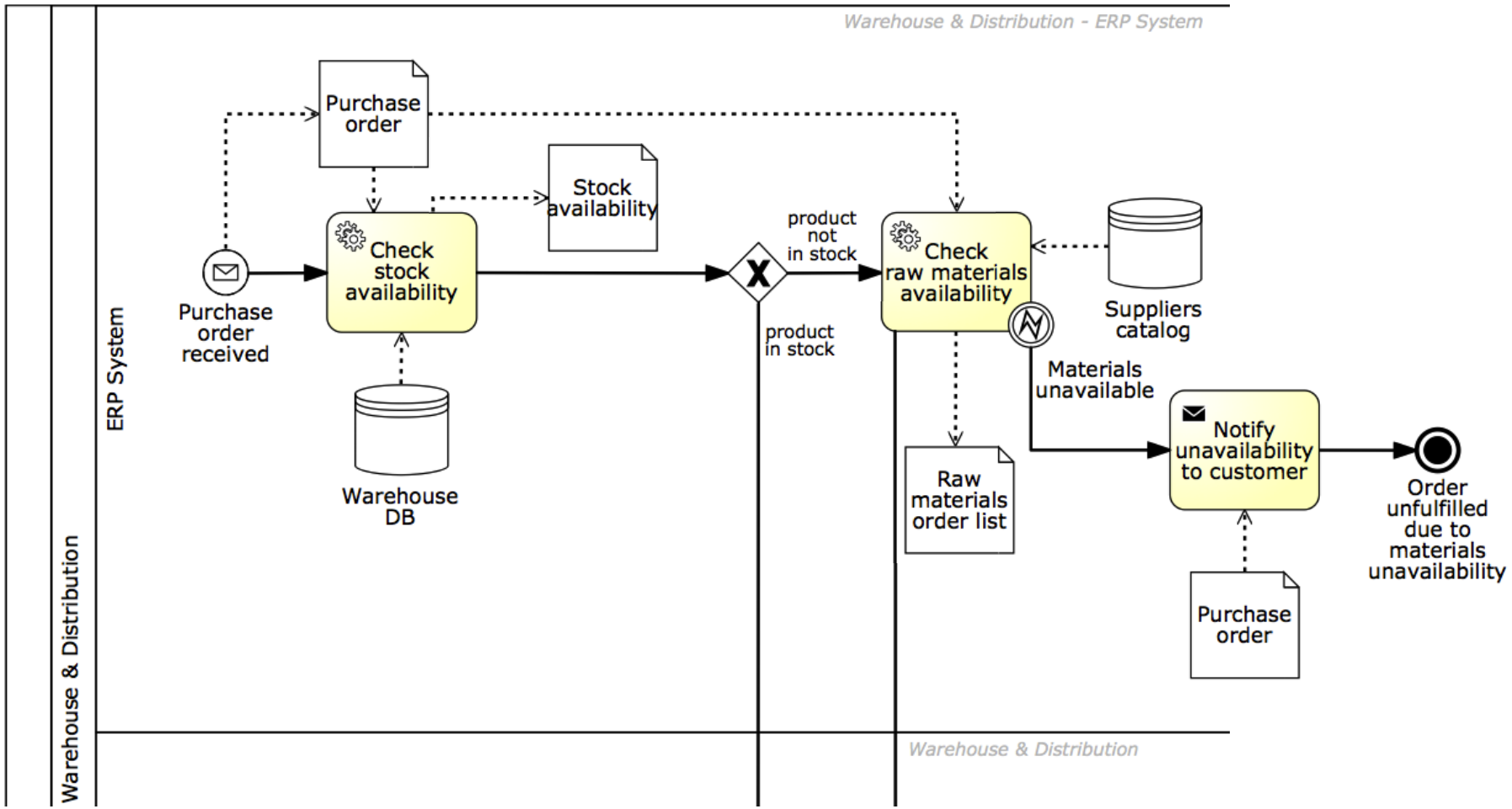
Principle: exceptions are the rule.

- Consider incomplete paths
- Rules of thumb
  - If we send something to another party, what happens if they do not respond? What happens if the response comes late? What happens if they do not respond the way we expect?
  - For each task: Can it go wrong and what happens if it goes wrong?
  - For each external party: Have we captured all messages or queries they might send us?

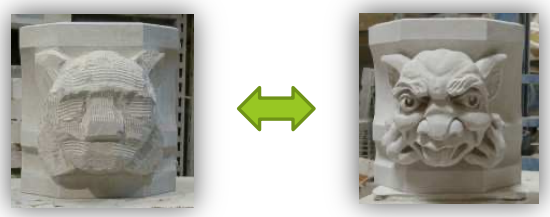
Principle: no data = no decisions, no tasks handover.

- Specify all (electronic) business objects
- For each task, determine which business objects it creates, reads, updates, delete (CRUD)
- For each decision, determine which objects it needs

In our example…

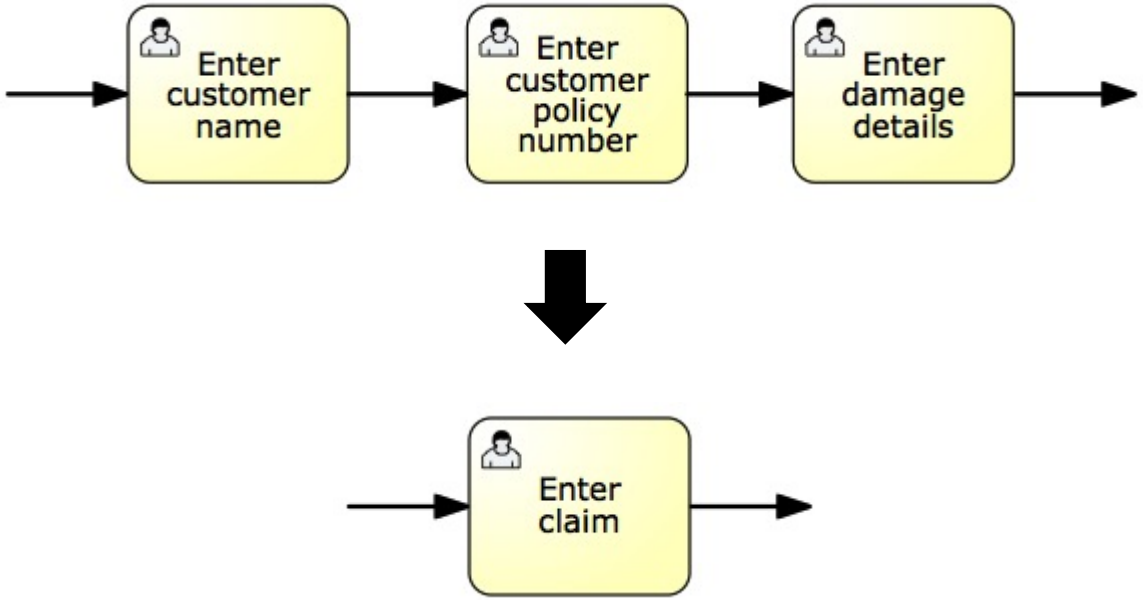


4. Adjust task granularity

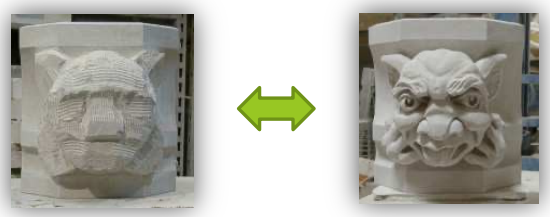


The goal is to coordinate handovers of work between resources, therefore:

-> Aggregate consecutive tasks assigned to the same performer



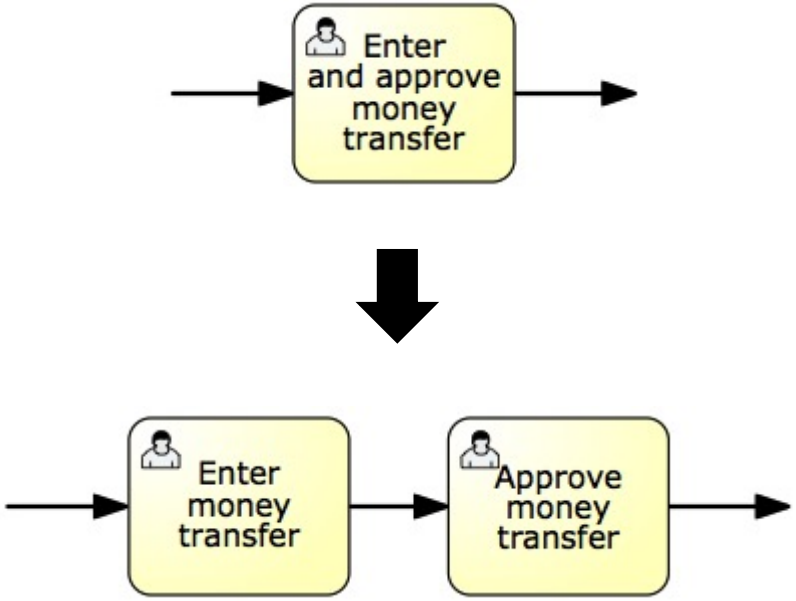
4. Adjust task granularity



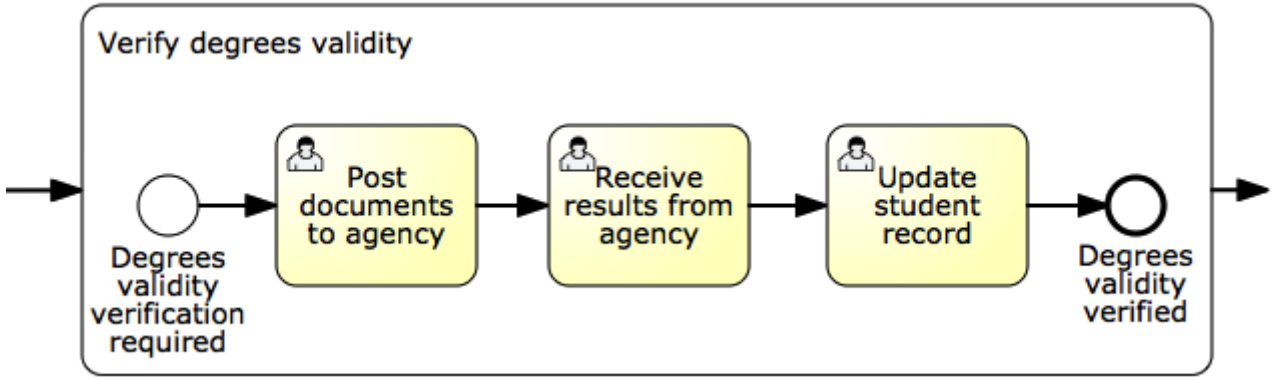
The goal is to coordinate handovers of work between resources, therefore:

-> Aggregate consecutive tasks assigned to the same performer

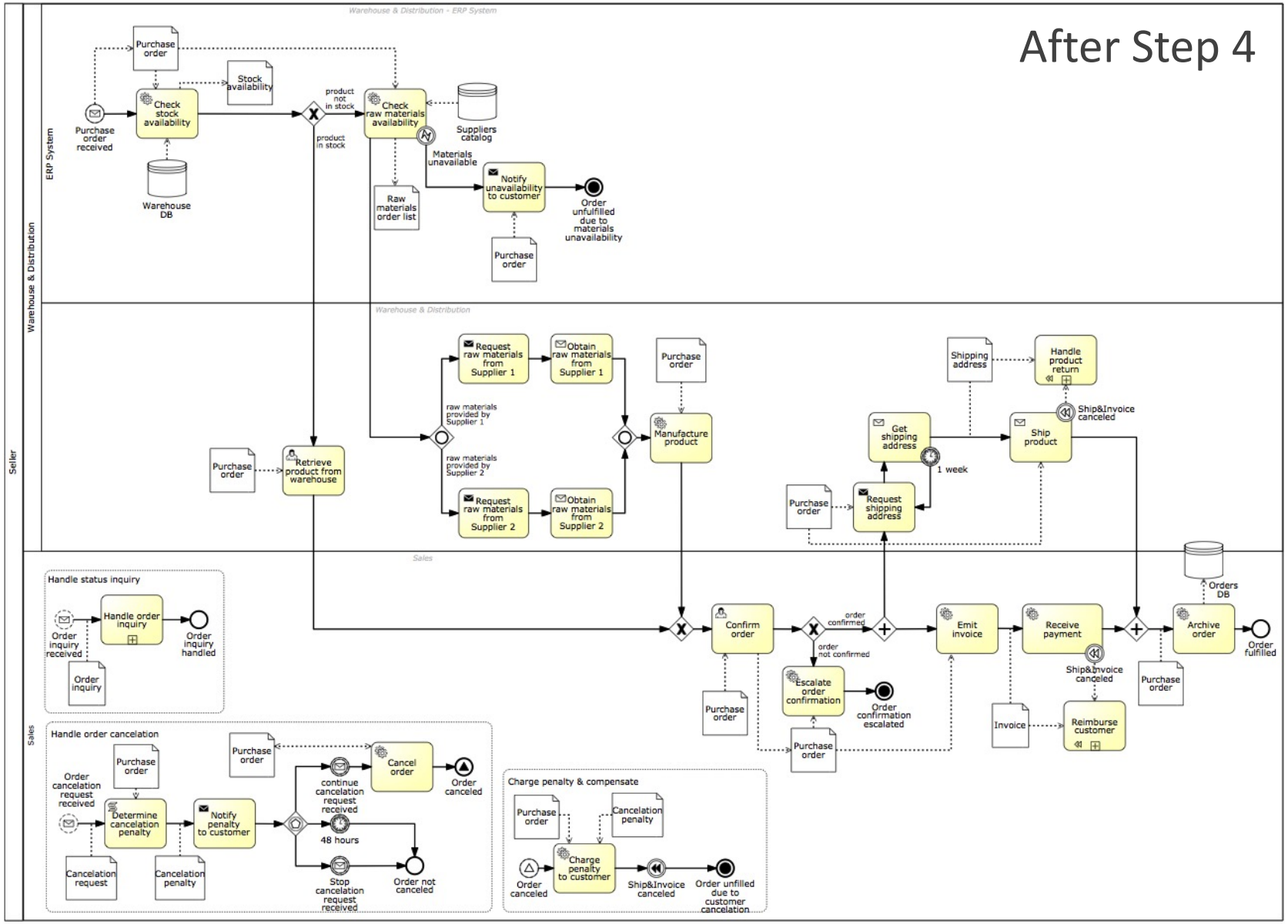
-> Split tasks if they require different performers



An exception to the rule



Our example…


After Step 4
Seller
Warehouse & Distribution
ERP System
Purchase order
Stock availability
Check stock availability
Purchase order received
Warehouse DB
product not in stock
product in stock
Check raw materials availability
Suppliers catalog
Materials unavailable
Raw materials order list
Notify unavailability to customer
Order unfulfilled due to materials unavailability
Purchase order
Warehouse & Distribution
Request raw materials from Supplier 1
Obtain raw materials from Supplier 1
raw materials provided by Supplier 1
raw materials provided by Supplier 2
Request raw materials from Supplier 2
Obtain raw materials from Supplier 2
Purchase order
Manufacture product
Purchase order
Retrieve product from warehouse
Shipping address
Handle product return
Ship&Invoice canceled
Get shipping address
Ship product
1 week
Purchase order
Request shipping address
Sales
Handle status inquiry
Order inquiry received
Handle order inquiry
Order inquiry handled
Order inquiry
Orders DB
Confirm order
order confirmed
order not confirmed
Escalate order confirmation
Order confirmation escalated
Purchase order
Purchase order
Emit invoice
Receive payment
Ship&Invoice canceled
Invoice
Reimburse customer
Archive order
Order fulfilled
Purchase order
Sales
Handle order cancelation
Order cancelation request received
Purchase order
Determine cancelation penalty
Notify penalty to customer
Cancelation request
Cancelation penalty
Purchase order
continue cancelation request received
Cancel order
Order canceled
48 hours
Stop cancelation request received
Order not canceled
Charge penalty & compensate
Purchase order
Cancelation penalty
Charge penalty to customer
Order canceled
Ship&Invoice canceled
Order unfilled due to customer cancelation

Bridging the gap: one task at a time

1. Identify the automation boundaries
2. Review manual tasks
3. Complete the process model
4. Adjust task granularity
5. Specify execution properties

5. Specify execution properties

→ Process variables, messages, signals, errors

→ Task and event variables and their mappings to process variables

→ Service details

→ Code snippets

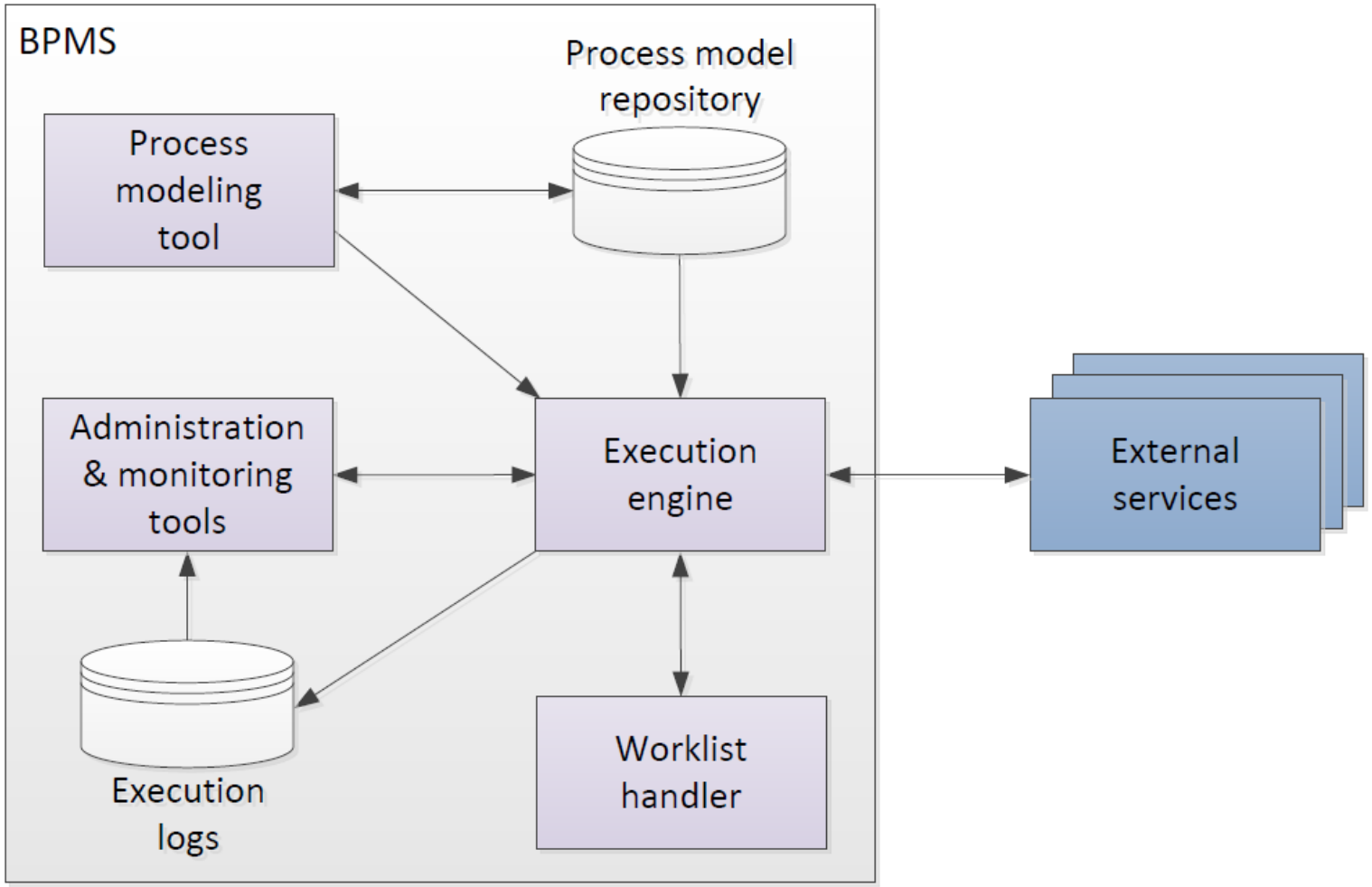
→ Participant assignment rules

→ User interface specification (e.g. forms)

→ Task, event and sequence flow expressions

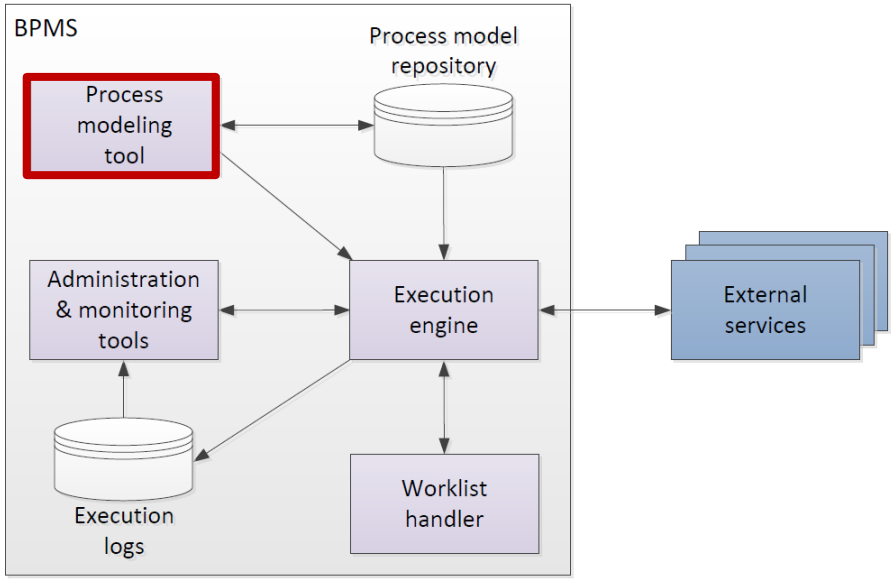
→ BPMS-specific: work queues, forms, connectors…

Business Process Management System

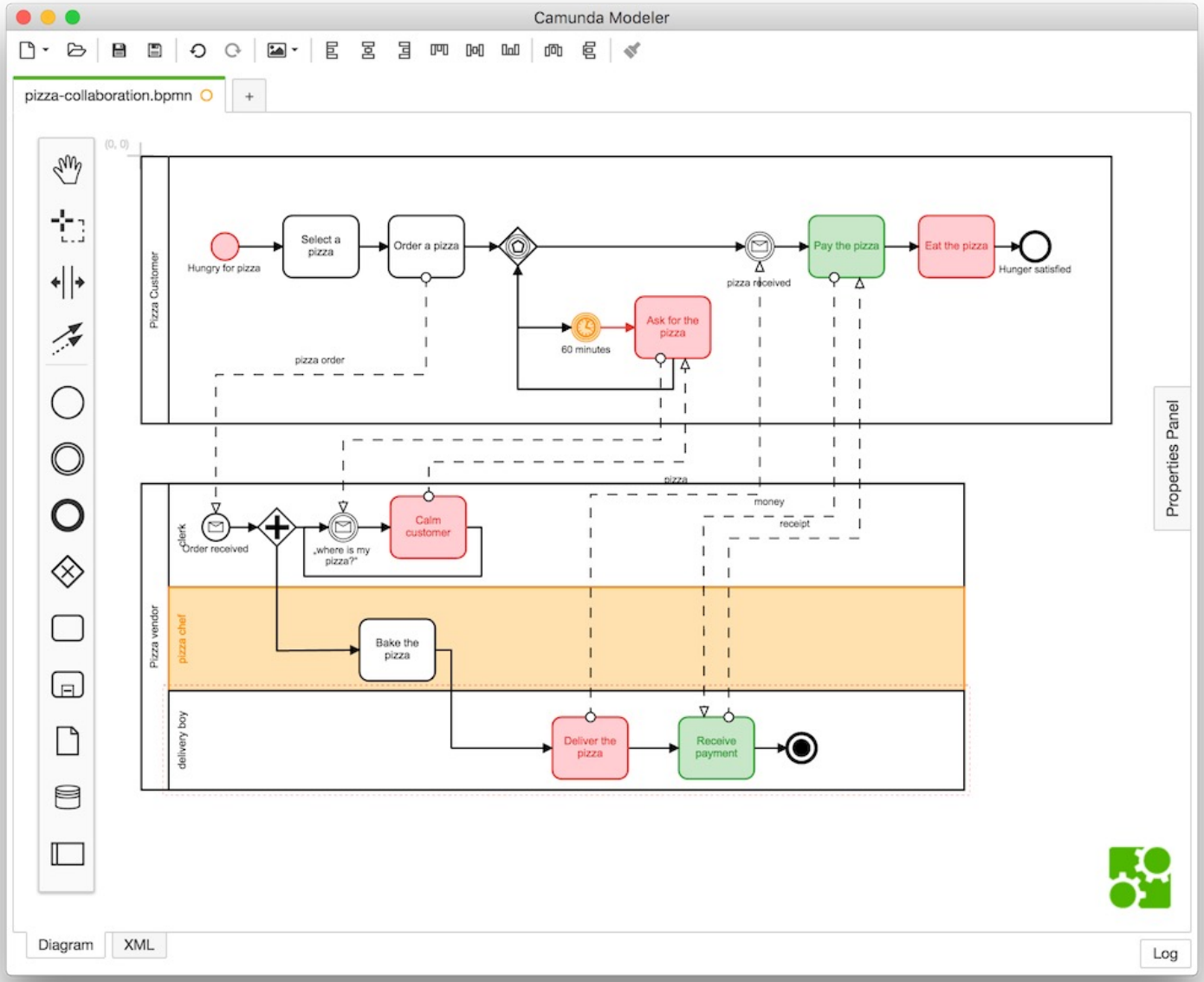


Process modeling tool

- To create and modify executable process models (by specifying execution properties)
- To store and retrieve automation solutions from a process model repository



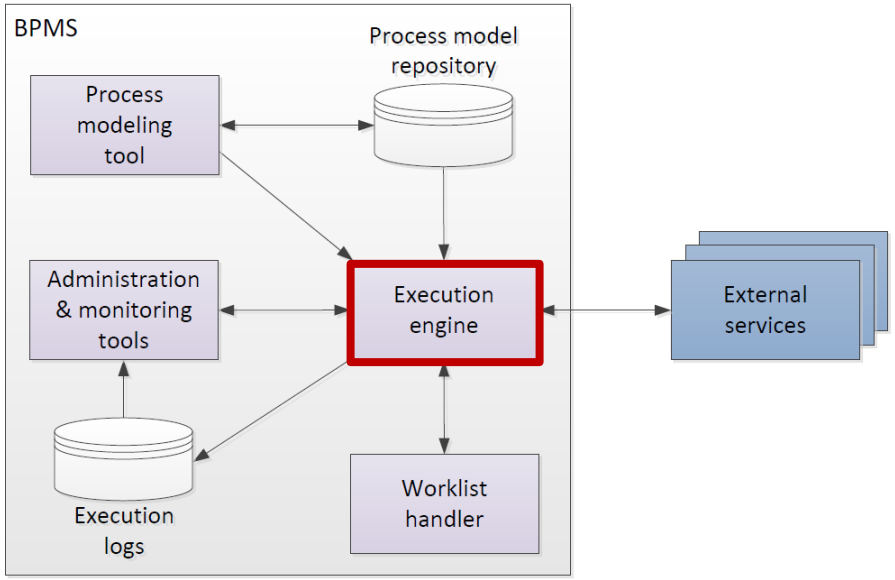
Example of a Process modeling tool



Camunda Modeler

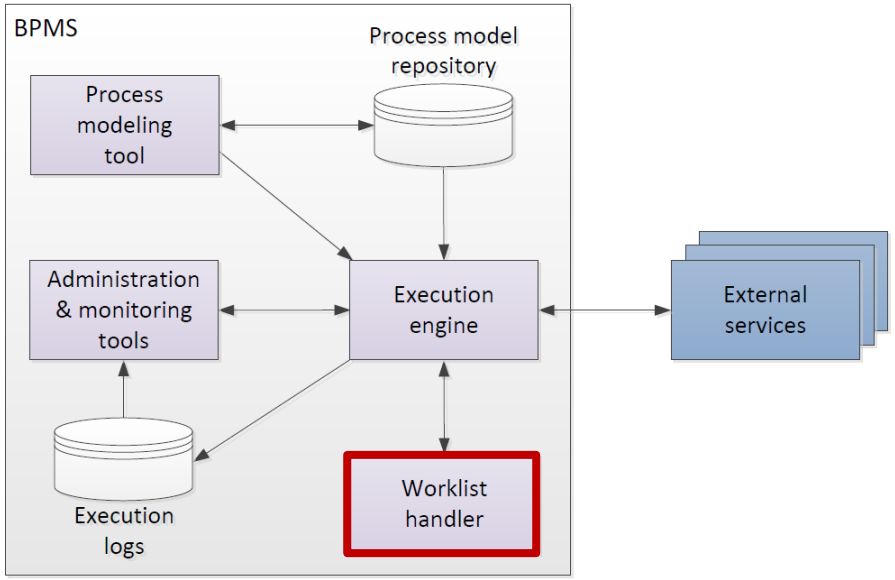
Execution Engine

- Instantiates executable process models (also called "cases")
- Orchestrates distribution of work items to process participants and software services in order to execute a business process from start to end
- Logs execution data

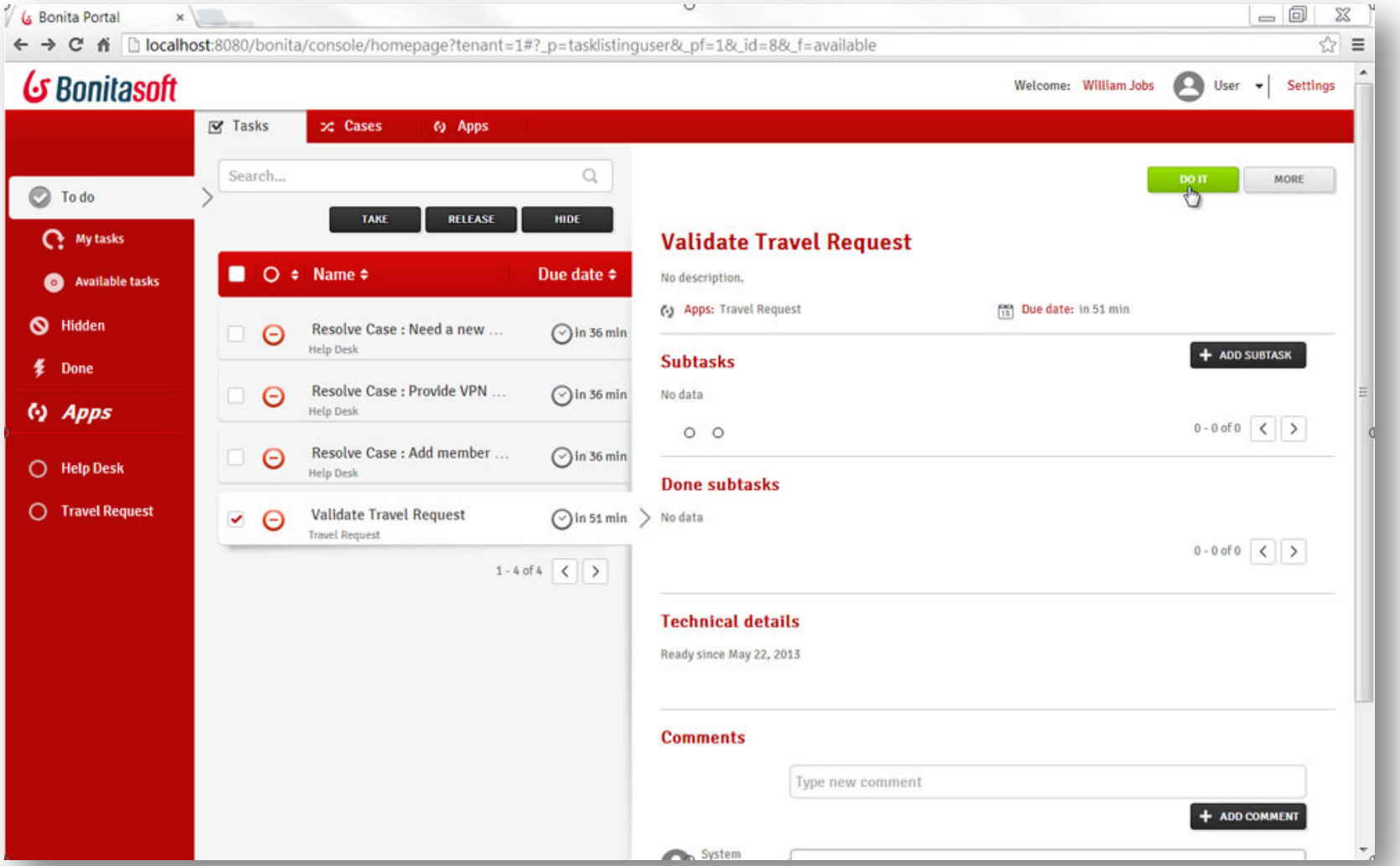


Worklist Handler

- Imagine it as an “inbox”
- Offers work items to process participants and allows participants to commit to these work items
- Handles participants’ work queues and work item priorities
- May provide social network capabilities



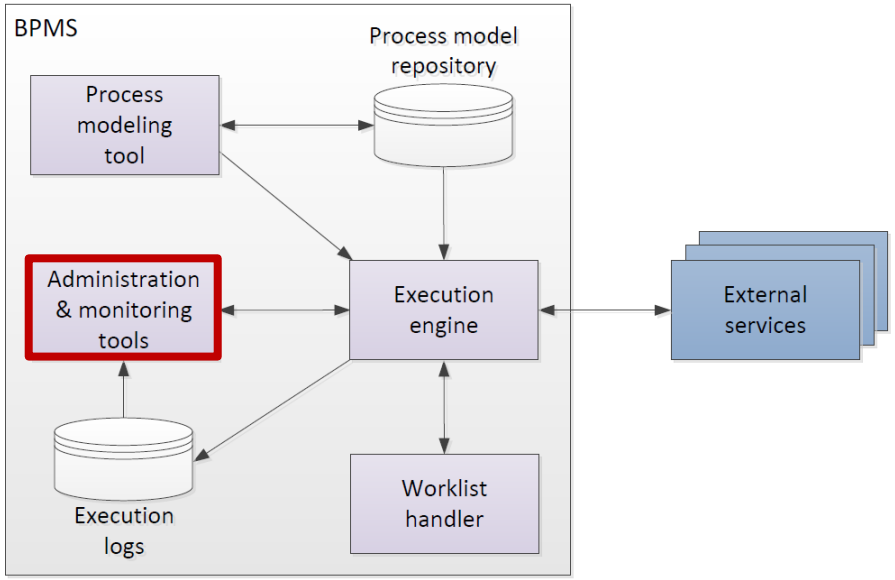
Example of a Worklist Handler



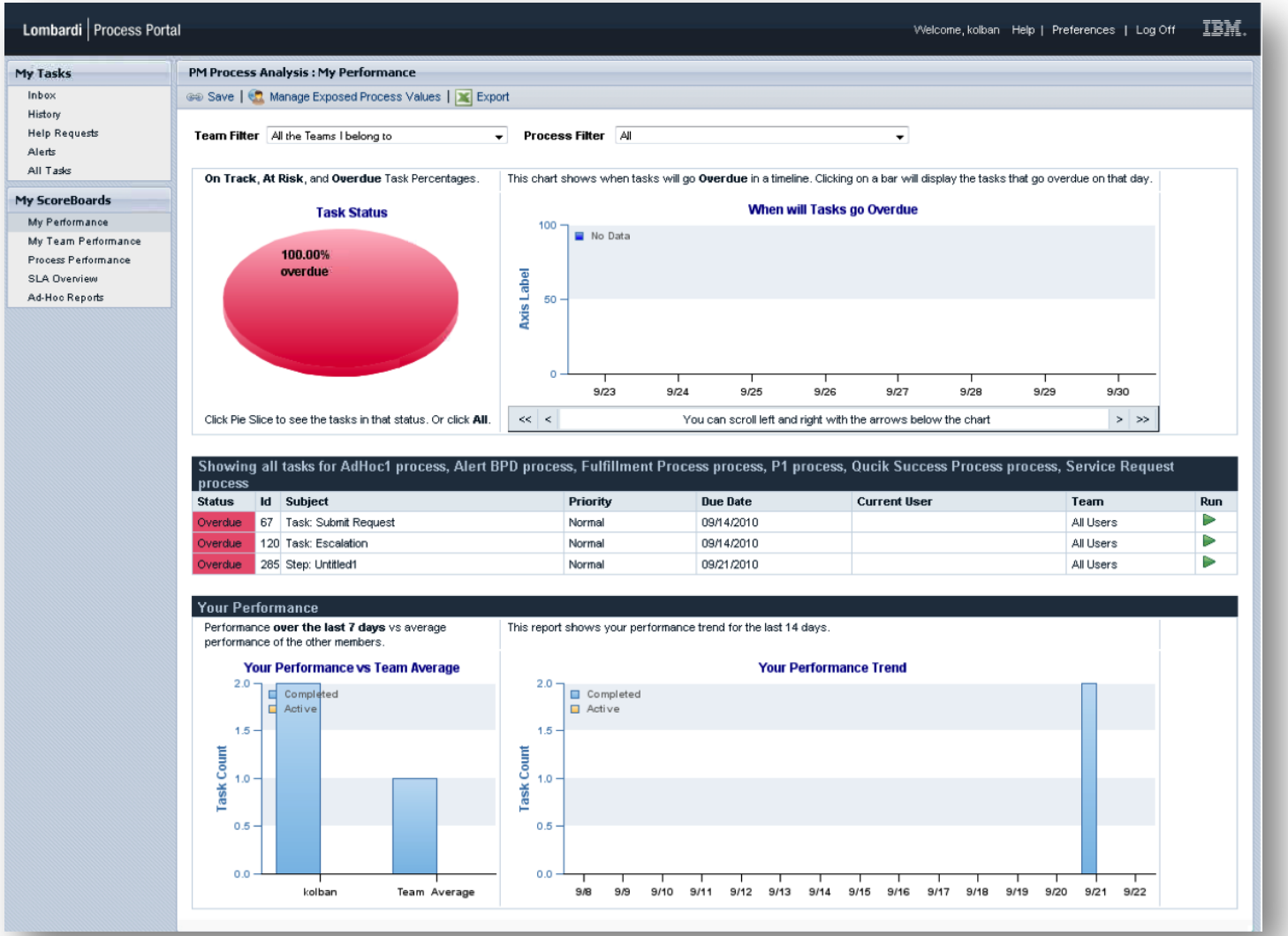
Bonita Soft Bonita Open Solution

Administration & Monitoring Tools

- To manage automation solutions
- To configure access to system components
- To monitor participants availability and performance of process cases



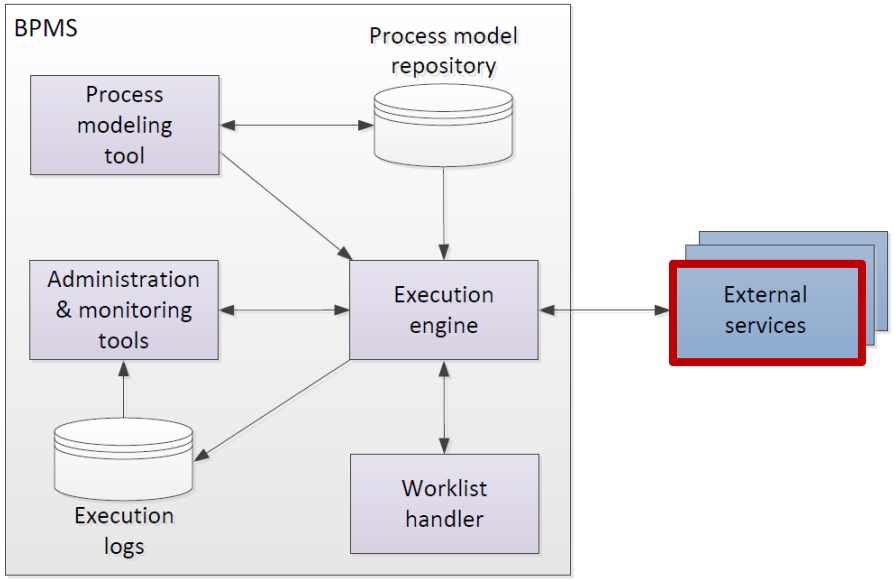
Example of an Administration & Monitoring Tool



IBM BPM Process Portal

External Services

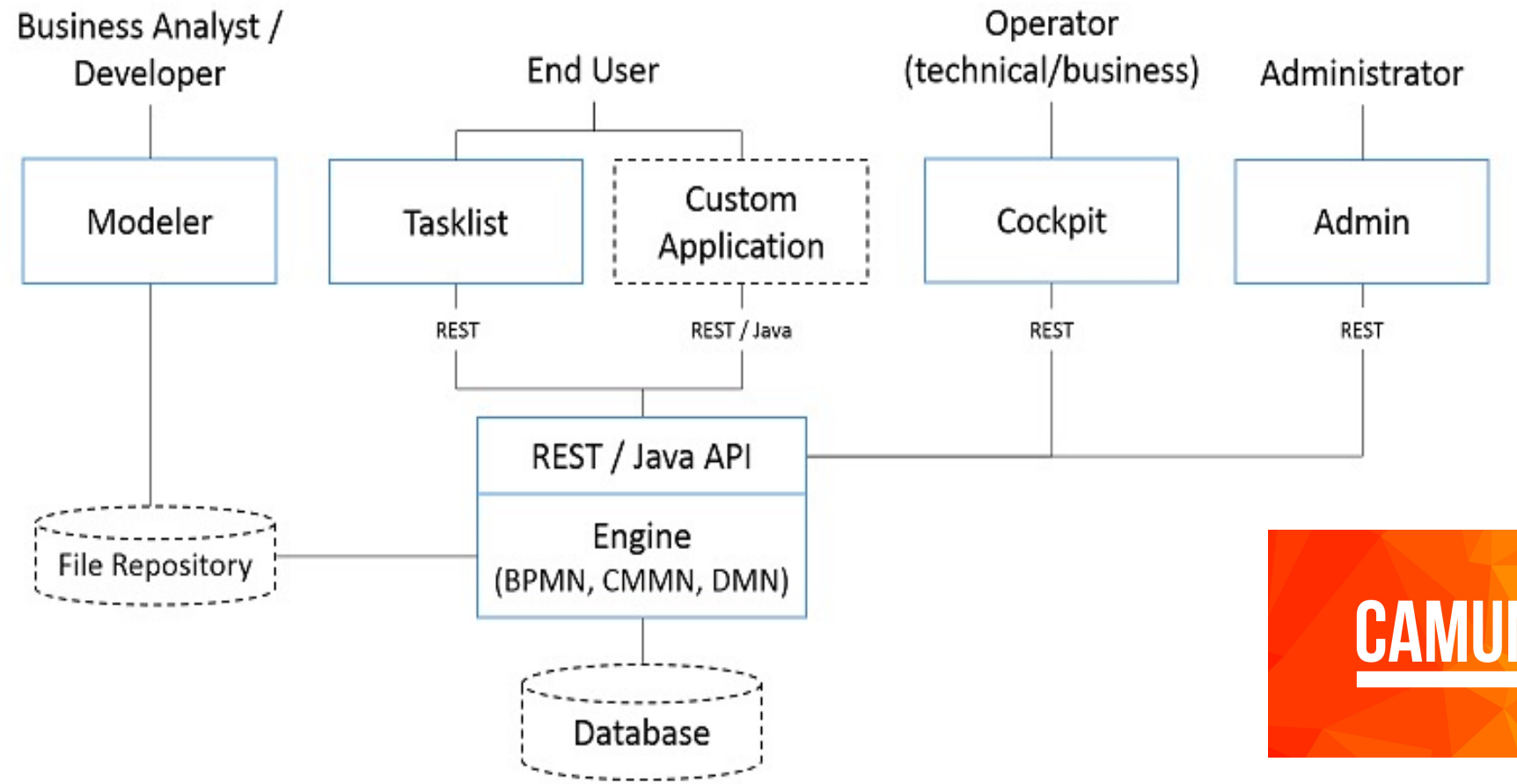
- Expose a service interface with which the engine can interact
- The engine provides the invoked service with the necessary data it will need to perform the activity for a specific case
- Examples: rules engine, email or Twitter notification, DB connector, CRM connector…



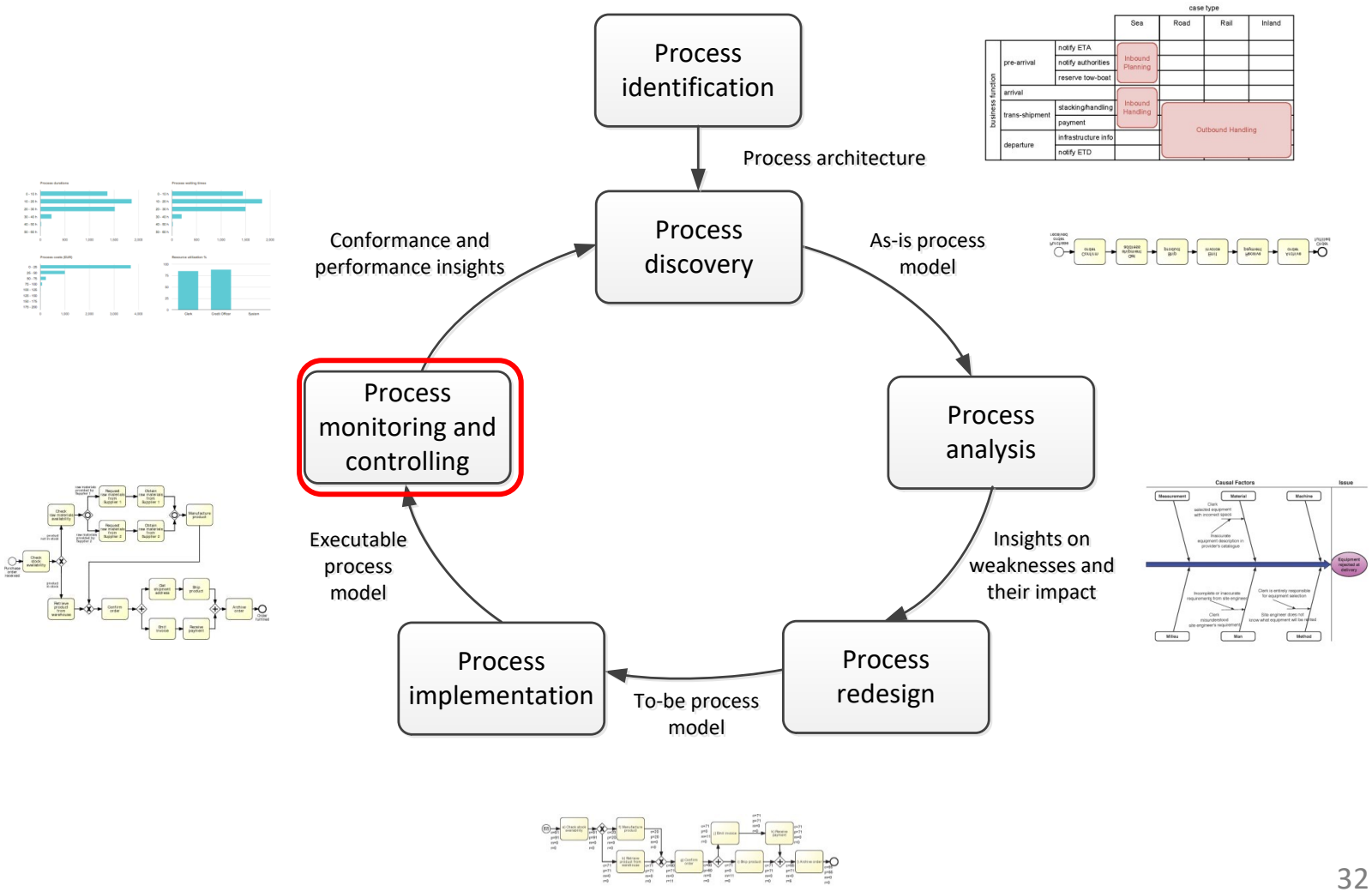
BPMS Landscape

Closed source	Commercial open-source
• BIC Platform • BizAgi • IBM BPM • OpenText BPM • Oracle BPMS • Pega Platform • Software AG webMethods	• Bonita Platform • Camunda • Flowable • JBoss jBPM • ProcessMaker

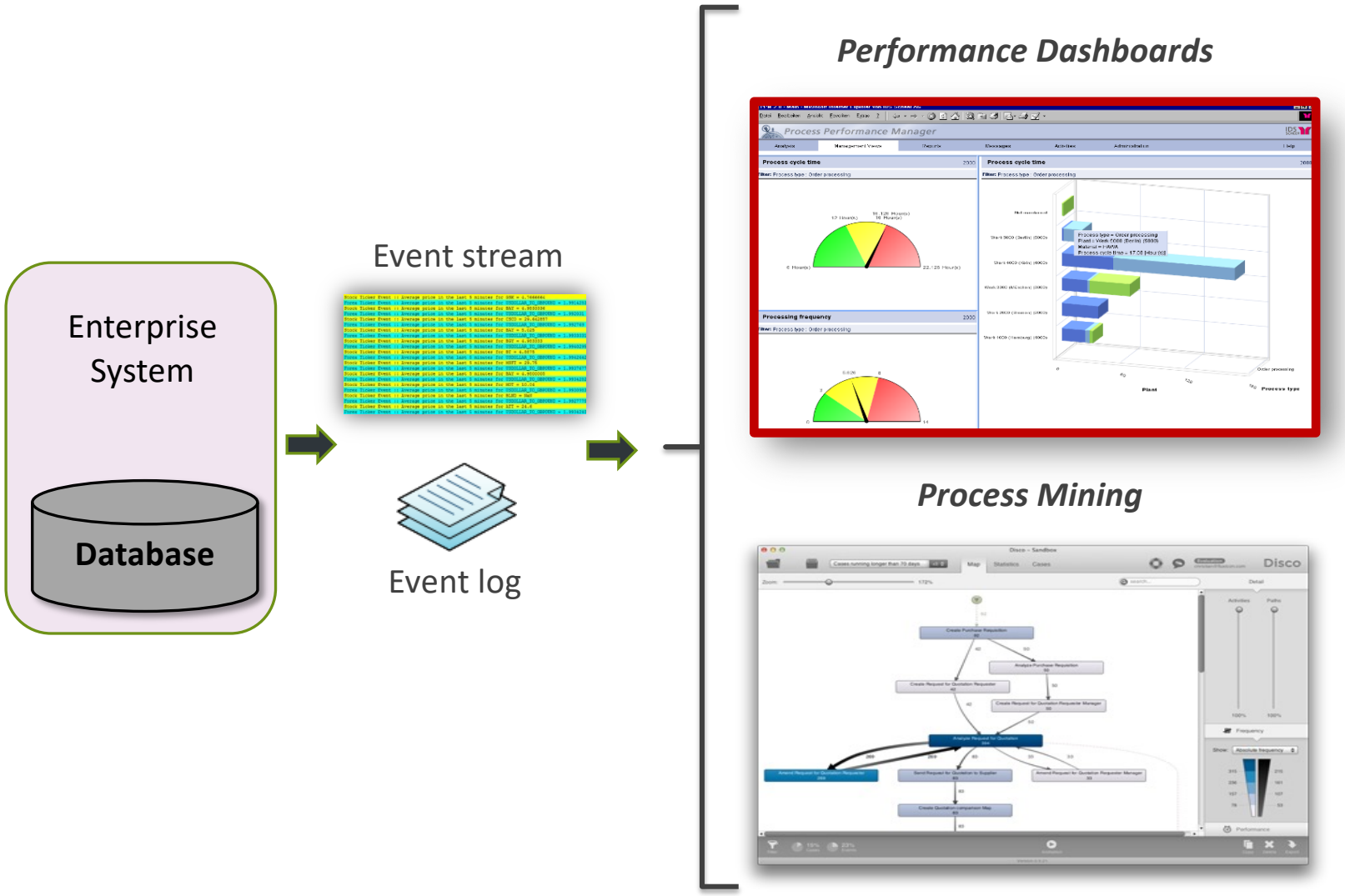
Camunda Architecture



Where are we?


Process identification
Process architecture
Process discovery
As-is process model
Process analysis
Insights on weaknesses and their impact
Process redesign
To-be process model
Process implementation
Executable process model
Process monitoring and controlling
Conformance and performance insights
case type
Sea
Road
Rail
Inland
business function
pre-arrival
notify ETA
notify authorities
reserve tow-boat
arrival
trans-shipment
stacking/handling
payment
departure
infrastructure info
notify ETD
Inbound Planning
Inbound Handling
Outbound Handling
Causal Factors
Issue

Business Process Monitoring

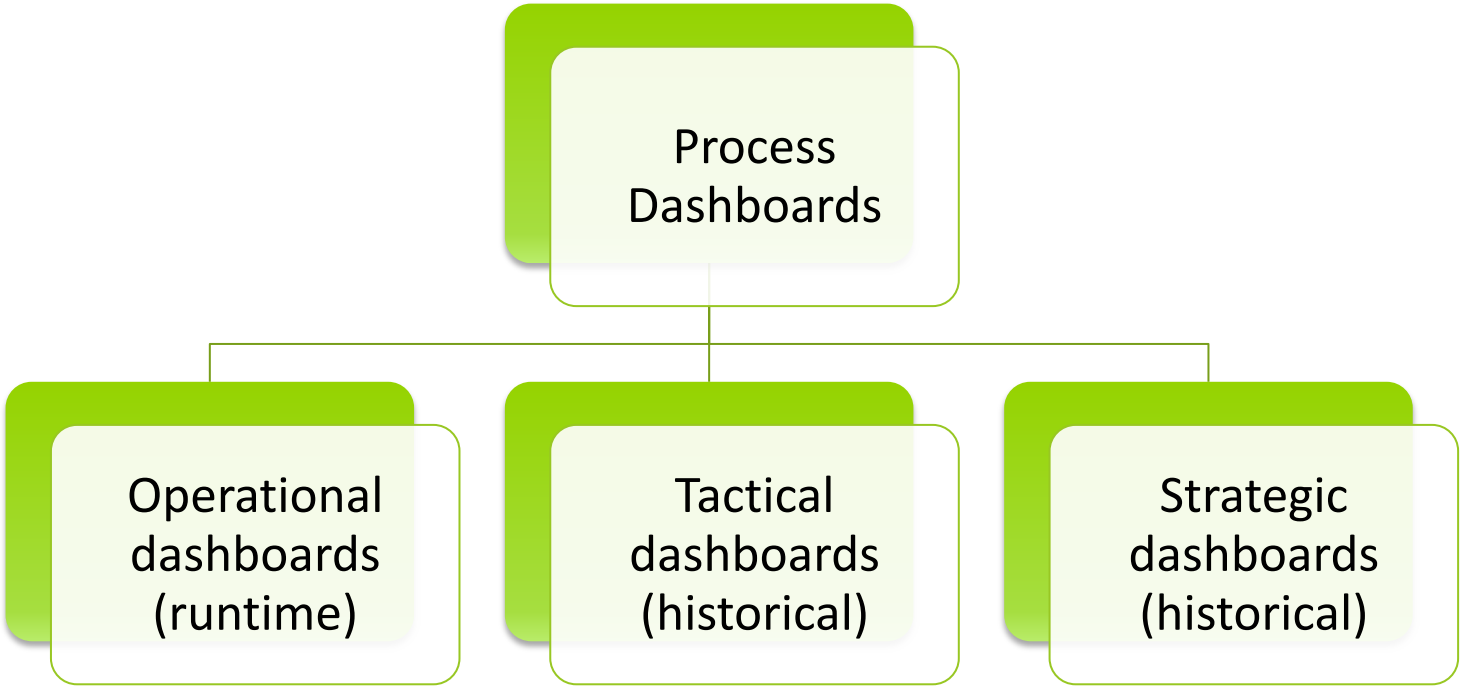


Structure of a Business Process Event Log

Case ID	Timestamp	Activity	Resource	Loan goal	Requested amt	Offered amt
C001	18-10-2016	Check completeness	Sue	Mortgage	100 000	-
C001	19-10-2016	Check credit history	Sue	Mortgage	100 000	-
C001	19-10-2016	Calculate risk score	Bob	Mortgage	100 000	-
C001	20-10-2016	Make offer	Mike	Mortgage	100 000	70 000
C001	25-10-2016	Make offer	Mike	Mortgage	100 000	80 000
C002	20-10-2016	Check completeness	Sue	Car	15 000	-
C002	20-10-2016	Check credit history	Sue	Car	15 000	-
C002	22-10-2016	Calculate risk score	Elsa	Car	15 000	-
C002	24-10-2016	Reject application	Elsa	Car	15 000	-
C003	02-11-2016	Check completeness	Maria	Mortgage	30 000	-
C003	04-11-2016	Ask for additional data	Maria	Mortgage	30 000	-
C003	10-11-2016	Check credit history	Maria	Mortgage	30 000	-
...	...	...	...	...	...	...

Process performance dashboard

A visualization of a collection of process performance measures to put into evidence information and patterns to help business users answer one or more business questions

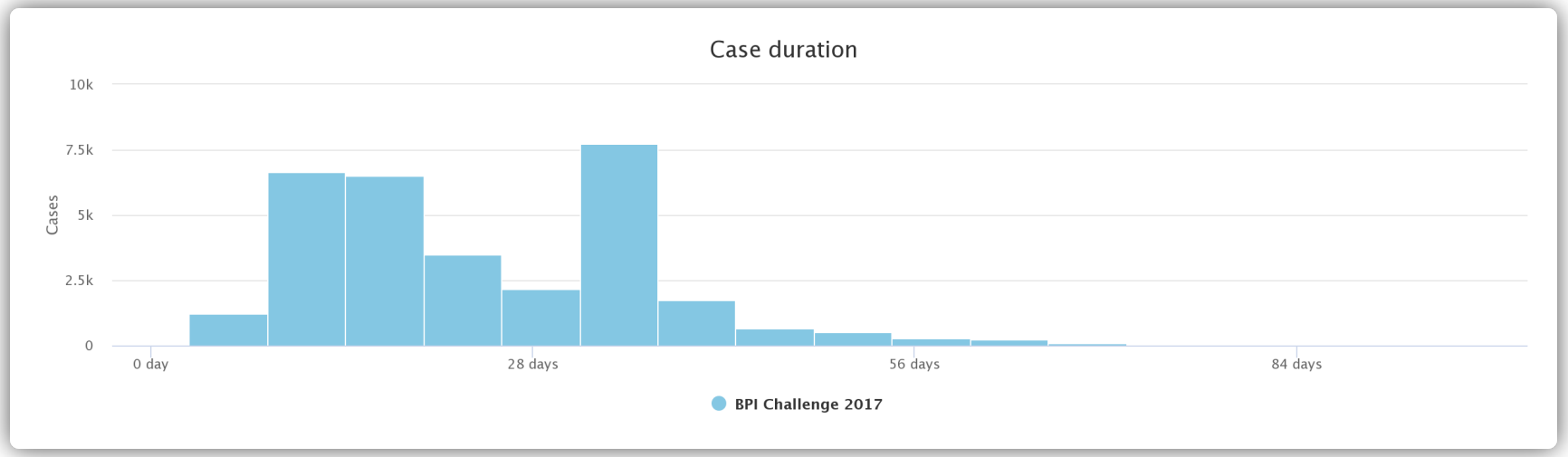


Operational process dashboards

- Aimed at process workers & operational managers
- Emphasis on monitoring (detect-and-respond), e.g.:
  - Work-in-progress
  - Problematic cases – e.g., overdue/at-risk cases
  - Resource load

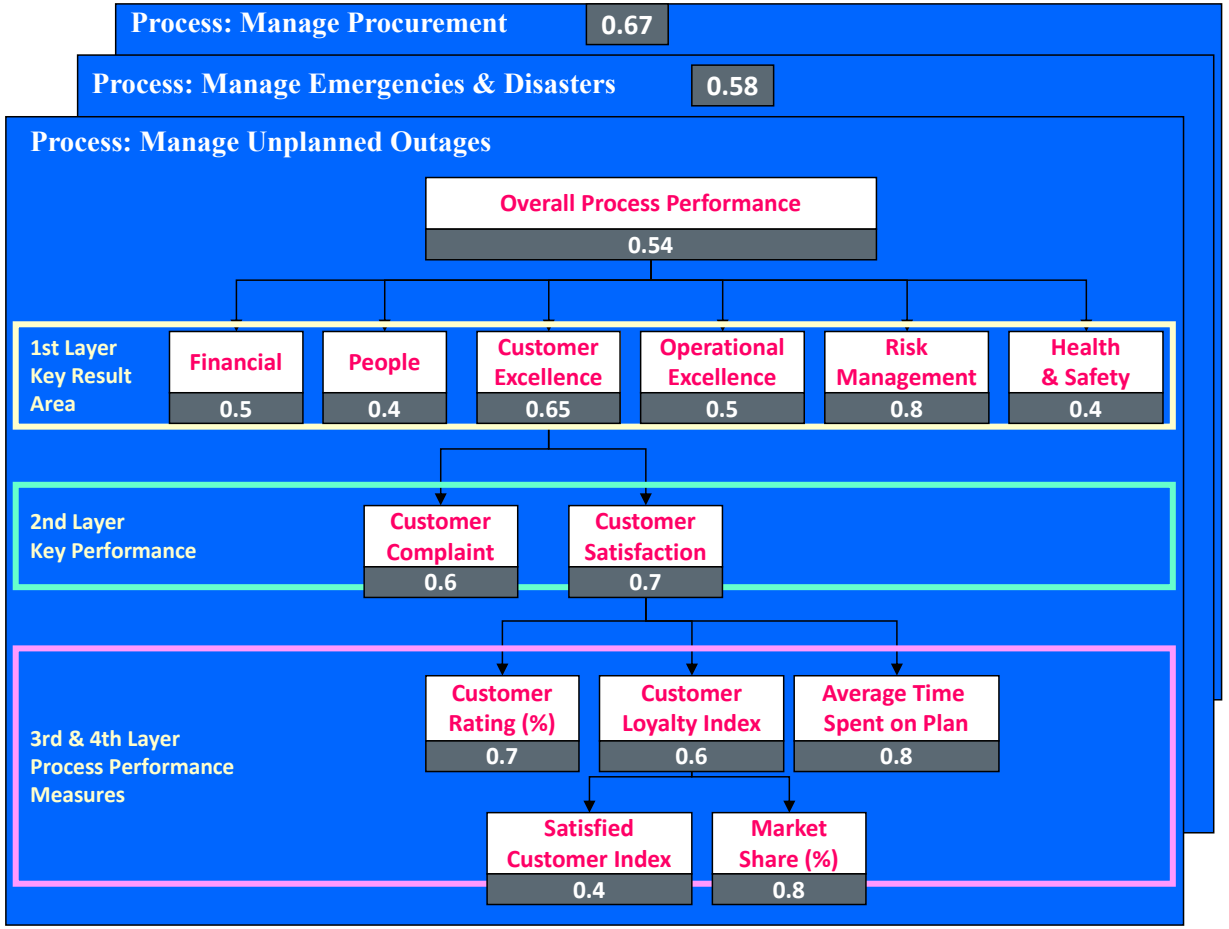
Tactical performance dashboards

- Aimed at process owners / managers
- Emphasis on analysis and management
  - E.g., detecting bottlenecks
- Typical process performance indicators
  - Cycle times
  - Error rates
  - Resource utilization



Strategic performance dashboards

- Aimed at executives & managers
- Emphasis on linking process performance to strategic objectives



Process performance dashboards

A dashboard is a collection of widgets

Each widget displays at least one performance measure

E.g. The average case duration

Four types of widgets:

Single-dimensional widgets

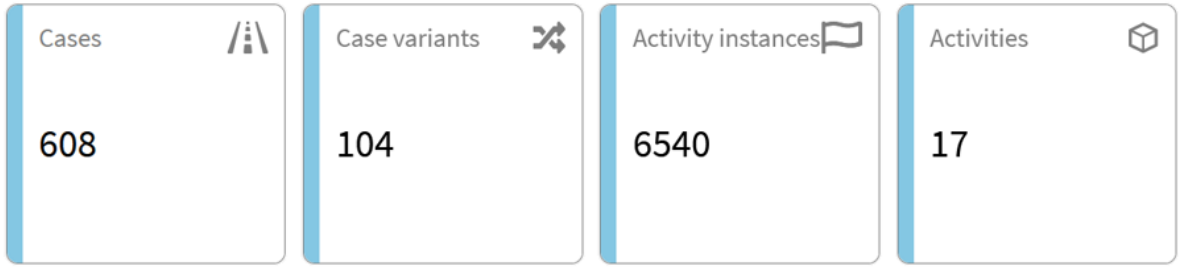
Two-dimensional widgets

Three-dimensional widgets

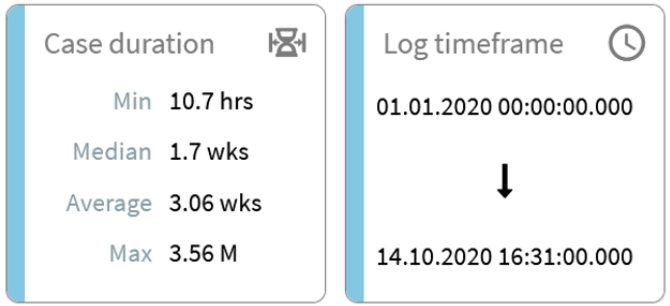
N-dimensional widgets (tables)

Process performance dashboard widgets

- Single-dimensional widgets display one variable



- We can group multiple measures in one widget for convenience.



Process performance dashboard widgets

Two-dimensional widgets display two variables:

- Independent variable (x-axis) – variable with respect to which we want to analyze
- Dependent variable (y-axis) – the variable that we wish to analyze

Usually, a two-dimensional widget is a chart

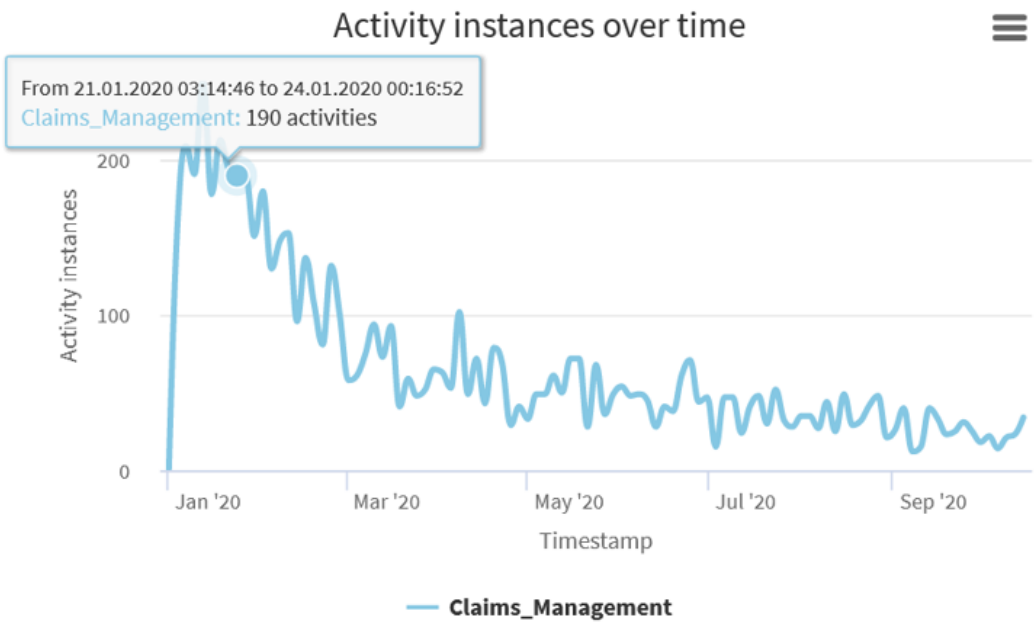
- Bar chart, pie chart, etc.

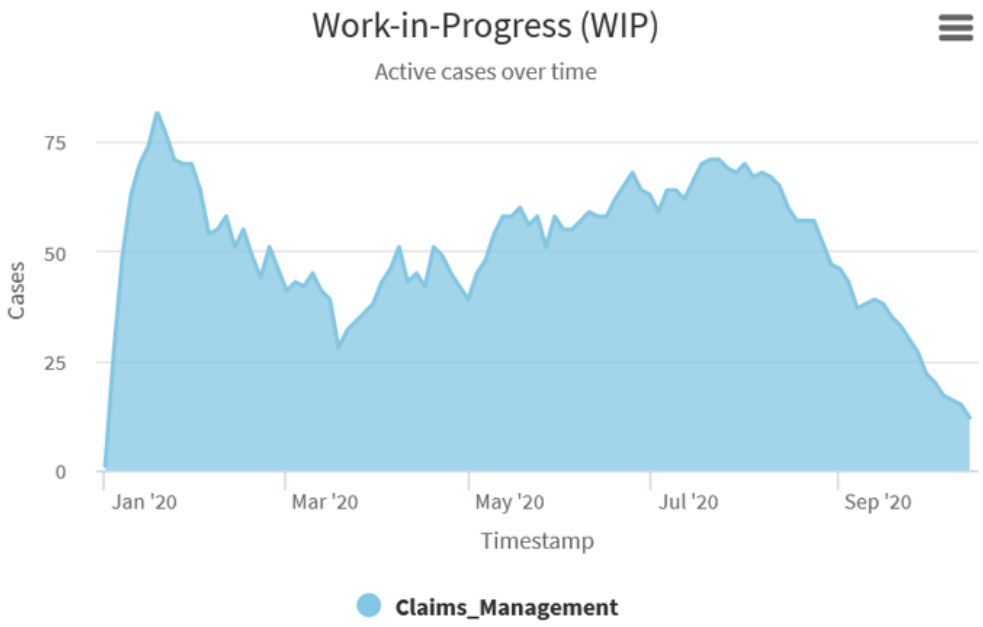
Types of two-dimensional charts:

- Longitudinal charts: x-axis is time, y-axis is a performance measure
- Cross-sectional chart: x-axis is an attribute (e.g. activity, resource, country), y-axis is a performance measure
- Histogram: x-axis is a performance measure, y-axis is a “count”

Process performance dashboard widgets

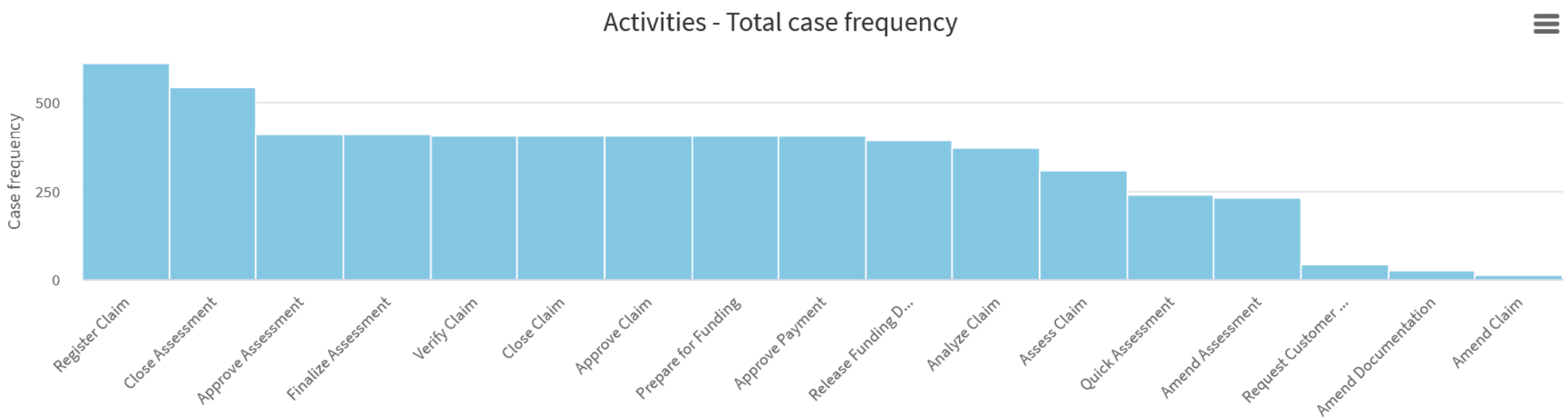
Longitudinal chart





Process performance dashboard widgets

Cross-sectional chart

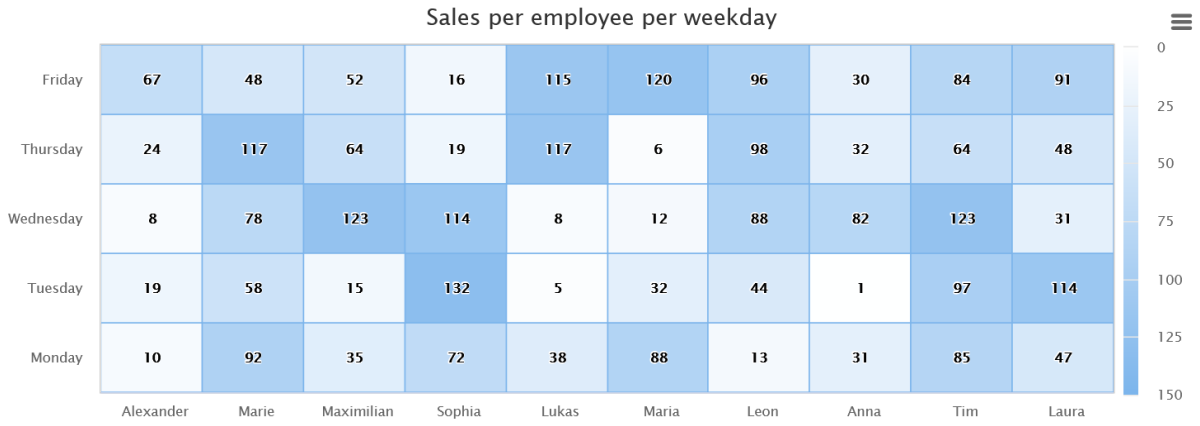


Process performance dashboard widgets

Cross-sectional chart

Process performance dashboard widgets

- Three-dimensional widgets display three variables, usually:
  - Two cross-sectional variables
  - A performance measure



- The third dimension can be coded in different ways:
  - Heat map: intensity and/or color
  - Bubble chart: size and/or intensity
  - Scatter plot: useful when the third dimension is Boolean

Process performance dashboard widgets

- N-dimensional widgets display N dimensions, usually:
  - A cross-sectional variable (e.g. the customer, the case identifier, etc.)
  - Several performance measures

- In general, N-dimensional widgets are tables

Cases

Case ID	Activity instances	Variant	Start time	End time	Duration
361	38	69	20.02.2020 20:38:00	13.05.2020 00:42:00	2.7 months
249	32	103	03.02.2020 04:57:00	25.04.2020 23:31:00	2.72 months
223	29	74	29.01.2020 08:22:00	13.05.2020 05:05:00	3.44 months
637	28	96	02.04.2020 15:24:00	27.06.2020 21:53:00	2.83 months
872	25	99	09.05.2020 05:06:00	20.08.2020 20:01:00	3.4 months
438	24	65	05.03.2020 01:49:00	13.06.2020 15:20:00	3.3 months
929	24	48	16.05.2020 08:41:00	09.08.2020 18:03:00	2.81 months
1011	23	62	28.05.2020 13:20:00	03.09.2020 21:13:00	3.23 months
459	23	51	07.03.2020 22:12:00	28.05.2020 17:36:00	2.69 months

1 / 61 [1 - 10 / 608]

Designing Process Performance Dashboard

Identify a user or users and define a clear question or set of questions that users will answer with this question

Identify the type of dashboard elements based on the insights required to answer the question

Identify a type of visualization (e.g. type of chart) for the element

Determine x-axis (independent variable): longitudinal time), cross-sectional (attribute)

Determine y-axis (dependent variable): performance measure or attribute and the aggregation function

Tools for performance dashboarding

Operational-Tactical – Business Activity Monitoring (BAM)

- Axway AMPLIFY
- VITRIA Operational Process Intelligence
- Oracle BAM
- SAP Operational Process Intelligence (OPI)

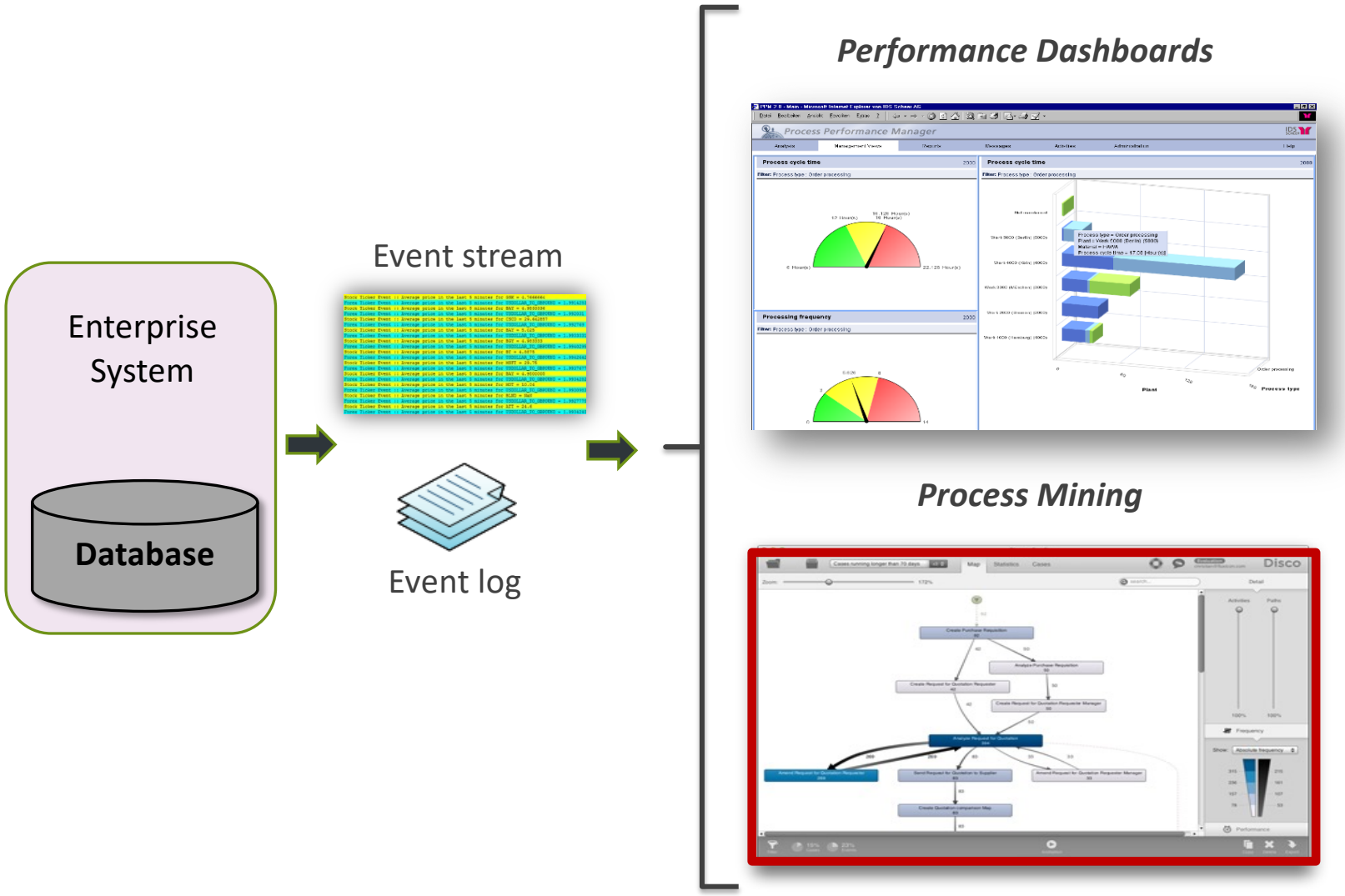
Tactical

- Business Intelligence (BI) tools: PowerBI, Qlikview, Tableau…
- Process Mining (dashboard modules): Apromore, Celonis, QPR, …

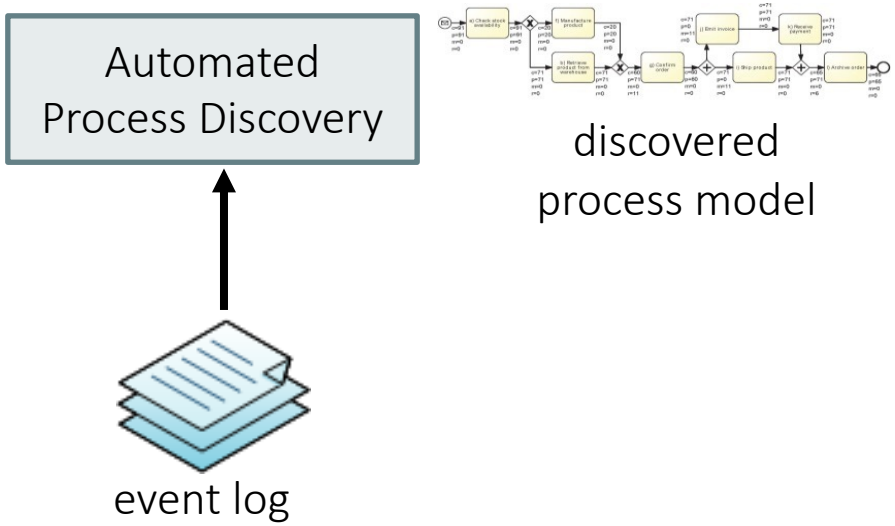
Strategic Level – Balanced Scorecard tools

- BSC Designer Online, Quickscore, Sisense, etc.

Business Process Monitoring



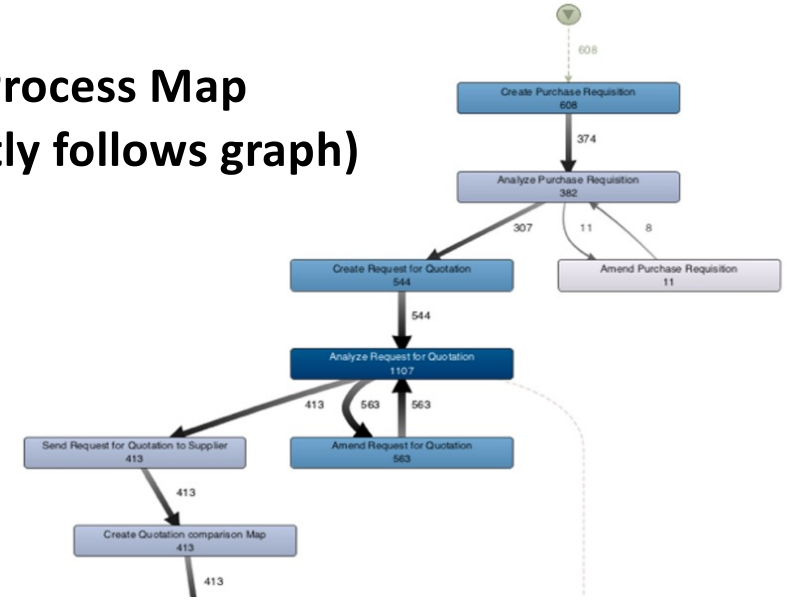
Process Mining



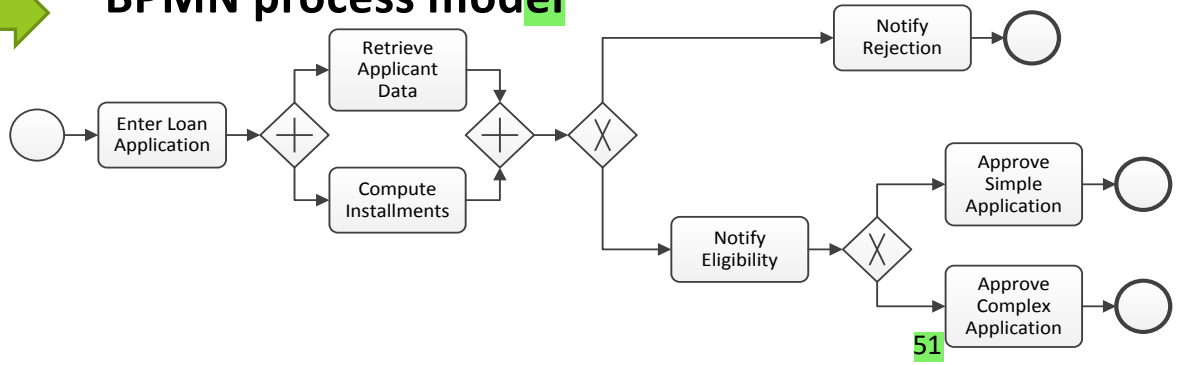
Automated Process Discovery

CID	Task	Time Stamp	...
13219	Enter Loan Application	2007-11-09 T 11:20:10	-
13219	Retrieve Applicant Data	2007-11-09 T 11:22:15	-
13220	Enter Loan Application	2007-11-09 T 11:22:40	-
13219	Compute Installments	2007-11-09 T 11:22:45	-
13219	Notify Eligibility	2007-11-09 T 11:23:00	-
13219	Approve Simple Application	2007-11-09 T 11:24:30	-
13220	Compute Installements	2007-11-09 T 11:24:35	-
...	...	...	...

Process Map
(directly follows graph)



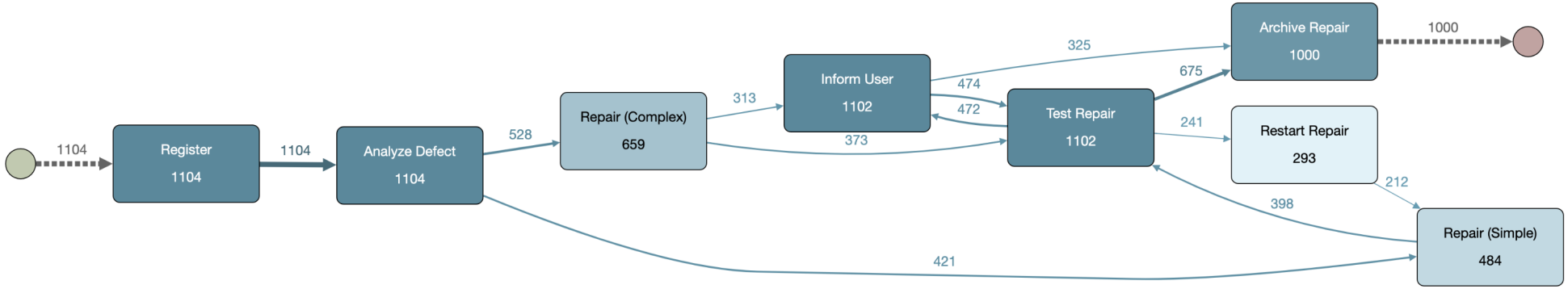
BPMN process model



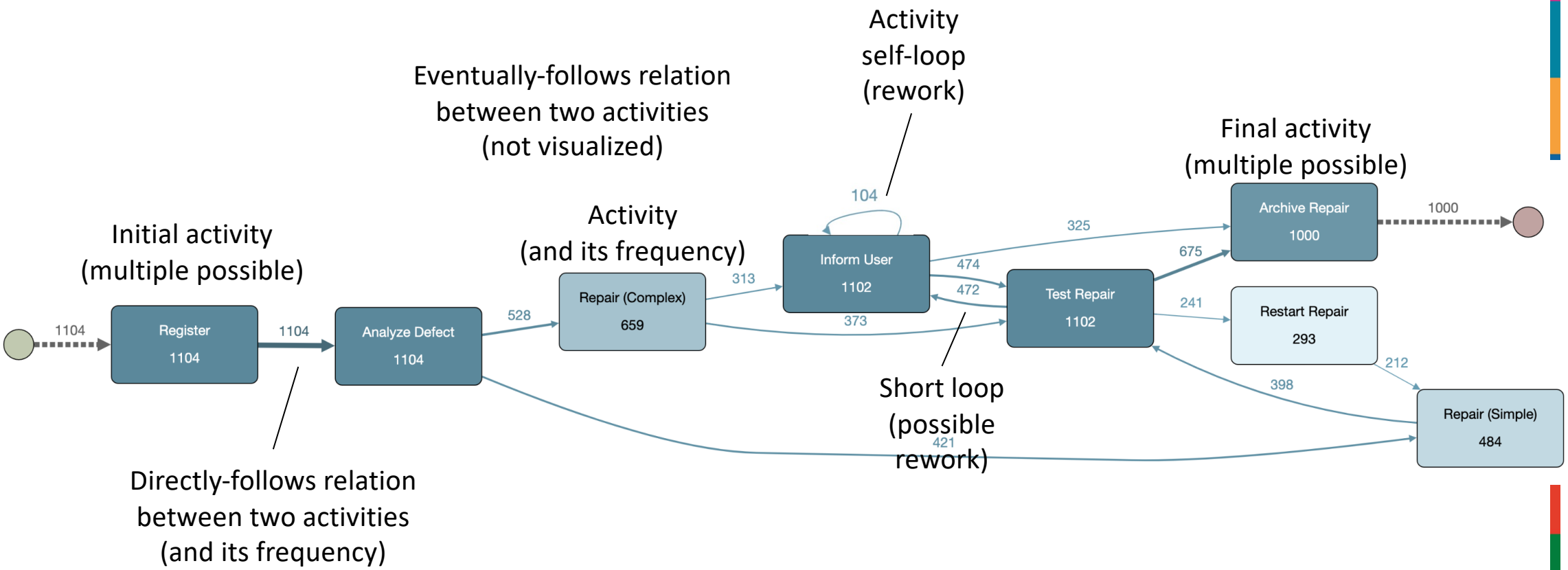
Process maps (aka Directly-follows graphs)

A process map of an event log is a graph where:

- Each activity is represented by one node
- An arc from activity A to activity B means that B is <u>directly followed</u> by A in at least one case in the log



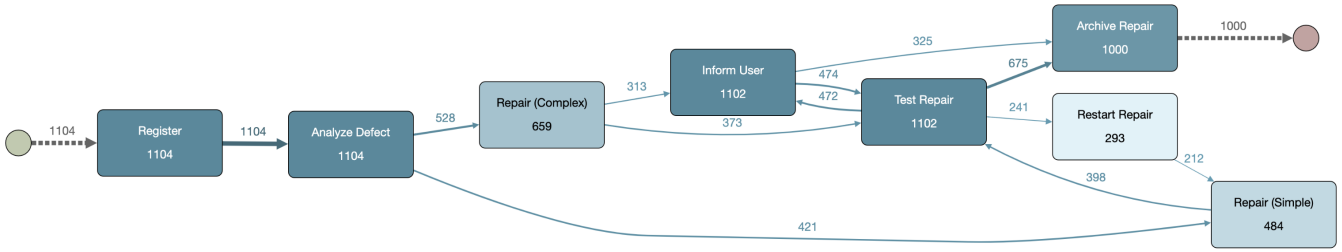
Anatomy of a Process map



Process maps: overlays

Nodes in a process map can be coloured, and arcs' thickness can be changed, to capture:

- **<u>Frequency</u>**: How often a given activity or directly-follows relation occurs?
- **Duration**: processing times and cycle times for activities, waiting times for directly-follow relations
- **Other attributes**: some tools support enhancement by other attributes, e.g., cost, revenue, sales volume, if these data attributes are available



Example: Process Map

with frequency

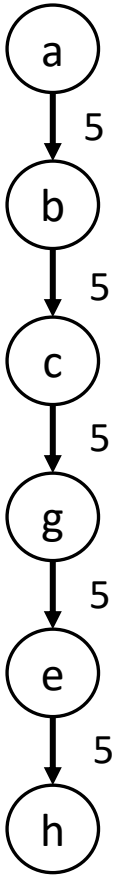
Event log

5: a,b,c,g,e,h

4: a,b,c,f,g,h

3: a,b,d,g,e,h

3: a,b,d,e,g,h



Example: Process Map

with frequency

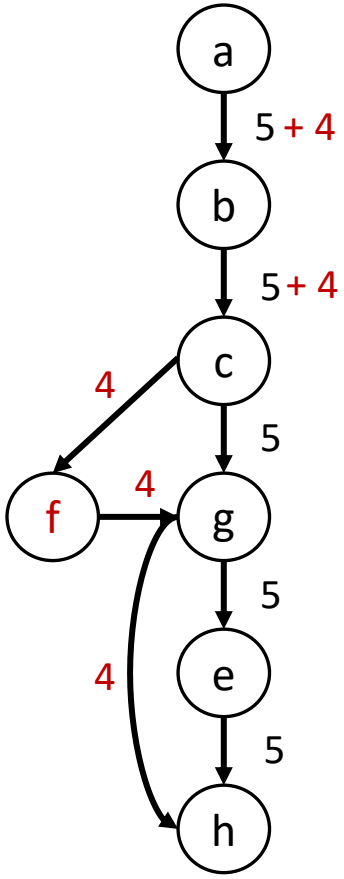
Event log

5: a,b,c,g,e,h

4: a,b,c,f,g,h

3: a,b,d,g,e,h

3: a,b,d,e,g,h



Example: Process Map

with frequency

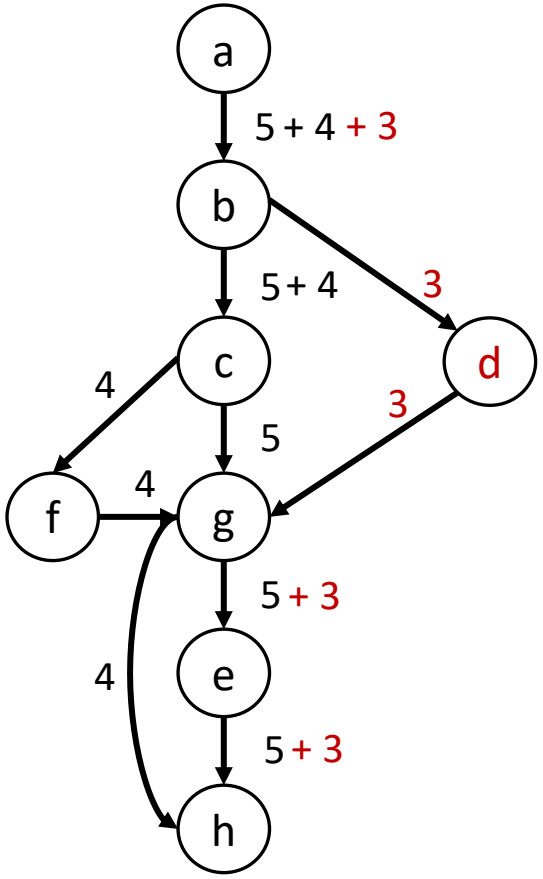
Event log

5: a,b,c,g,e,h

4: a,b,c,f,g,h

3: a,b,d,g,e,h

3: a,b,d,e,g,h



Example: Process Map

with frequency

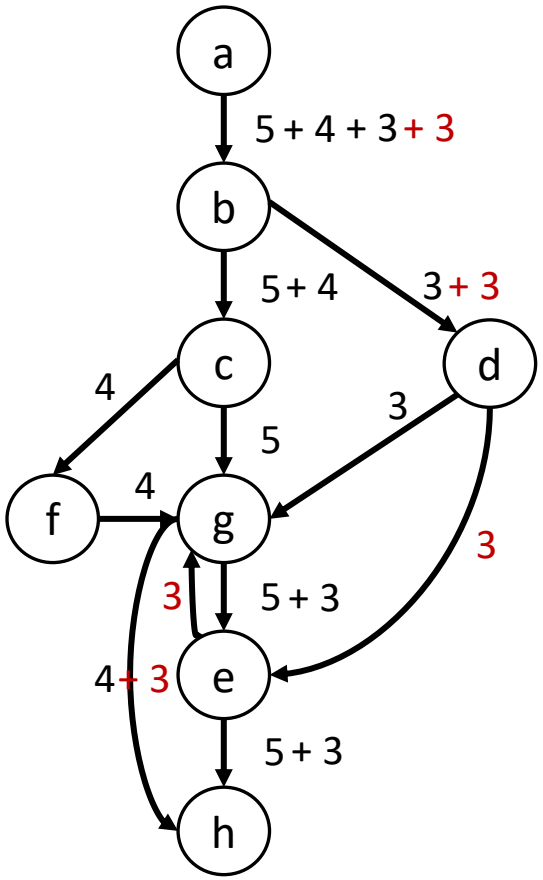
Event log

5: a,b,c,g,e,h

4: a,b,c,f,g,h

3: a,b,d,g,e,h

3: a,b,d,e,g,h



Example: Process Map

with frequency

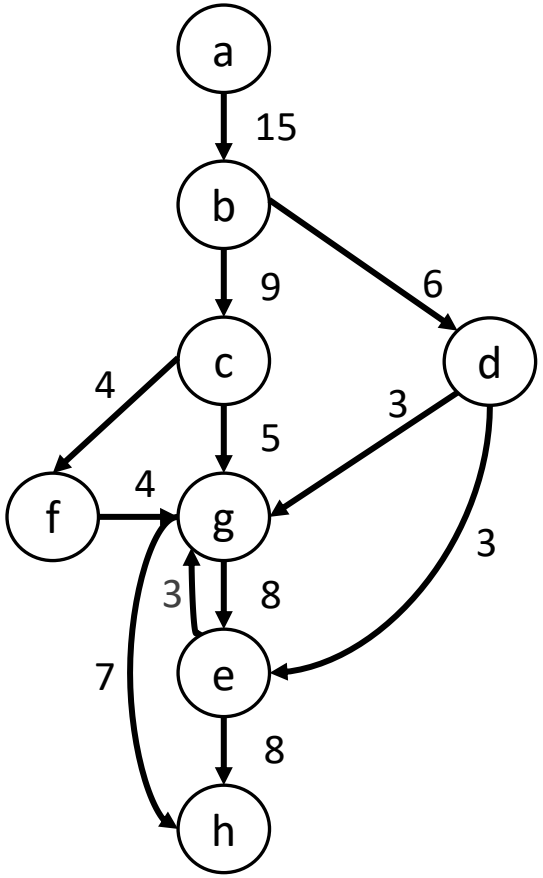
Event log

5: a,b,c,g,e,h

4: a,b,c,f,g,h

3: a,b,d,g,e,h

3: a,b,d,e,g,h

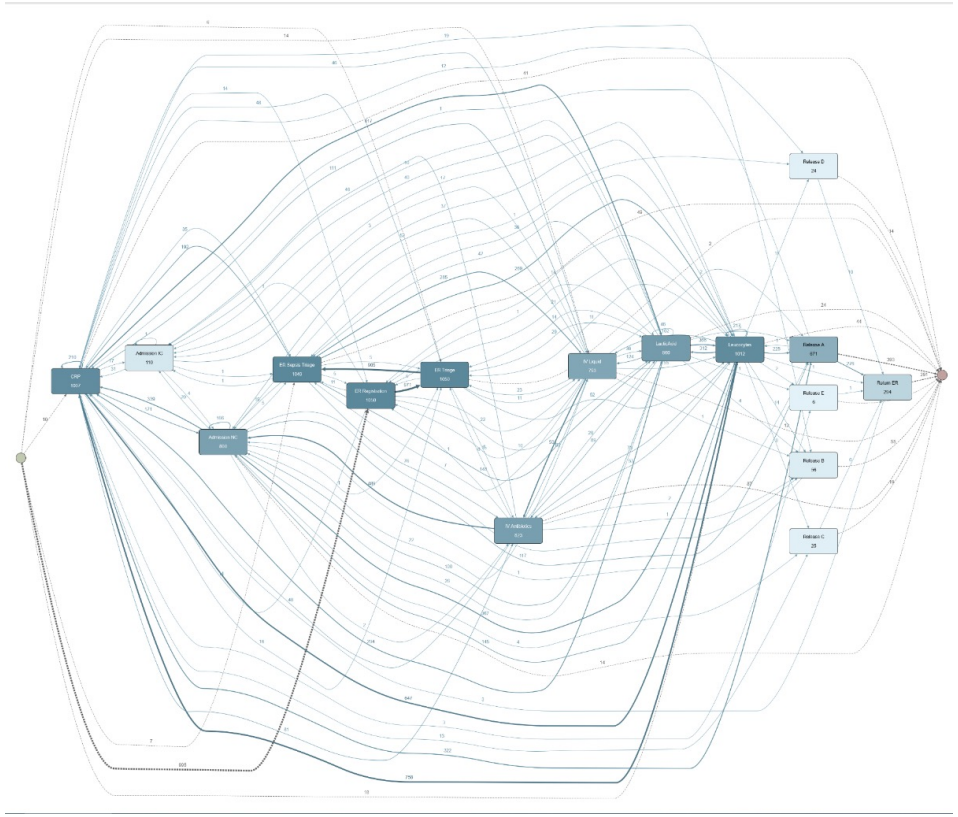


Process Maps

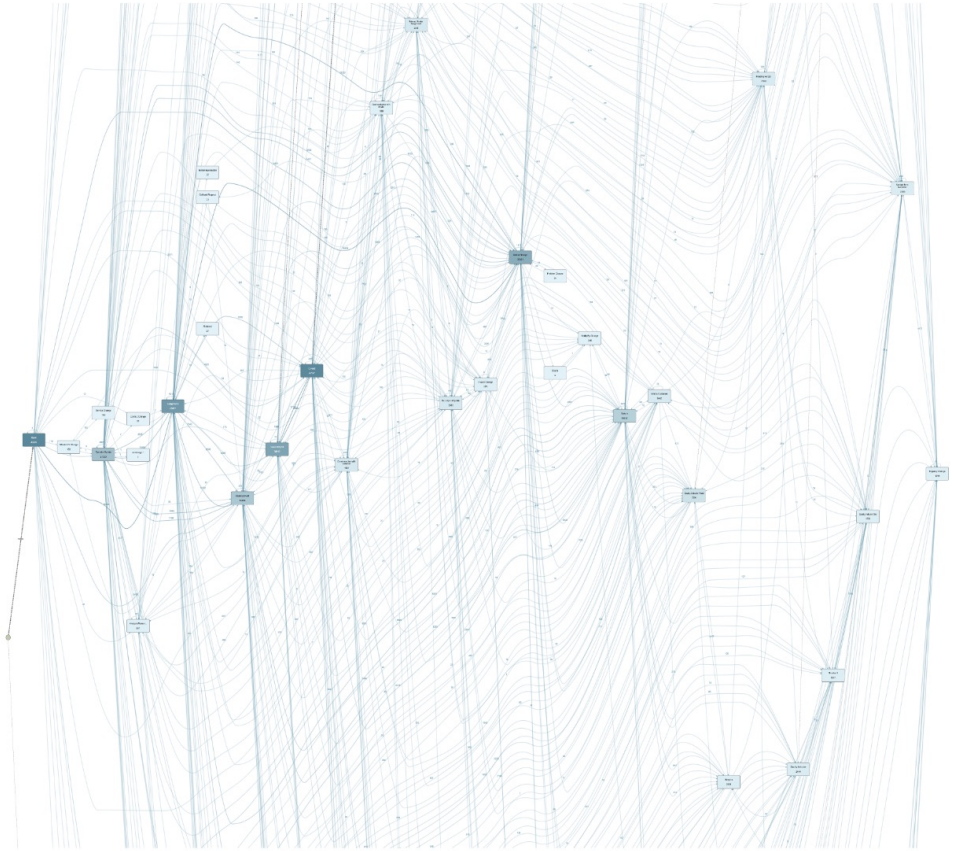
- Commercial process mining tools use process maps as the main visualization technique for event logs
- These tools also provide three types of operations:
  1. **Abstract** the process map:
  2. **Filter** the traces in the event log…

Abstraction and Filtering

- Real-life processes are really messy



Patient Treatment Process @ Hospital (Sepsis infections)

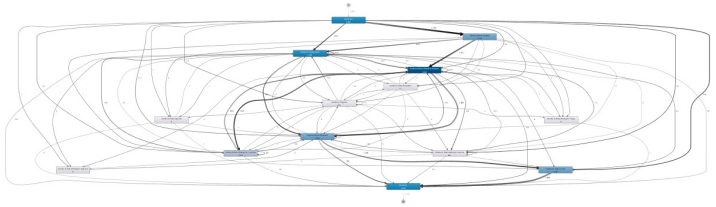


IT Incident Management @ Bank

Process map: Abstraction

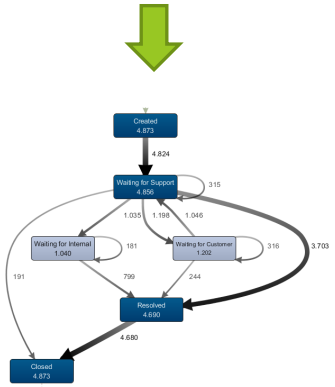
Common process mining tools provide abstraction capabilities on top of process maps, to:

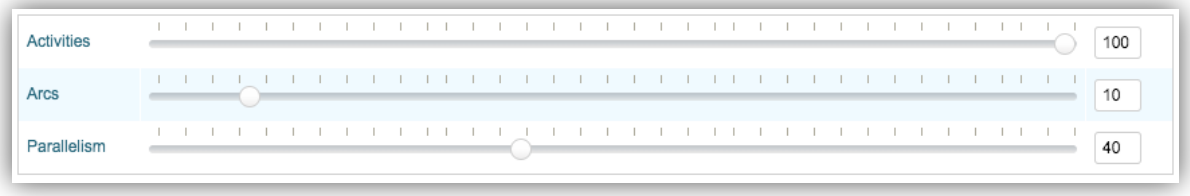
- Show only **most frequent** activities
- Show only **most frequent** arcs



Some tools offer further abstraction capabilities to:

- Show only **least frequent** activities
- Show only **least frequent** arcs
- Show only **fastest/slowest** activities
- Show only **fastest/slowest** arcs





Log filters

- Case-Level Log Filters
  - Retain (or remove) all cases that fulfill a given condition, for example all cases that take more than 3 days
- Event-Level Filter
  - Retain or remove individual events (*activity instances*) that fulfill a given condition, e.g., all activity instances that were performed by “John Smith”
- A case-level filter either retains an entire case or it removes the whole case while an event-level filter may remove some events in a case and remove others.

Case-Level log filters

Attribute filters

- Retain only cases such that at least one event fulfils a condition over one or more attributes, for example: *at least one event where Activity = "Close order" and Resource = "Anti Alman"*

Timeframe filter

- Retain traces that start, end, or are contained in a time period (e.g. all cases that started in January)

Performance filter

- Retain only cases that have a duration above or below a given value (e.g. 3 days)

Path filter

- Retain traces where there is a pair of events that fulfil a given condition (e.g. "Create invoice" eventually followed by "Create purchase order")

Repetition filter

- Retain traces that start with or finish with an event that fulfils a given condition

Event-Level log filters

Attribute filters

- Retain only events that fulfil a condition, for example: *at least one event where Activity = "Close order"*

Timeframe filter

- Retain all events that happen during a certain time period (e.g. all events in January)

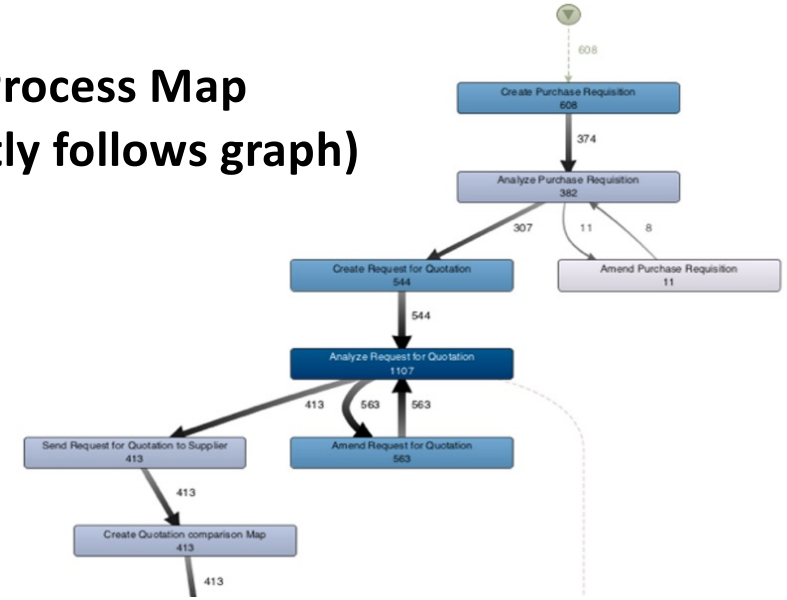
Between filter

- For each case, retain all events between the (first or last) occurrence of an activity and the (first or last occurrence) of another activity (e.g. between "Create PO" and "Approve PO")

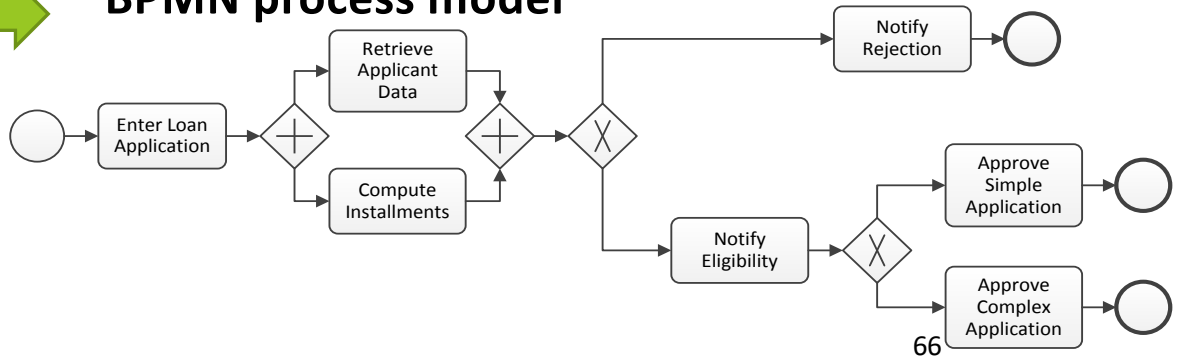
Automated Process Discovery

CID	Task	Time Stamp	...
13219	Enter Loan Application	2007-11-09 T 11:20:10	-
13219	Retrieve Applicant Data	2007-11-09 T 11:22:15	-
13220	Enter Loan Application	2007-11-09 T 11:22:40	-
13219	Compute Installments	2007-11-09 T 11:22:45	-
13219	Notify Eligibility	2007-11-09 T 11:23:00	-
13219	Approve Simple Application	2007-11-09 T 11:24:30	-
13220	Compute Installements	2007-11-09 T 11:24:35	-
...	...	...	...

Process Map
(directly follows graph)



BPMN process model



Limitations of process maps

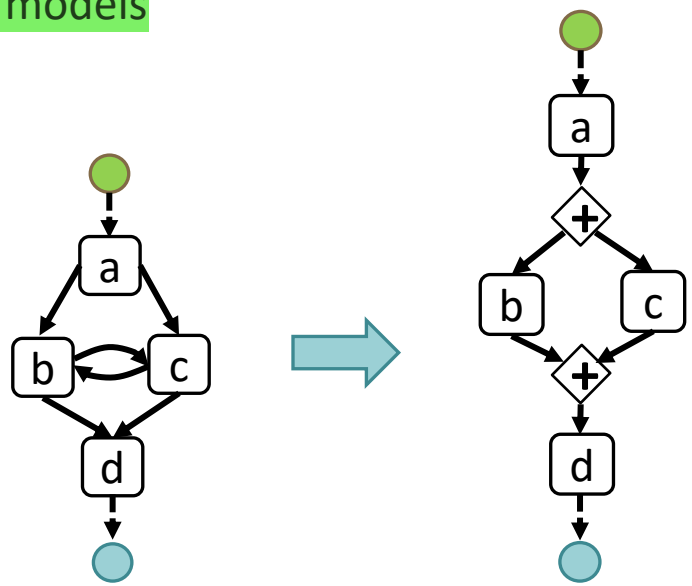
Process maps over-generalize: some paths of a process map might actually not exist and might not make sense

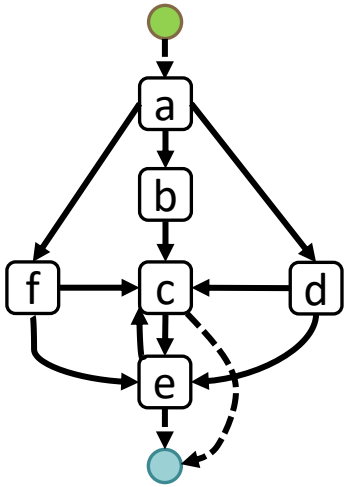
- Example: Draw the process map of [abc, adc, afce, afec] and check which cases it can recognize for which there is no support in the event log.

Process maps make it difficult to distinguish conditional branching, parallelism and loops

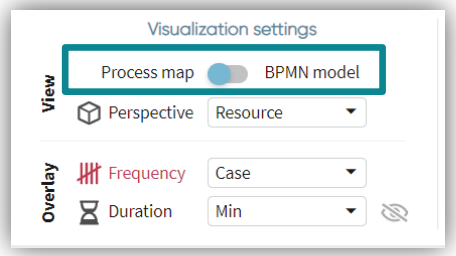
- See previous example… or a simpler one: [abcd, acbd]

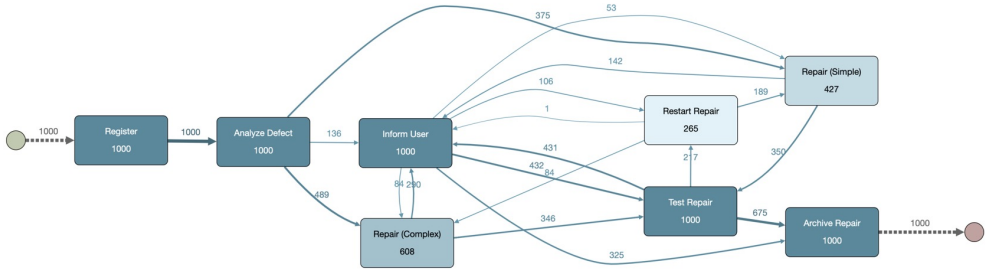
Solution: automated discovery of BPMN models



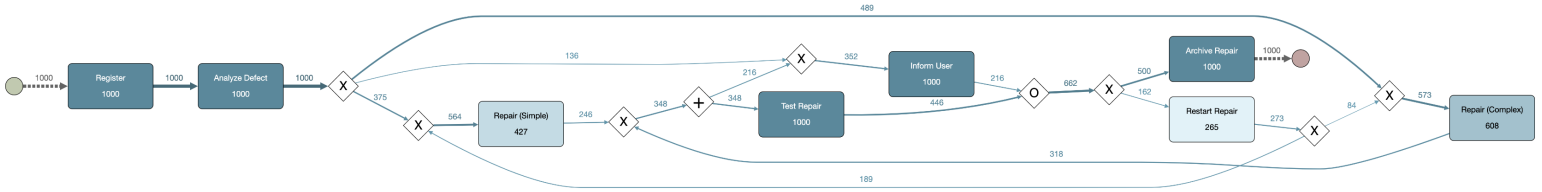


Discovery of BPMN models



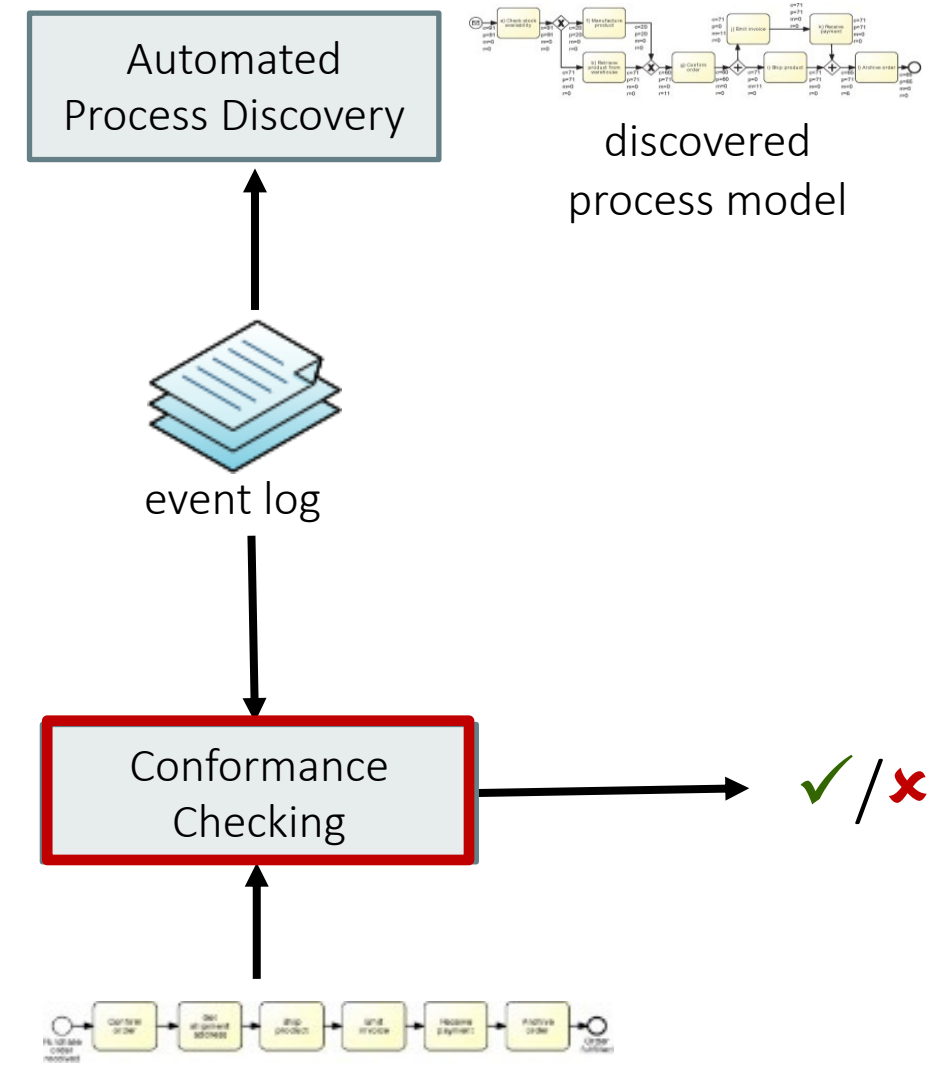


A given configuration of node/arc sliders, and parallelism slider

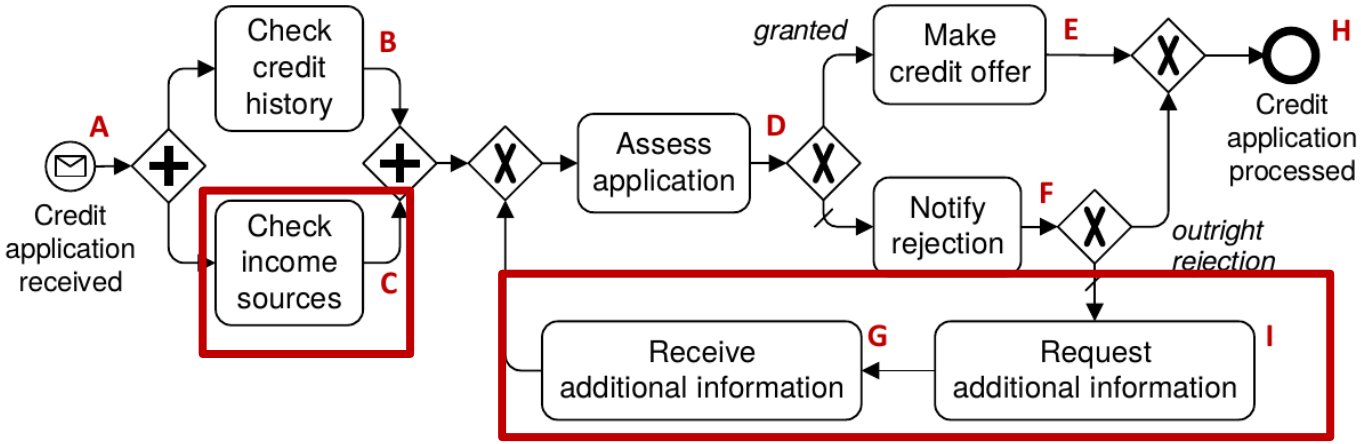


*Note: Apromore uses an algorithm called **Split Miner** to turn process maps into process models*

Process Mining



Conformance Checking (model-based)



Event log:

ABCDEH

ACBDEH

ABCDFH

ACBDFH

ABDEH

ABDFH

Unfitting behaviour:

- Task C is *optional* (i.e. may be skipped) in the log

Additional behavior:

- The *cycle* including IGDF is not observed in the log

Conformance checking (rule-based)

- Oftentimes, we don’t have a full process model available
- Instead, we have a set of compliance rules, e.g.
  - Certain tasks MUST be executed in every case
  - Certain tasks can be executed at most once
  - When a task A occurs, another task B must also occur (within e.g. 2 hours)
  - Two tasks A and B CANNOT be executed by the same person (four-eyes principle)

Conformance Checking Levels

Accordingly, we can identify the following *compliance checking levels*:

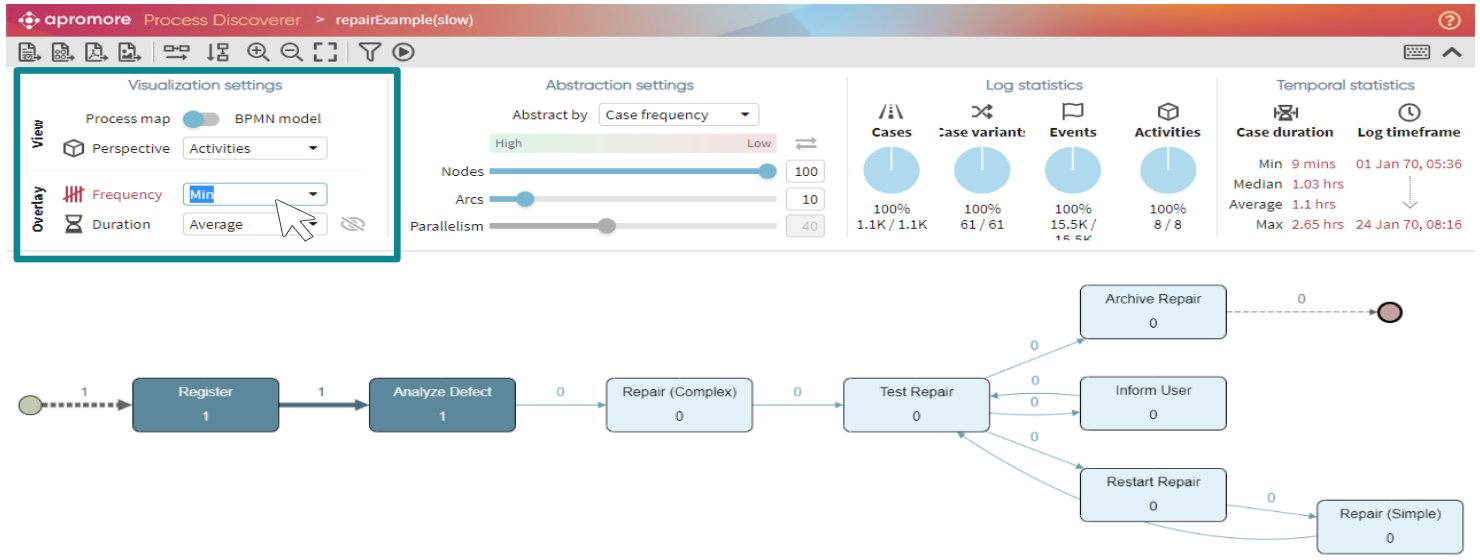
- **Flow:** have all mandatory activities been performed? Have activities been performed in the right order and with the right frequency?
- **Temporal:** has a given timeframe between activities, process milestones or for the end-to-end process been adhered to?
- **Resource:** did everyone obey the segregation of duties?
- **Exception:** explore highly-infrequent behavior to identify possible deviations from the standard process

Procedure:

1. Define compliance rules (based on attribute-value pairs, activity orders or performance targets)
2. Apply one or more filters to keep cases that violate the rules in question
3. If needed, switch perspective to highlight non-compliance issues (e.g. show resources that violate a segregation of duties principle)
4. Assess impact on process performance

Flow Compliance Checking

1 Mandatory tasks: Have all mandatory tasks been completed?

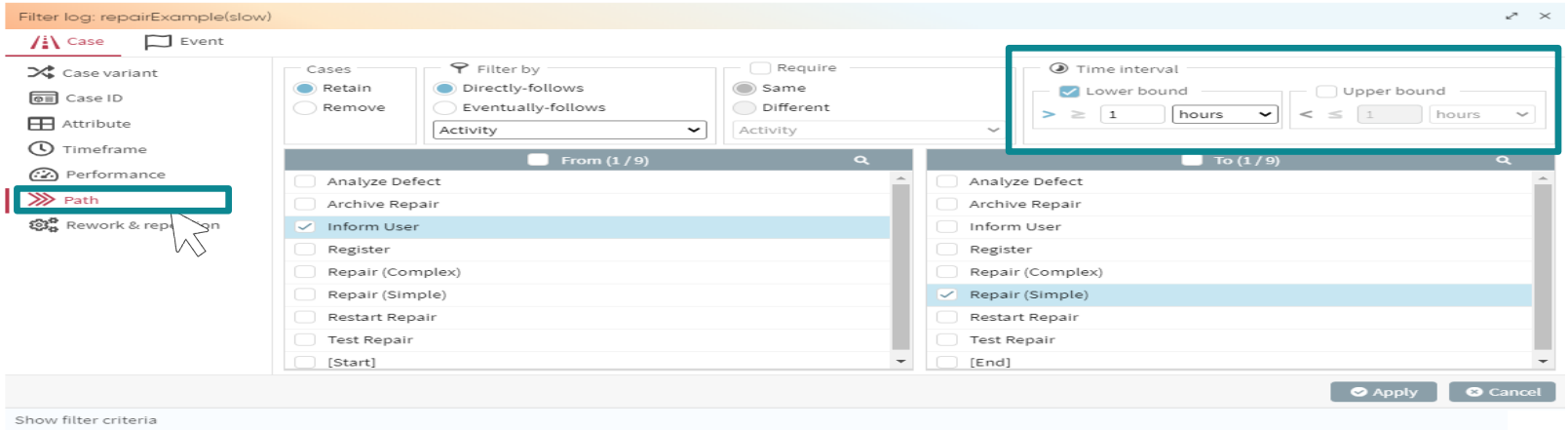


To check whether all your tasks have been executed at least once, change the frequency view to “Min.” in the visualization settings of your Process Discoverer.

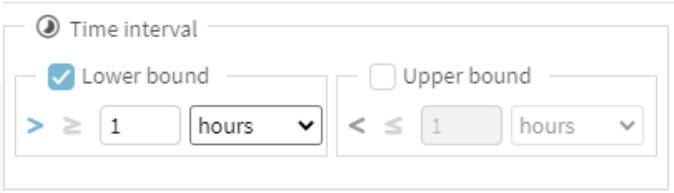
The numbers in your activities now indicate the minimum number of times it has been completed in all process instances.

Temporal Compliance Checking

2 Prescribed time frame: Has the given time frame been adhered to?



Select the “Path” filter and define the steps that must be performed in a defined time frame. Use “Time interval” to then set your temporal figures.



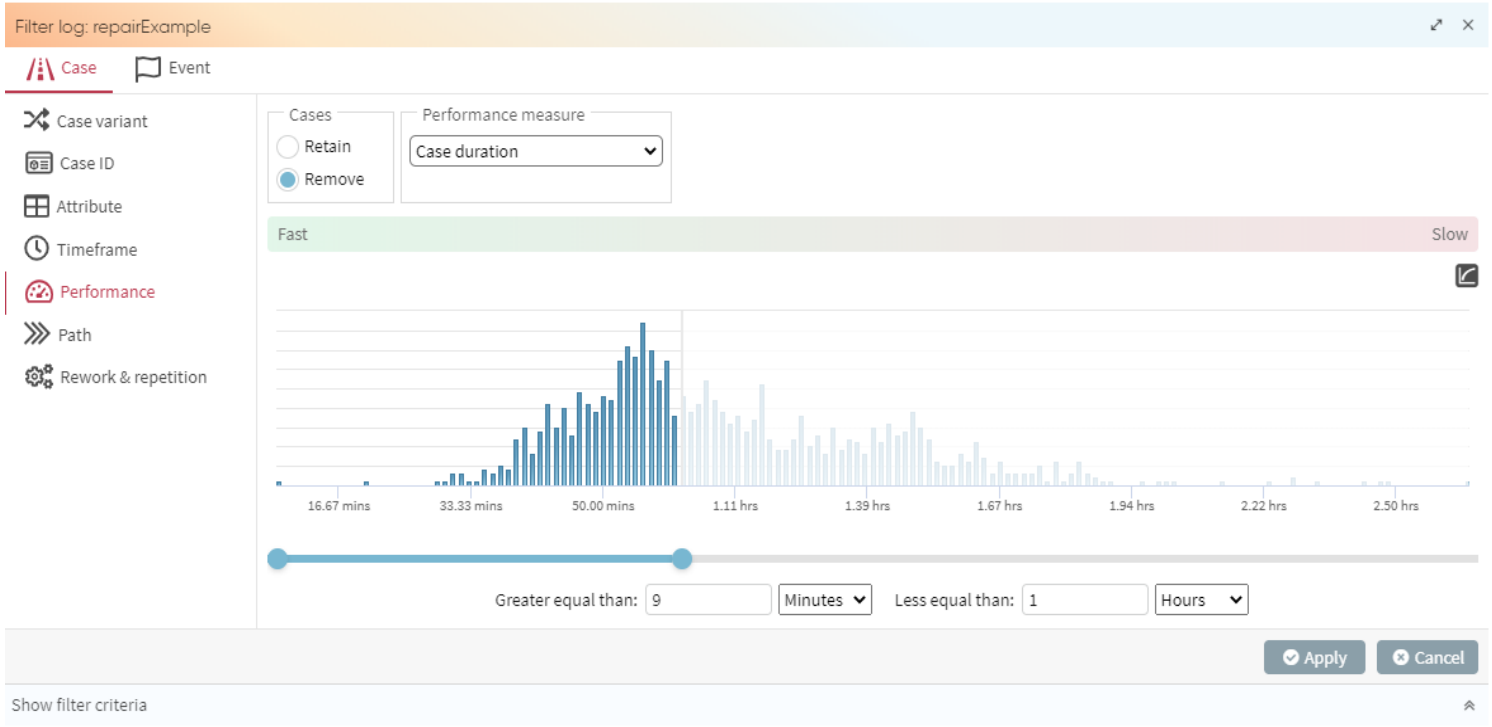
Example:

Compliance rule:
Waiting time between “Inform User” and “Repair (Simple)” should not exceed 1 hours.

“Time Interval“ Filter:
> 1 hour, so that only those cases are retained that do not comply with the compliance rule.

Temporal Compliance Checking

1 Prescribed time frame: Has the given time frame been adhered to?

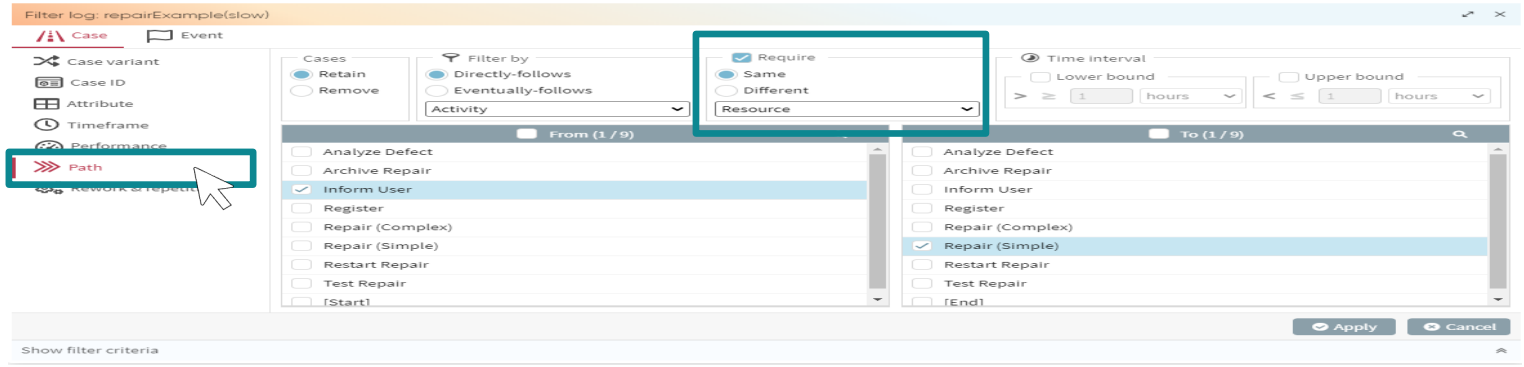


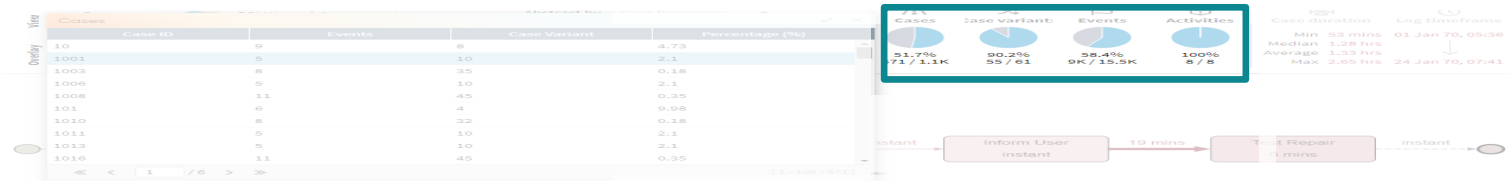
Performance filter can help to check for end-to-end SLA violations.

For example, retain only those cases that gets completed in at most 1 hour.

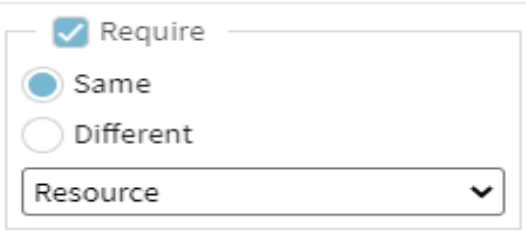
Resource Compliance Checking

1 Segregation of duties: Did everyone obey the segregation of duties?



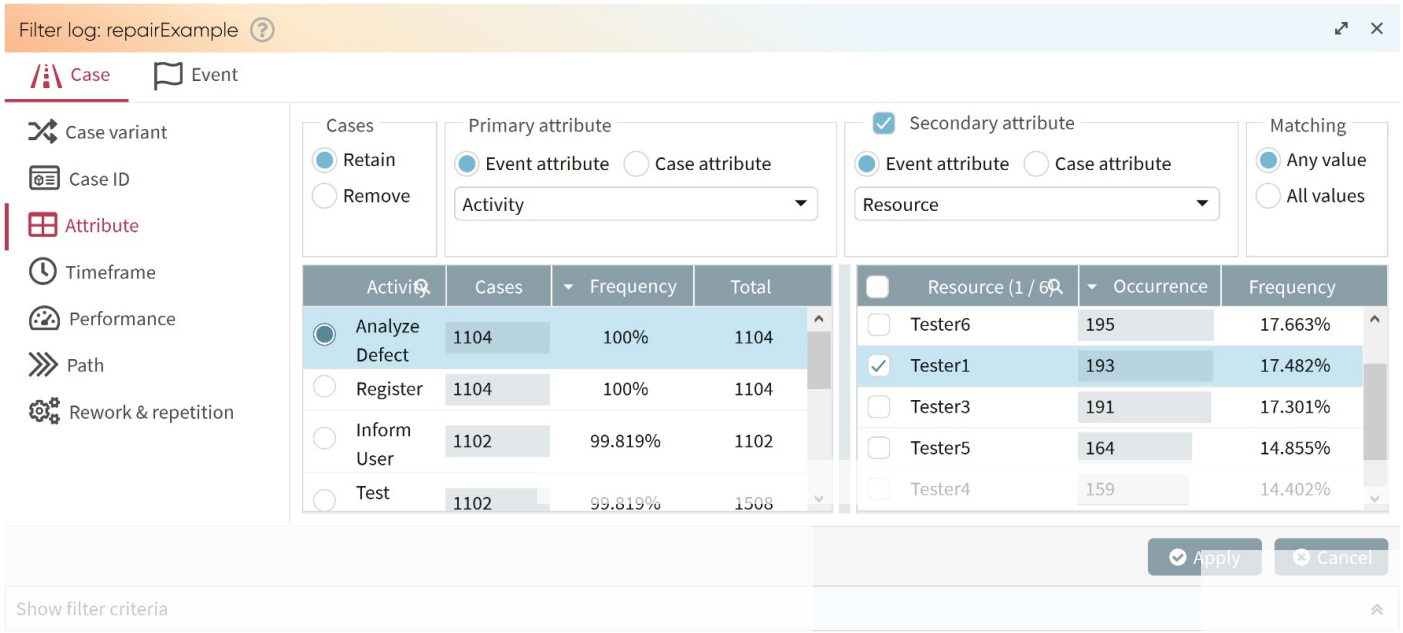


To check whether the segregation of duties has been breached, select a “Path” filter. Now, set another filter which indicates that the same resource must have completed the tasks (e.g. creating & approving an invoice)



Resource Compliance Checking

1 Unexpected resource assignments

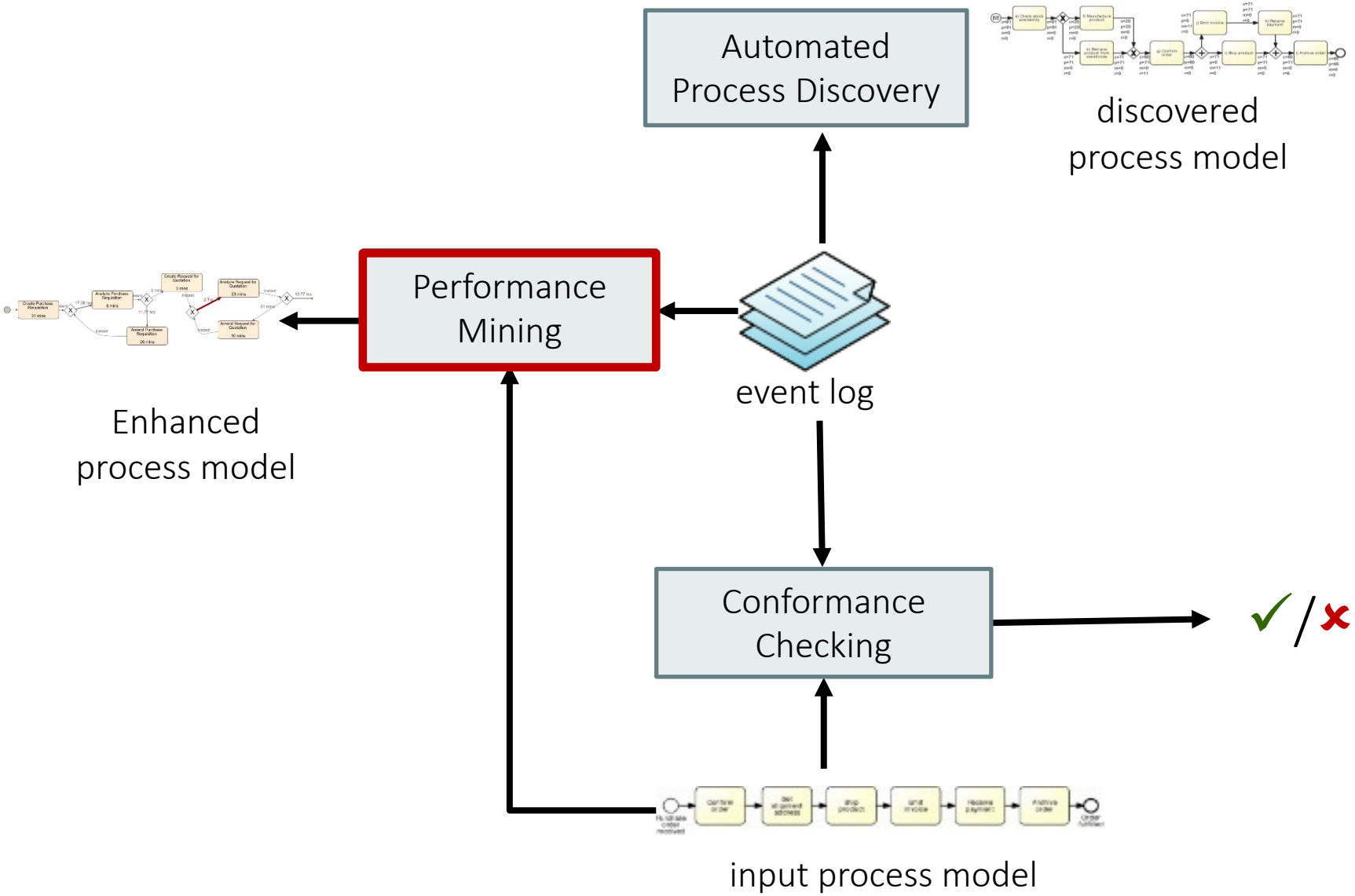


User the Attribute filter with two attributes (primary and secondary) to retrieve cases where a given activity was performed by a given resource (or any other combination of attributes)

Conformance Checking: Analysis Template

What?	How?
Flow compliance checking 1. Check skipped mandatory activities 2. Check forbidden repetitions. 3. Check activity exclusion or co-occurrence relations	Identify skipped activities in the process map or BPMN model Visualize process map with the min/max frequency metrics Visualize min/max statistics in Activity tab of the dashboard (EE) Use Path filter (directly-follows or eventually-follows)
Temporal compliance checking Check temporal constraints violations	Use performance filter (to check for end-to-end SLA violations) Use Path filter (eventually-follows) with a duration constraint to identify violations of temporal constraints between two activities
Resource compliance checking Check four-eyes principle or same-case-hander principle Check if an activity has been performed by a resource who should not do so.	Use Path filter (eventually-follows) and require the same Resource (or different resources) Use the Attribute filter and search with “Activity” as the primary attribute and “Resource” as secondary attribute to check if a resource performed an activity they were not supposed to do.

Process Mining



Process performance mining: Analytics & Analysis types

Extract process execution performance in an aggregate or detailed level (i.e. at the level of individual activities and directly-follows relations)

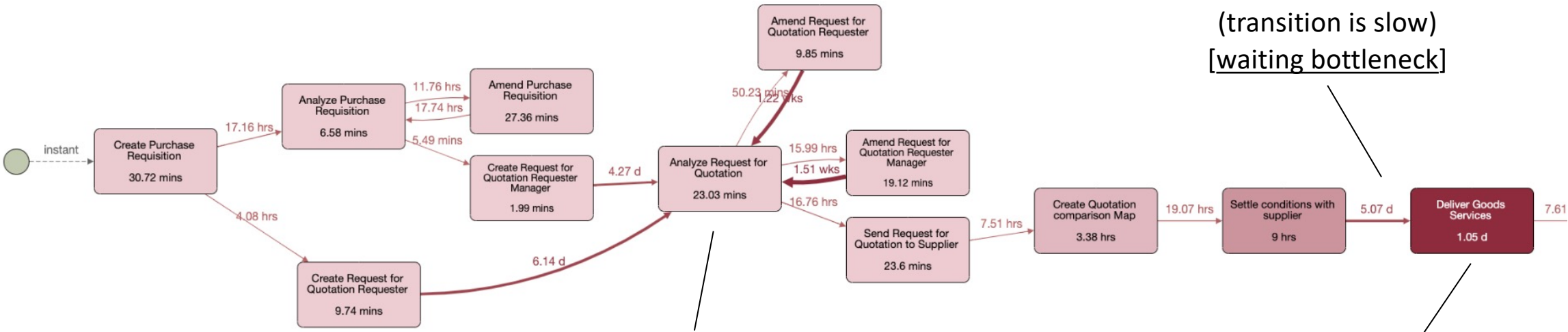
Analytics:

- Duration-enhanced process maps (activity, resource perspective…)
- Duration-enhanced BPMN models
- Process performance dashboards

Analysis types:

- Bottleneck analysis
- Workload analysis
- Rework analysis

Bottleneck analysis in the Activity perspective

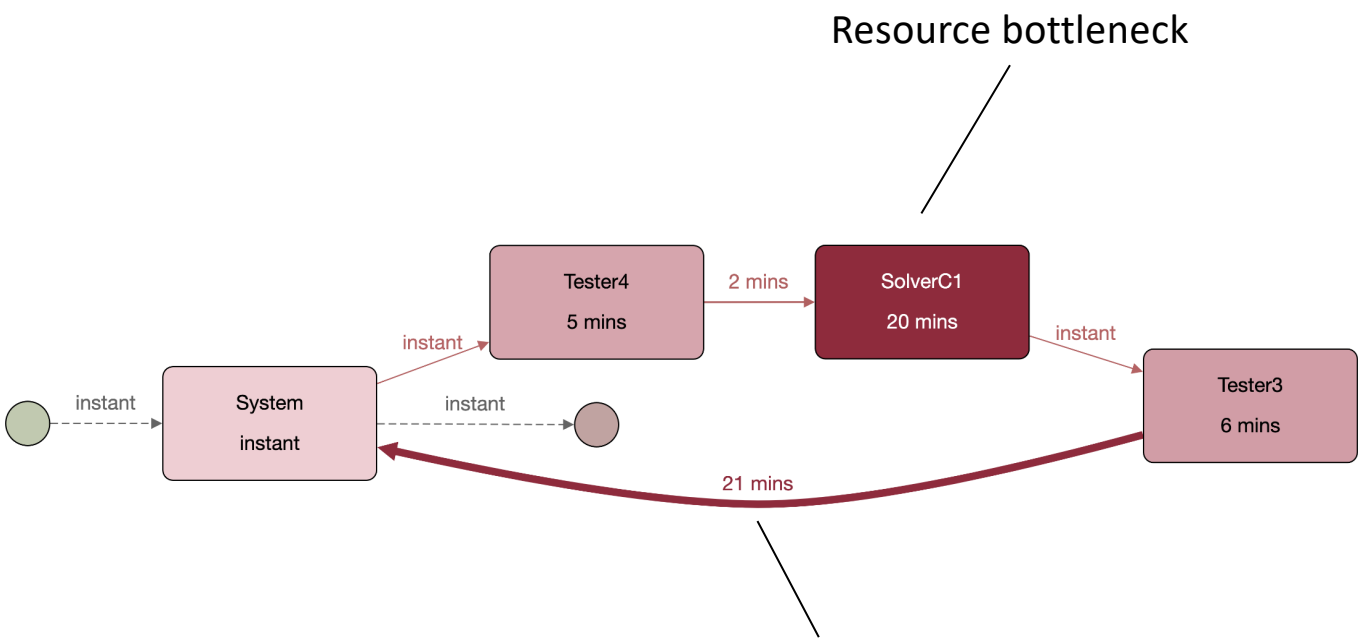


Slow handoff between activities (transition is slow) [waiting bottleneck]

Resource-capacity bottleneck (all incoming arcs are slow) [waiting bottleneck]

Activity bottleneck (activity is slow)

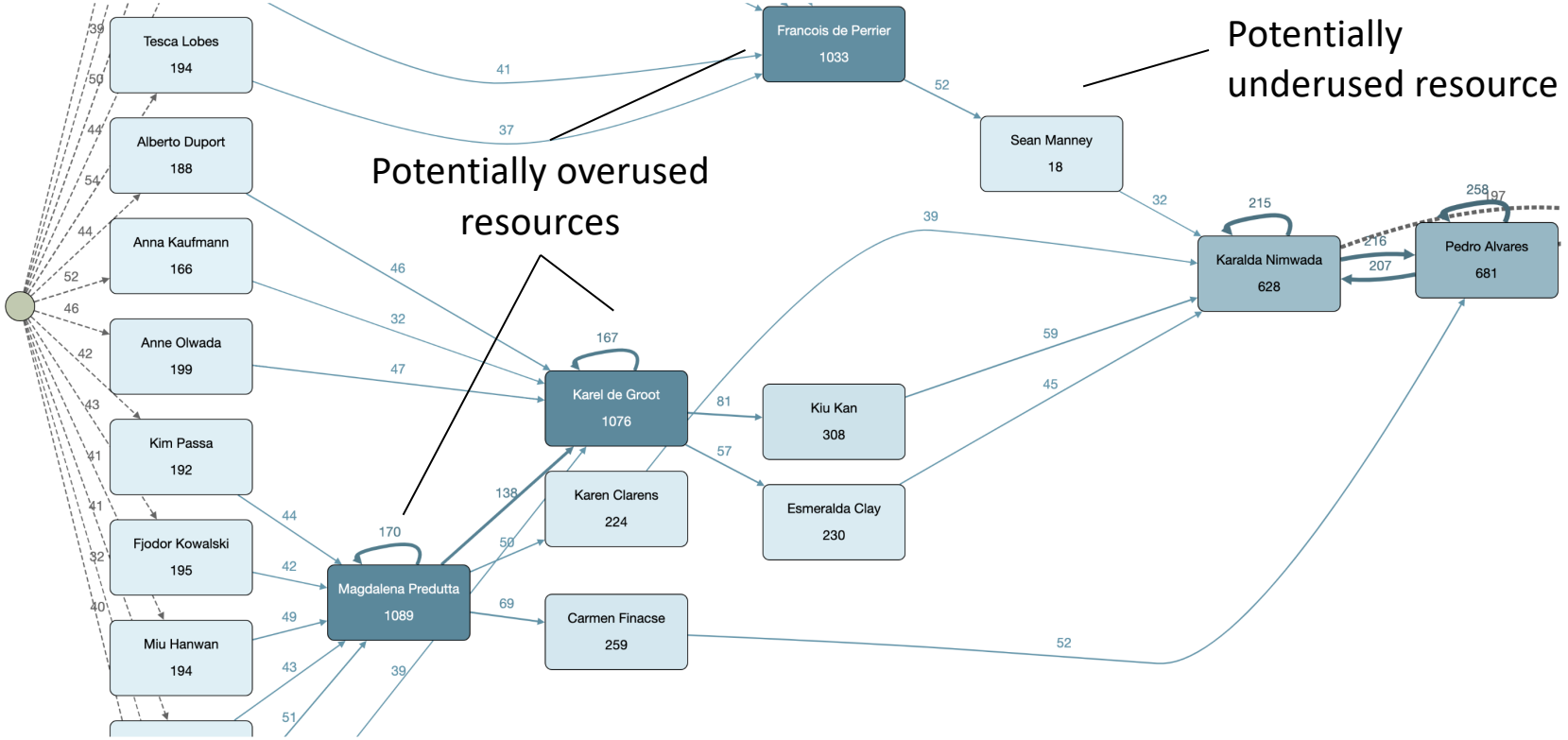
Bottleneck analysis in the Resource perspective



Slow handoff
between
specific resources
[waiting bottleneck]

Workload analysis: overused vs underused resources

- **Potentially overused resources** are those with high total frequency and/or <u>high total duration</u>
- **Potentially underused resources** are those with low total frequency and/or <u>low total duration</u>



Rework analysis

Rework means repetition of activities within the same case:

- **Self loop:** an activity takes place n times in a row
- **Short loop ("ping-pong effect"):** two activities are repeated one after the other. Note: this may be confused with *parallelism*
- **Indirect repetition (loop):** several activities are executed several times in sequence.

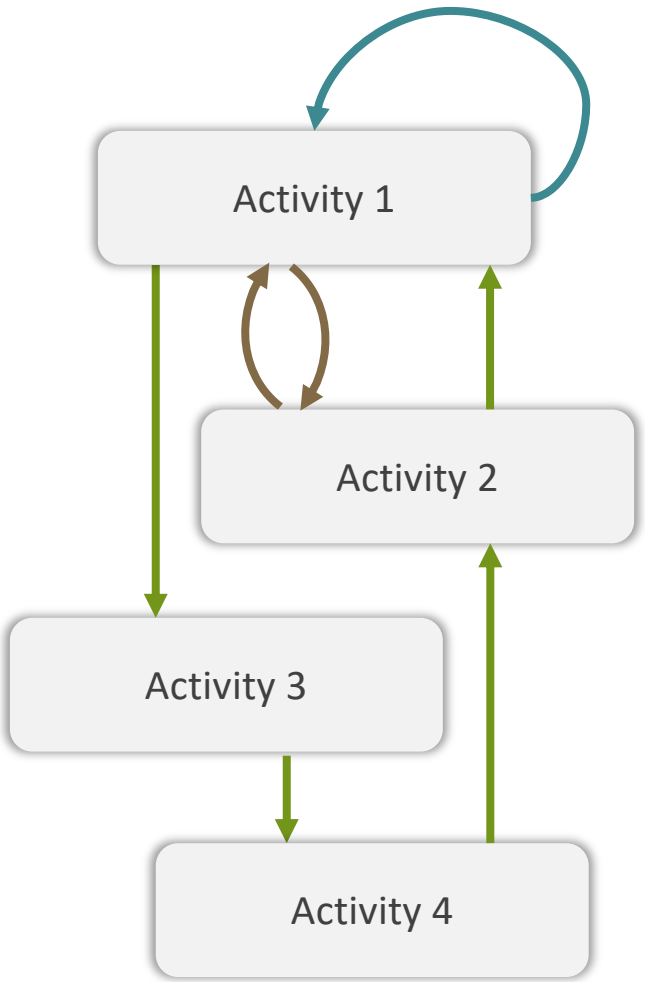
To analyze repetitions, one can answer the following questions:

(1) What type of repetition is present?

(2) How often is an activity / series of activities repeated?

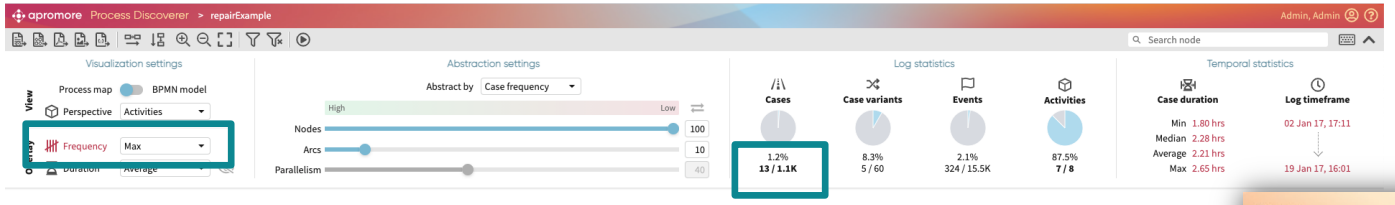
(3) How many cases showcase this repetition? How often does it occur?

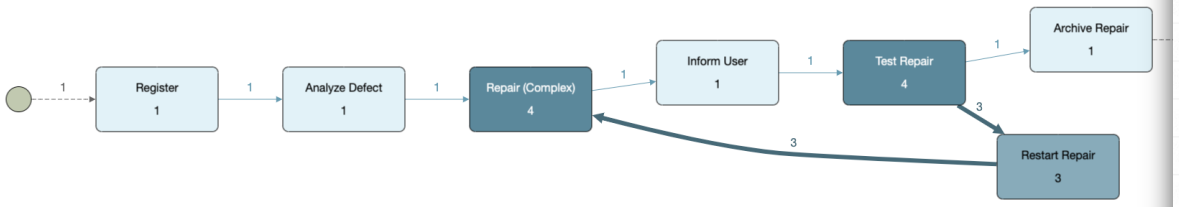
(4) Which cases are affected exactly? And how do they differ from cases where there is no repetition?



Rework analysis

To **assess the impact of rework on process performance (time and cost)**, we should apply filtering to isolate cases that are affected by rework.





Cases with Repair (Complex) ≥ 3

Cases

Case ID	Events	Case Variant	Percentage (%)
165	12	30	30.77
17	12	30	30.77
205	12	30	30.77
372	12	40	15.38
446	12	34	23.08
500	12	30	30.77
501	15	43	15.38
551	15	43	15.38
670	12	40	15.38
682	12	39	15.38

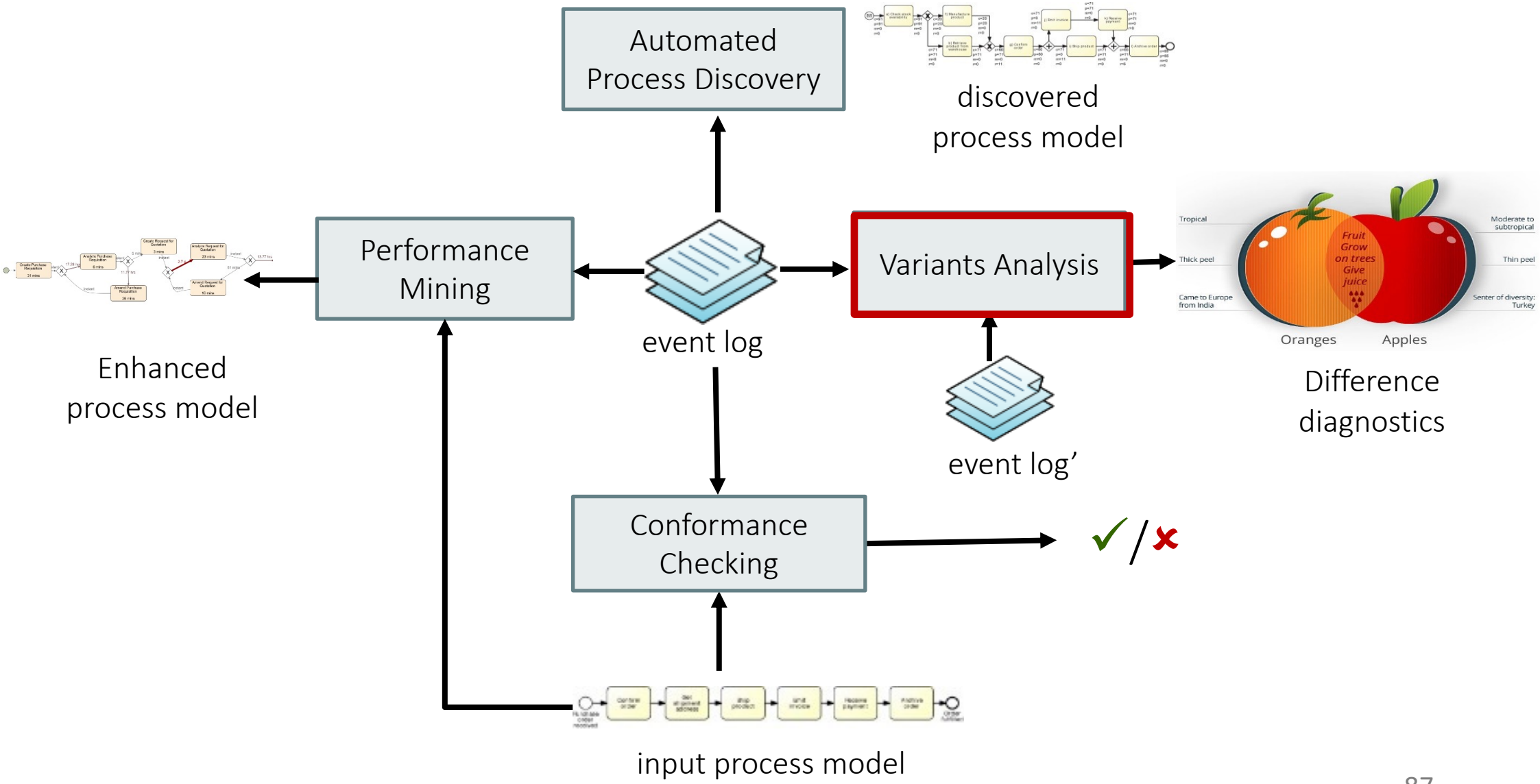
Click on a case to inspect it

Download

Performance Mining: Analysis Template

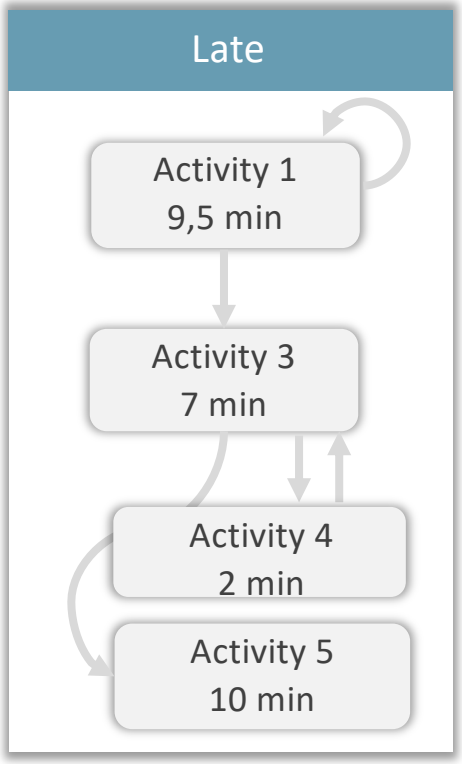
What?	How?
Bottleneck analysis • Identify activity bottlenecks (high-effort activities) • Identify waiting bottlenecks (slow handoffs and capacity bottlenecks) • Identify resource bottlenecks	• In the process map, select the duration overlay (average or total duration): activity bottlenecks have long processing time; slow handoffs between activities are arcs with long waiting times; resource-capacity bottlenecks relate to activities with all/most incoming arcs with long waiting times • Switch to the Resources perspective in the map: slow handoffs between resources are arcs with long waiting times; resource bottlenecks are resources with long avg duration (slow resources) • Use slider with Average duration to focus on slowest or fastest nodes/arcs • Use Activities tab in Dashboard to visualize activity processing times • Use Animation plugin to visualize the build-up of bottlenecks.
Workload analysis • Identify overused vs underused resources/idleness waste • Analyze resource workload	• In the Resources perspective: overused resources have high total frequency or high "total duration"; underused resources have low total frequency and low total duration • Use also the Resources tab in Dashboard to visualize resource workload distribution • Use Filtering to focus on cases where the resource(s) with the highest workload are involved.
Rework analysis Identify defects/errors and rework waste	• Use the Max frequency overlay in the process map to visualize the max amount of repetitions of each activity / transition • Use the Case > Rework & repetition filter to retain cases with activity repetition; use the Case > Path filter with eventually-follows relation between an activity and itself to isolate cases with specific loops • Use the chart in the Activities Dashboard to visualize the distribution of max activity frequencies per case.

Process Mining



Variant analysis

Given two or more logs representing variants of the same process, find the *differences* and *root causes* for variation or deviance between the logs.



The comparison of process variants offers insights into the the "why": Why do certain process cases differ from others?

- Why some take longer than others? (fast vs. slow cases)
- Why some end successfully, while others don't? (positive vs. negative outcomes)
- Why is employee XY more efficient and more successful when it comes to their activities than others? (resource XY vs. remaining resources)

Variant analysis: drivers

Performance

Identify and compare variants based on performance measures and their targets (e.g. slow vs fast cases)

Logical

Identify and compare variants based on case attributes (e.g. product type, customer segment, geographic area, loan amount, claim outcome)

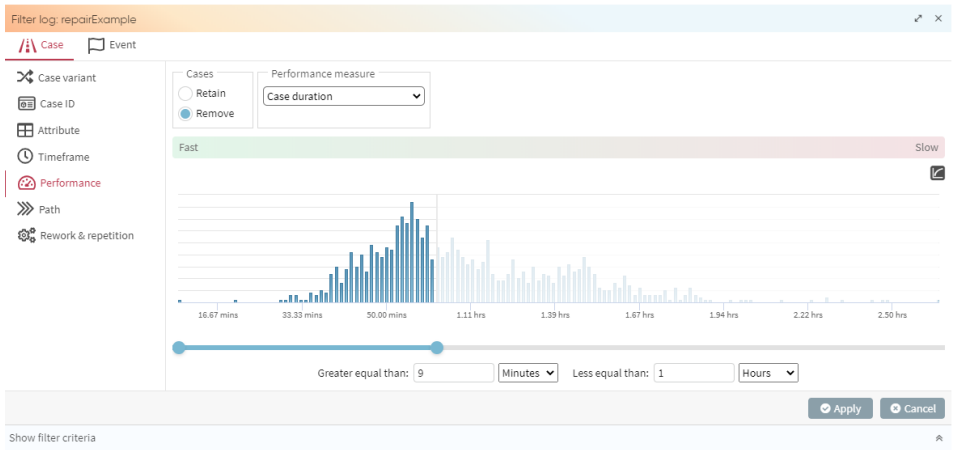
Temporal

Identify and compare variants based on different log timeframes (e.g. before and during COVID)

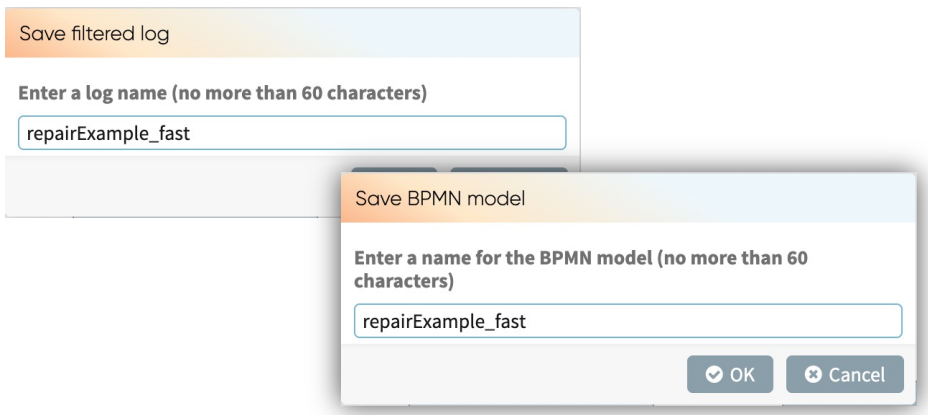
Variant analysis: approach

1. Apply the same filter one per variant, to isolate the respective sublog:
   - **Performance:** use Case > Performance filter
   - **Logical:** use Case > Attribute filter
   - **Temporal:** use Case > Timeframe filter
2. Save filtered process map and BPMN model back to the repository
3. Analyze the variants using one or more analysis techniques.

Apply filter



Save filtered log & BPMN model



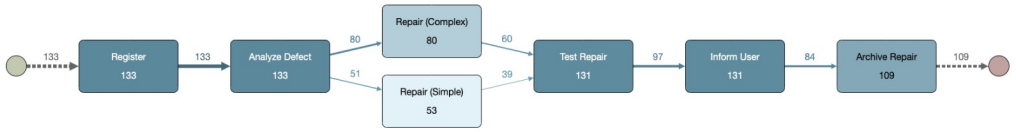
Variant analysis: comparison via maps/models

Visual comparison of process maps/BPMN models, to identify structural differences:

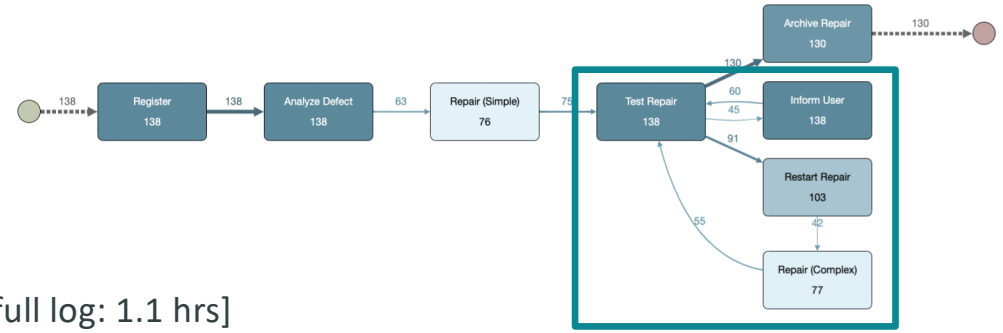
- Flow
- Frequency & rework
- Bottlenecks
- etc.

Consider different perspectives (Activities, Resources…)

Variant 1: "fast" (up to 45 min)



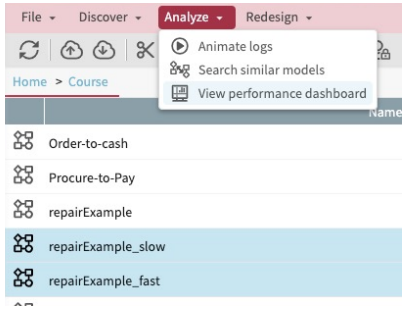
Variant 2: "slow" (more than 1.5 hrs)

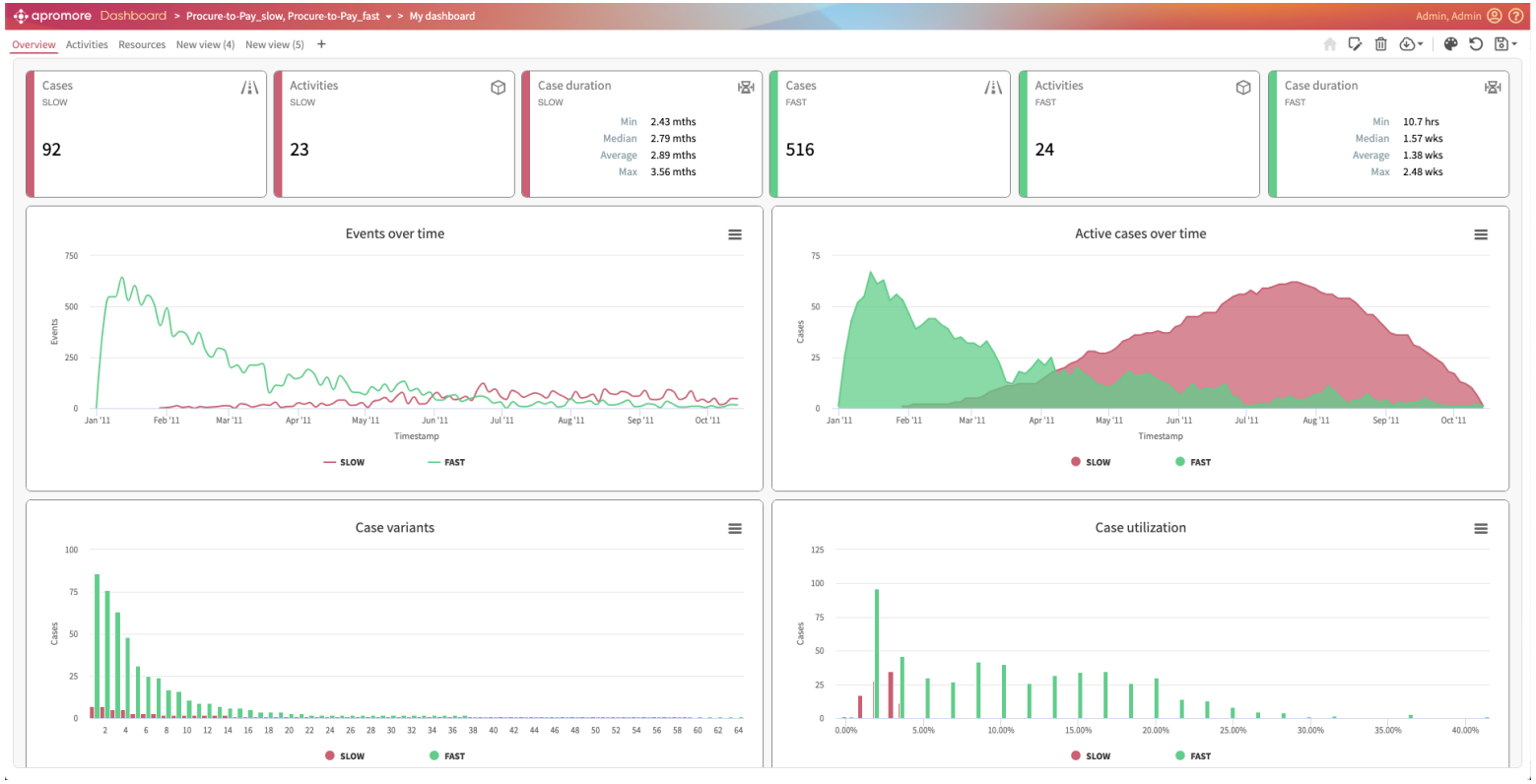


[Avg duration for full log: 1.1 hrs]

Variant analysis: comparison via dashboard

Statistical comparison via dashboard charts and tables. Variants are color-coded (e.g. red for slow, green for fast cases)



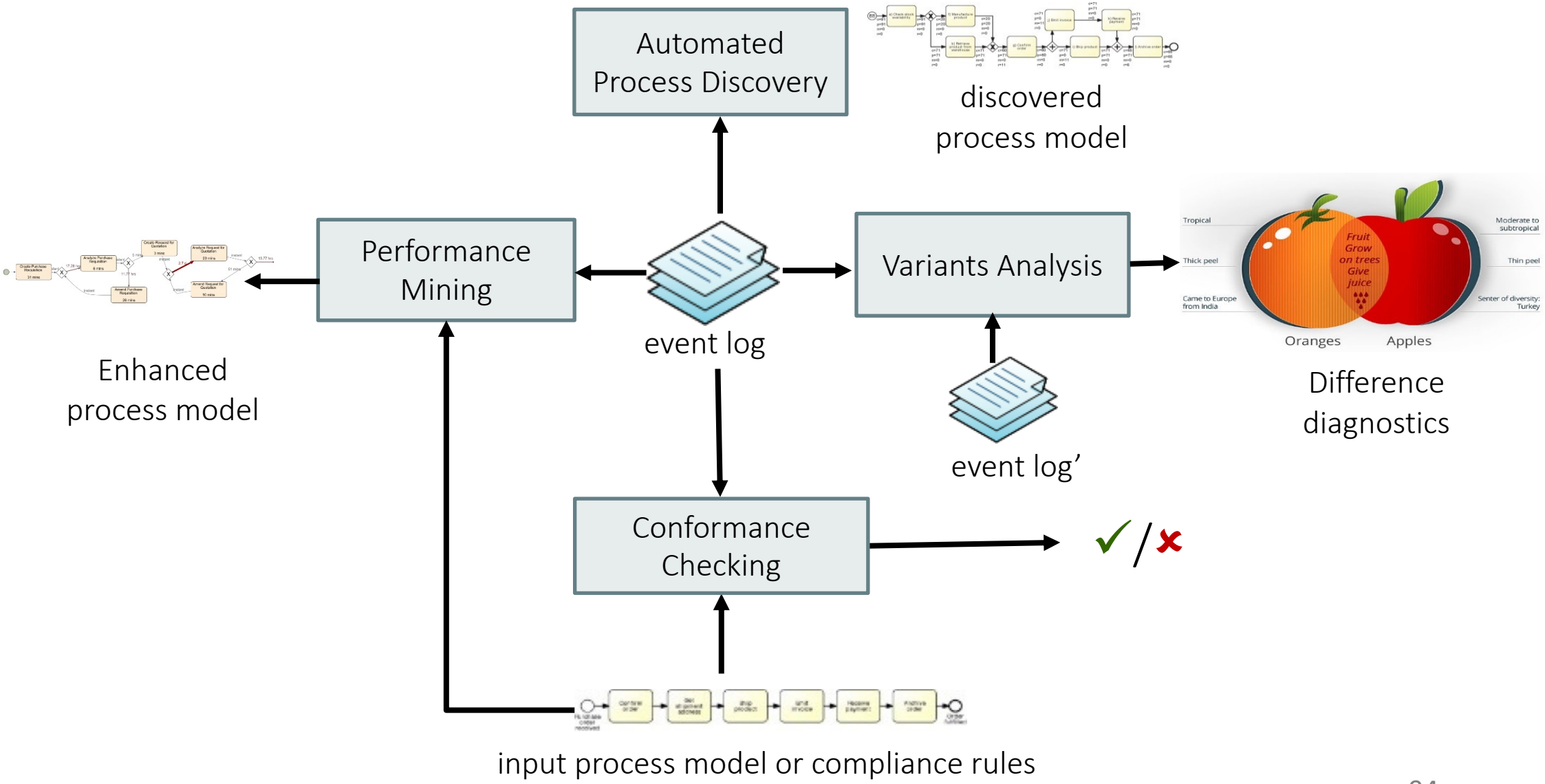


Select your filtered logs and launch the Dashboard. All dashboard functionalities are also available when comparing multiple variants.

Variant Analysis Template

What?	How?
Flow comparison Identify notable diffs in activity flow	• Discover a BPMN process model from the log of each variant. Visualize them side-by-side. Look for the gateways around each task to detect for example situations where two tasks are in parallel in one variant, but sequential in the other.
Frequency & rework comparison Compare activity or handoff frequency and rework	• Compare the maps of the process variants side-by-side (as above), but using the frequency overlay instead of the duration overlay. This allows us to detect the most frequent transitions between tasks. Look also for “thick” loop-backs. • Use a multi-low dashboard to compare frequencies of activities, resources or other attributes.
Bottleneck comparison Compare the location and magnitude of bottlenecks across variants	• Use filtering interface to retain all traces that fulfill the condition to belong to a variant (then repeat for the other). • Open the maps of the process variants side-by-side and compare the dependencies (arcs) with the highest waiting times. If the maps are too complex, use the abstraction slider with the abstraction metric “average duration” and the ordering from slow (right) to fast (left). Use the arc slider to retain the slowest dependencies. • Compare activity durations using side-by-side comparison of process maps or a multi-log dashboard (open the logs of 2+ variants in a single dashboard). • Consider comparing the two variants using the “resource” perspective to identify hand-offs between resources.

Summary


Automated Process Discovery
discovered process model
Performance Mining
Enhanced process model
event log
Variants Analysis
Tropical
Thick peel
Came to Europe from India
Fruit Grow on trees Give juice
Moderate to subtropical
Thin peel
Senter of diversity: Turkey
Oranges
Apples
Difference diagnostics
event log'
Conformance Checking
✓/✗
input process model or compliance rules

Summary: Four Process Mining Capabilities

The bulk of process mining techniques can be divided into four capabilities.

EVENT LOGS

Automated Process Discovery

By using the automated process discovery, an actual process model is created from the event log. This represents the current sequence of work steps and does not necessarily have to correspond to the target process.

Conformance Checking

Conformance checking provides a comparison of the actual and the target process and thus enables the detection of differences and deviations.

Variant Analysis

Process outcomes often differ in practice. But why?
The variant analysis enables a simple comparison of several process variants and serves as the basis for process optimization.

Performance Mining

By applying performance mining techniques, e.g. nodes or cycle times can be highlighted in color to simplify the subsequent analysis.

Acknowledgements

- All material comes from Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A. Reijers, authors of the “Fundamentals of Business Process Management” book.

INFO-H420
Management of Data Science and Business Workflows

Part II
Data Pipelines

Dimitris SACHARIDIS

2023-2024

Part II: Management of Data Science Workflows

- Introduction to Data Science Workflows
- Data Privacy
- Fairness
- Explainability

Data Science Workflows

Data Science – Definition

"***Data science*** *is the study of the generalizable extraction of knowledge from data.*"

- Vasant Dhar

- The term **science** implies knowledge gained through systematic study.
- A data scientist requires an **integrated skill set** spanning mathematics, machine learning, artificial intelligence, statistics, databases, and optimization, along with a deep understanding of the craft of problem formulation to engineer effective solutions.

Data Science and Prediction. Vasant Dhar. Communications of the ACM 56(12), 2013.
https://doi.org/10.1145/2500499

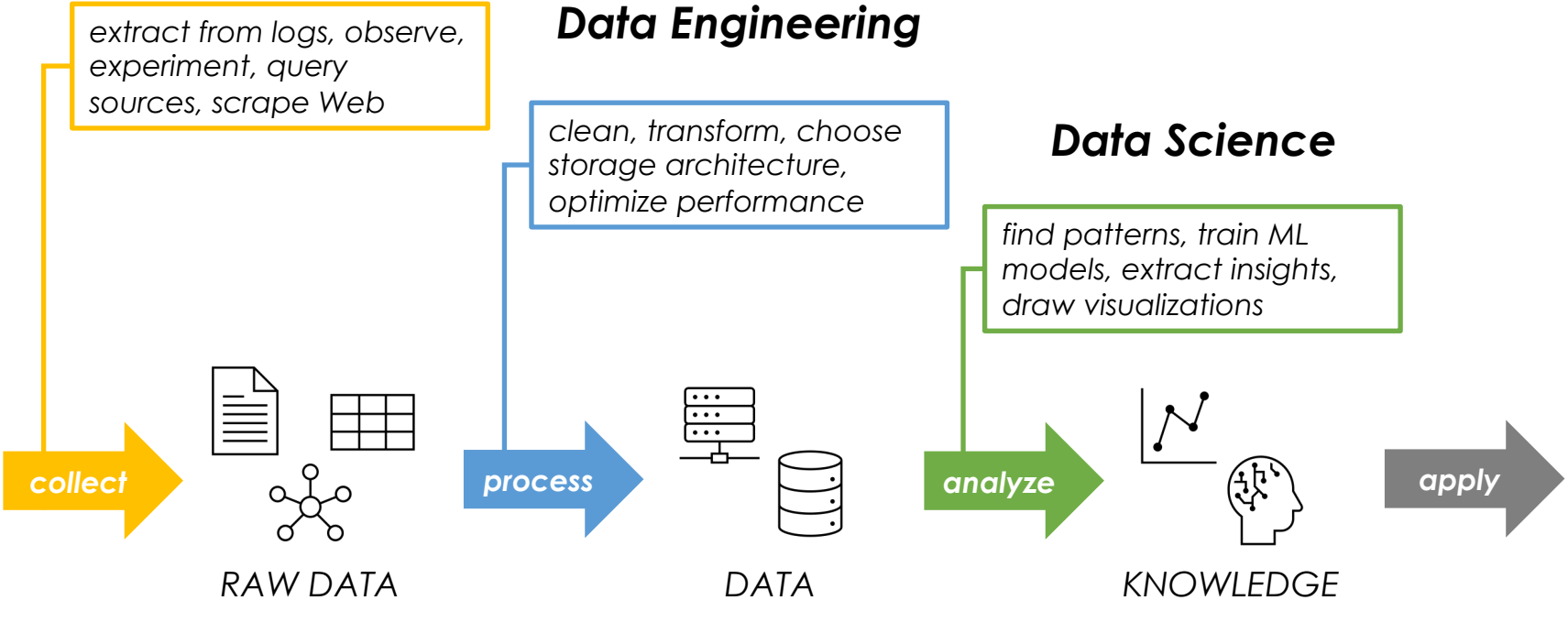
Data Science – Definition

Data science is an **interdisciplinary field that uses scientific methods**, **processes**, **algorithms and systems to extract** or extrapolate **knowledge** and insights **from** noisy, structured and unstructured **data**, **and apply knowledge from data** across a broad range of application domains.

Data science is related to **data mining**, **machine learning**, **big data**, **computational statistics**, and **analytics**.

- Wikipedia

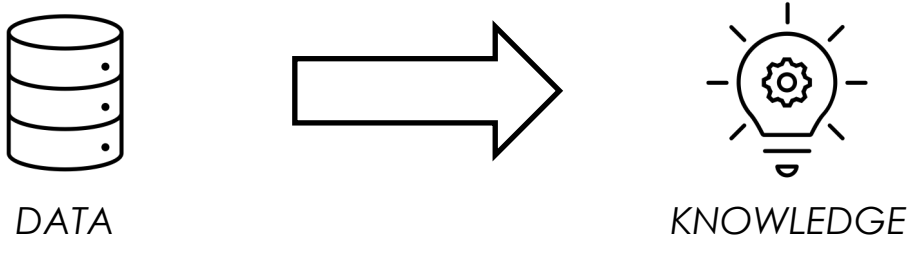
The Data Lifecycle



Data Science vs. Data Engineering

Same Goal

convert DATA to KNOWLEDGE



Different Focus

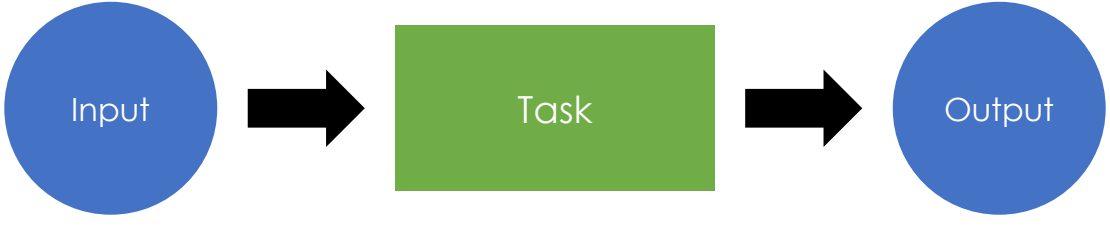
Data Engineering

is it fast?

Data Science

is it good?

Data Science Tasks



Input	Task	Output
Relational Table	SQL query	Relational Table
Unstructured Data	Information Extraction	Structured Data
Labelled Data	ML training	ML Model
ML Model	Fine Tuning	ML Model
Raw Data	Preprocessing	Clean Data
Structured Data	Visualization	Chart

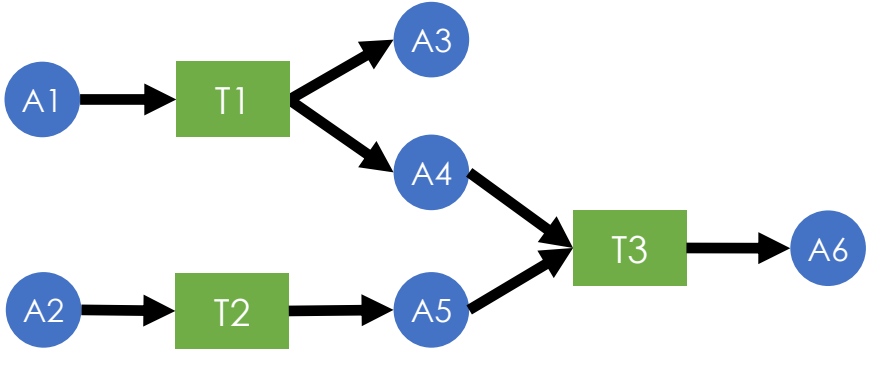
Because the **Input** and **Output** of a Task can have various forms/structures/types (data, chart, text, model), we simply call them **Artifacts**

Data Science Workflows

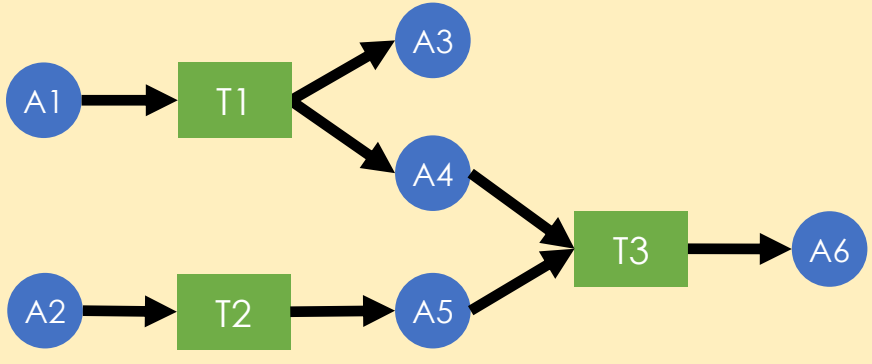
A Data Science **Workflow**, or Data **Pipeline**, is a sequence of

- artifacts
- tasks

Typically portrayed as a **graph**

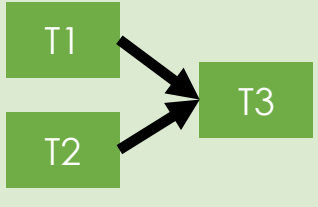


Data Science Workflows



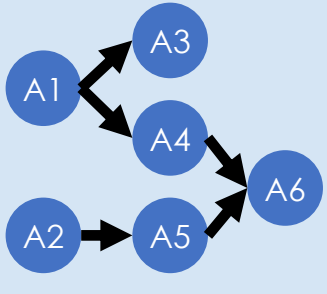
Task-Artifact Graph

- shows **flow** among tasks and artifacts
- two types of nodes: **tasks**, **artifacts**
- edges encode **input requirements**, and **outputs**



Task Graph

- shows **dependencies** among **tasks**
  - *task dependency graph*
- artifacts are hidden in edges



Artifact Graph

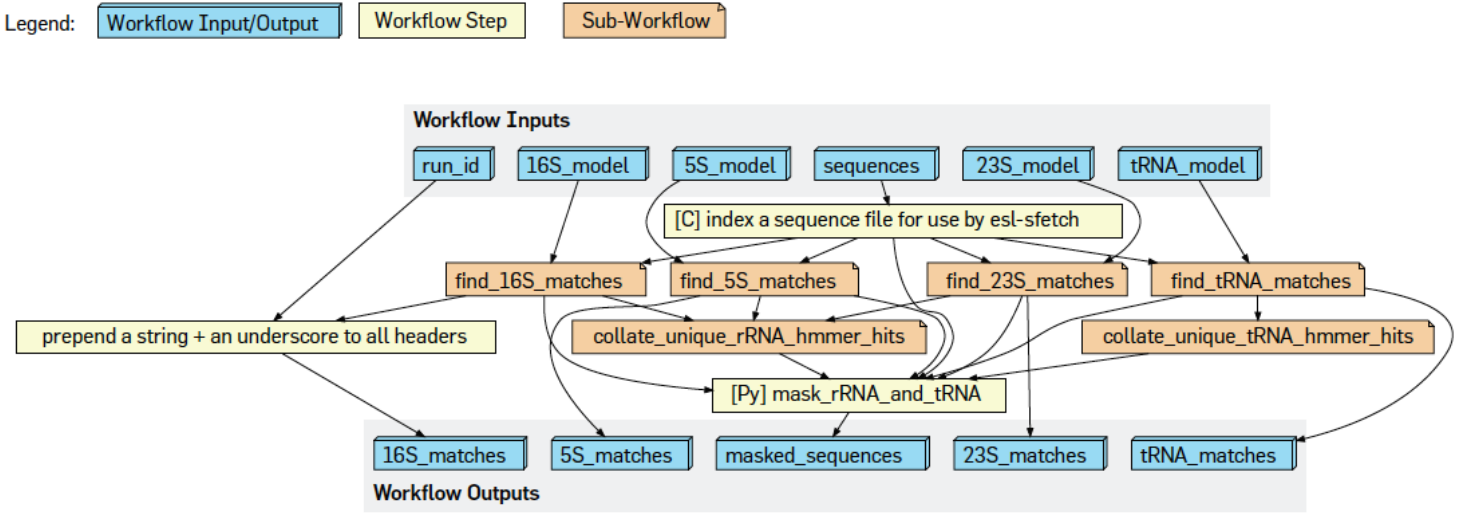
- shows relationships among **artifacts**
- edges encode **tasks**
- tasks are hidden in edges

- all graphs are **directed acyclic graphs** (DAGs)
  - directed edges indicate flow, dependency relationships
  - no (directed) cycles

Examples of Workflows

Workflows are very popular in Bioinformatics

Example of a **task-artifact** graph for a workflow that matches inputs of genomic sequences to provided sequence models, expressed in the Common Workflow Language (CWL)



Methods included: standardizing computational reuse and portability with the Common Workflow Language. Crusoe et al. Communications of the ACM 65(6), 2022. https://dl.acm.org/doi/10.1145/3486897

Examples of Workflows

Database queries result in execution plans

Consider the execution of the query:

MODEL = "CIVIC" AND YEAR = 2001 AND
(COLOR = "GREEN" OR COLOR = "WHITE")

on the following table:

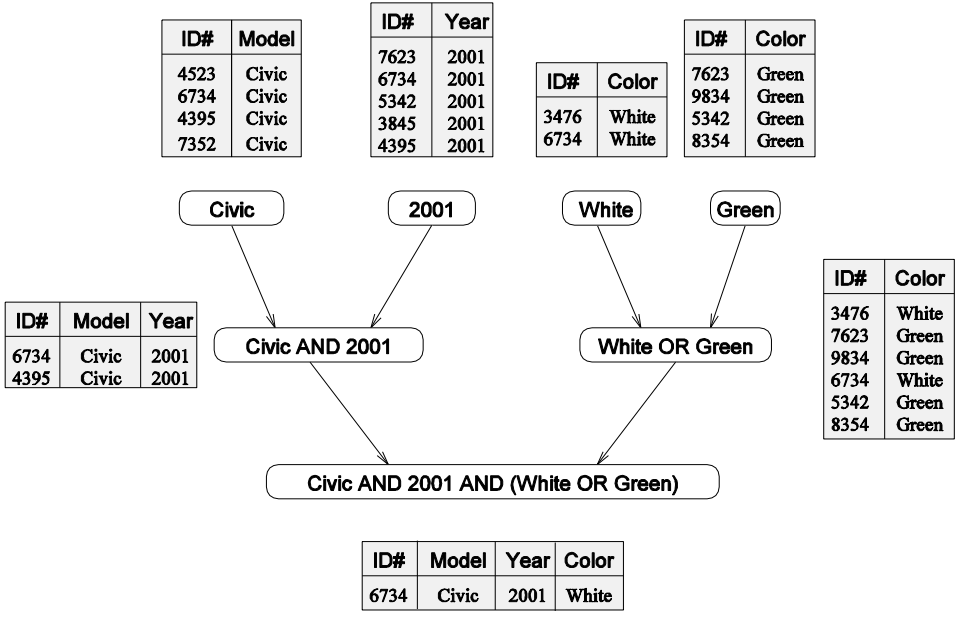
ID#	Model	Year	Color	Dealer	Price
4523	Civic	2002	Blue	MN	$18,000
3476	Corolla	1999	White	IL	$15,000
7623	Camry	2001	Green	NY	$21,000
9834	Prius	2001	Green	CA	$18,000
6734	Civic	2001	White	OR	$17,000
5342	Altima	2001	Green	FL	$19,000
3845	Maxima	2001	Blue	NY	$22,000
8354	Accord	2000	Green	VT	$18,000
4395	Civic	2001	Red	CA	$17,000
7352	Civic	2002	Red	WA	$18,000

Examples of Workflows

Database queries result in execution plans.

Each task can be thought of as generating an intermediate table of entries that satisfy a particular clause.

Here's an **artifact graph**.

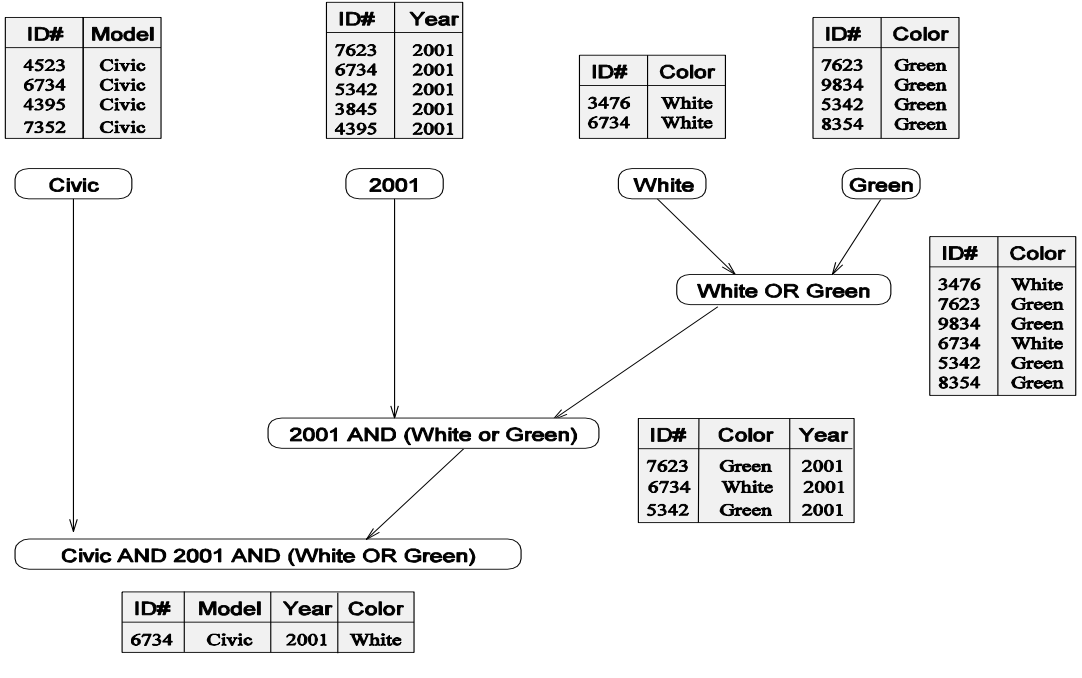


Examples of Workflows

Database queries result in execution plans.

An alternate decomposition of the given problem into subtasks, along with their data dependencies.

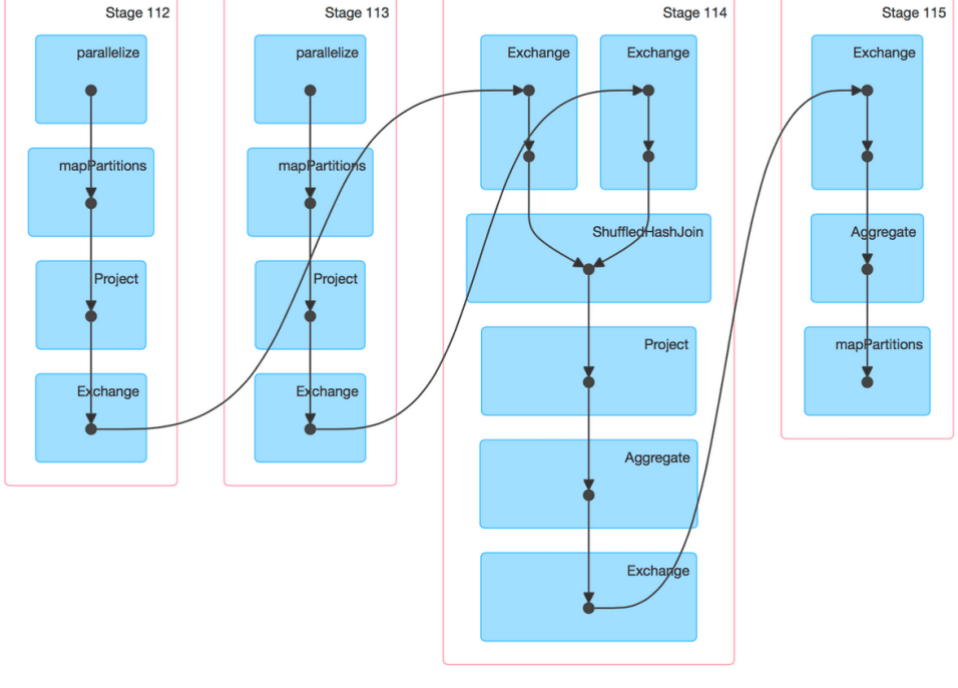
Here's an **artifact graph**.



Examples of Workflows

Big Data analytics is based on **parallel execution** of task graphs

Example of a **task graph** in Spark



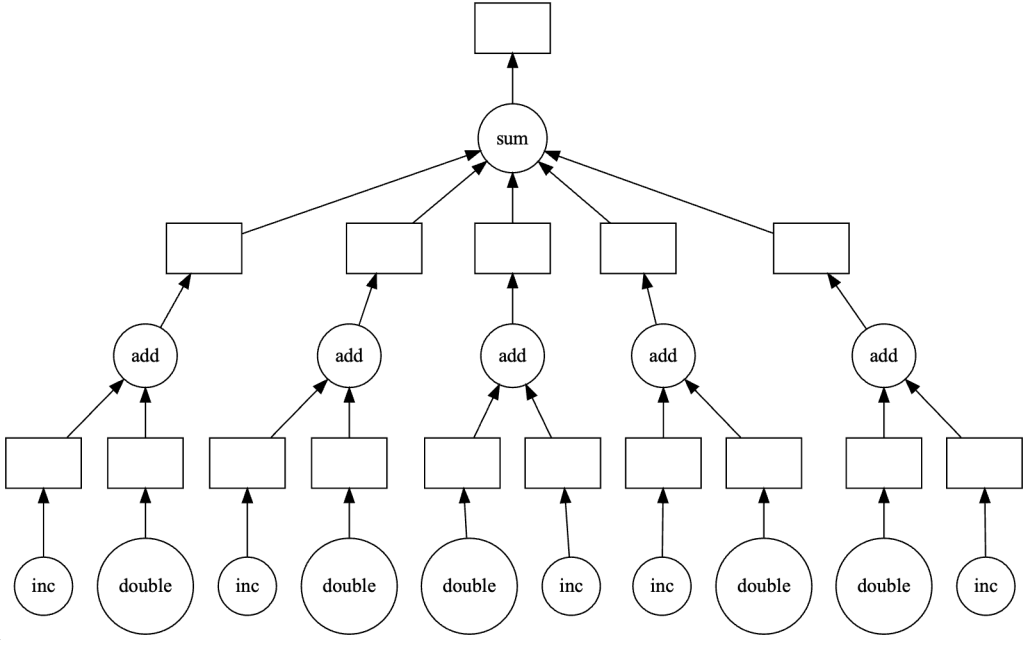
https://www.databricks.com/blog/2015/06/22/understanding-your-spark-application-through-visualization.html

Examples of Workflows

Big Data analytics is based on **parallel execution** of task graphs

Example of a **task-artifact** graph in Dask

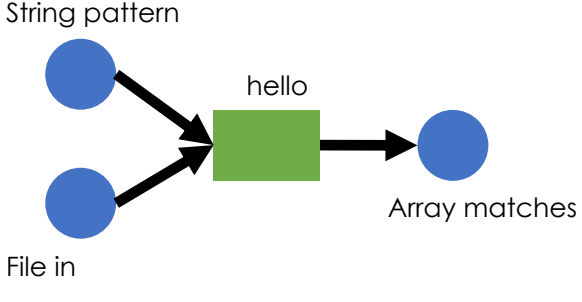
```python
import dask

@dask.delayed
def inc(x):
    return x + 1

@dask.delayed
def double(x):
    return x * 2

@dask.delayed
def add(x, y):
    return x + y

data = [1, 2, 3, 4, 5]

output = []
for x in data:
    a = inc(x)
    b = double(x)
    c = add(a, b)
    output.append(c)

total = dask.delayed(sum)(output)
```



Workflow Languages

- Workflows are typically described in some standardized language
- The **Workflow Description Language** (WDL) aims to have a very human-readable and -writeable syntax
  - A wdl file has syntax similar to Python

```
task hello {
  input {
    String pattern
    File in
  }

  command {
    egrep '${pattern}' '${in}'
  }

  runtime {
    docker: "broadinstitute/my_image"
  }

  output {
    Array[String] matches = read_lines(stdout())
  }
}

workflow wf {
  call hello
}
```



https://github.com/openwdl/wdl

Workflow Languages

- The **Common Workflow Language** (CWL) is written in YAML markup

a task

```yaml
cwlVersion: v1.0
class: CommandLineTool

doc: Spoa is a partial order aligment…

inputs:
  readsFA:
    type: File
    format: edam:format_1929
    doc: FASTA file containing a set of sequences…

requirements:
  InlineJavascriptRequirement: {}
hints:
  DockerRequirement:
    dockerPull: "quay.io/biocontainers/spoa:3.4.0--hc9558a2_0"
  ResourceRequirement:
    ramMin: $(15 * 1024)
    outdirMin: $(Math.ceil(inputs.readsFA.size/(1024*1024*1024) + 20))

baseCommand: spoa

arguments: [ $(inputs.readsFA), -G, -g, '-6' ]

stdout: $(inputs.readsFA.nameroot).g6.gfa

outputs:
  spoaGFA:
    type: stdout
    format: edam:format_3976
    doc: result in Graphical Fragment Assembly (GFA) format

$namespaces:
  edam: http://edamontology.org
```

1. Community Maintained File Format Identifier

2. Software Container

3. Dynamic Resource Requirements

a workflow

```yaml
cwlVersion: v1.0
class: Workflow

inputs:
  pattern: string
  sample_data: File[]

steps:
  find_matches:
    run: grep.cwl
    in:
      pattern: pattern
      files: sample_data
    out: [ text_matches ]

  count_lines:
    run: wc.cwl
    in:
      file: find_matches/text_matches
    out: [ lines ]

outputs:
  number_of_matches:
    type: int
    outputSource: count_lines/lines
```

C

COMMON WORKFLOW LANGUAGE

Methods included: standardizing computational reuse and portability with the Common Workflow Language. Crusoe et al. Communications of the ACM 65(6), 2022. https://dl.acm.org/doi/10.1145/3486897

Why describe Workflows

- One important reason is to enable Open Science and adhere to the **FAIR** Guiding Principles:
  - Findable, Accessible, Interoperable, Reusable

the three R's: Repeatability, Reproducibility, Reusability

- **repeat** the workflow with same input on same environment and get the same output
- **reproduce** the output of the workflow with the same input on a different environment
- **reuse** workflow, or parts of it, to solve a different problem

Workflow Registries

- A Workflow Registry is a Platform where one can
  - **register** workflows
    - upload a workflow description, in some supported language
    - receive a unique persistent identifier (pid)
  - **find** and **reuse** workflows
    - explore registered workflows
    - download workflows in supported languages
  - (**execute** workflows)

Dockstore
Create, Share, Use

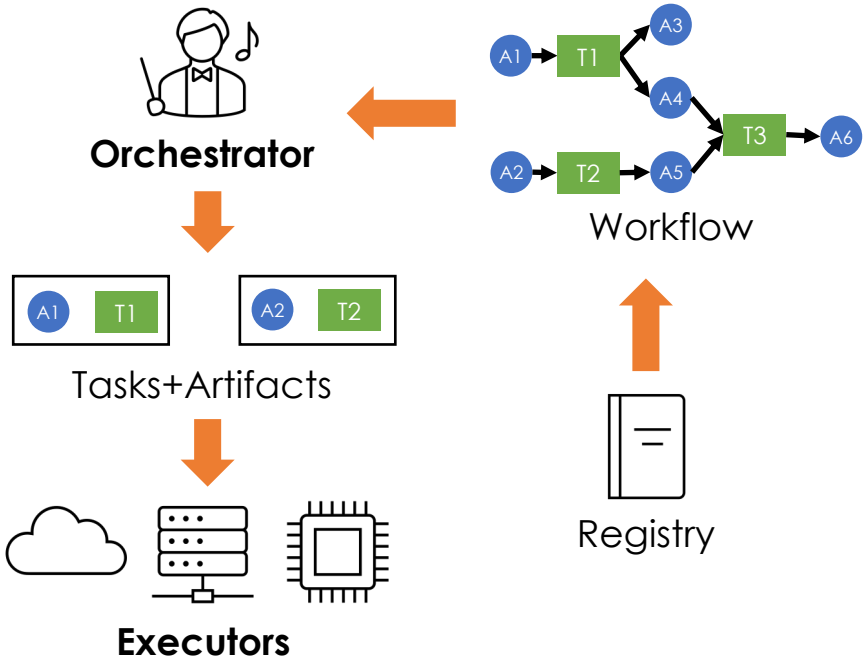
WorkflowHub

Executing Workflows

Executing Workflows

- Data Science Workflows are **compute**- and **data**-intensive
- Require heavy computational resources
  - computational clusters, high-performance computing (HPC) machines on premises or in the cloud

- To execute a workflow, you need:
  - An **Orchestrator** to decide how and when to assign tasks to Executors
  - **Executors** that run Tasks



Cloud Computing

- Cloud computing refers to the delivery of **computing services** such as storage, networking, and computing power **over the internet** (the "cloud")
- Cloud computing allows users to access and use these services on-demand, without having to manage the **underlying infrastructure** themselves
- This can be more cost-effective and scalable than maintaining your own **on-premises** infrastructure
- There are different types of cloud computing
  - **Public cloud** services are available to the general public, typically on a pay-as-you-go basis
  - **Private cloud** services are operated exclusively for a single organization, often on-premises
  - **Hybrid cloud** refers to a combination of public and private cloud

Cloud Computing

- Common examples of cloud computing services include **Infrastructure-as-a-Service** (IaaS), **Platform-as-a-Service** (PaaS), and **Software-as-a-Service** (SaaS)
- IaaS provides users with access to **fundamental computing resources** such as virtual machines, storage, and networking
- PaaS provides users with a **platform for developing and deploying** their own applications, without having to manage the underlying infrastructure
- SaaS provides users with access to **software applications** that are hosted and managed by the provider

Cloud Computing Platforms

- Amazon Web Services (AWS) (since 2006)
- Microsoft Azure (since 2010)
- Google Cloud Platform (since 2008)

Kubernetes

- Kubernetes (abbreviated as k8s) is an open-source platform for **managing** and **orchestrating** containerized applications
  - applications run on-premises, in the cloud, or in a hybrid environment
  - In containers, such as Docker containers
- Kubernetes uses a declarative approach, where users specify the **desired state** of their applications and Kubernetes automatically ensures that the application matches that state
- Kubernetes is **highly scalable**, allowing users to easily add or remove containers and resources as needed

kubernetes

Kubernetes

- Created by Google, open sourced and donated to the Cloud Native Computing Foundation
  - Comes from the Greek word for helmsman

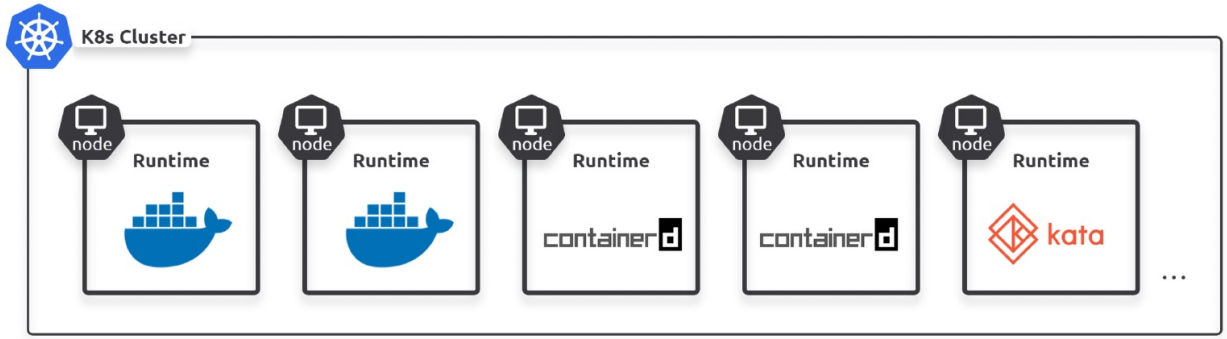
- Kubernetes is considered the **operating system of the cloud**
  - a traditional OS on a server **abstracts server resources** and **schedules applications**
  - Kubernetes on a cloud **abstracts cloud resources** and **schedules containerized applications**

Containers

- A **containerized application** is an app packaged and run in a container
  - Makes it portable, scalable, and easier to deploy and manage
- A **container** is a lightweight, standalone, and executable **package** of software that includes **everything needed to run the application**, such as the code, libraries, dependencies, and runtime
- Containers are **isolated** from each other and from the host operating system, allowing them to run consistently across different environments
- A popular tool for creating and managing containers is Docker

Kubernetes Cluster

- Kubernetes runs on a cluster in the cloud, where a group of nodes are used to run containerized applications
  - A node is a physical or virtual machine, provided by the cloud platform, that runs the Kubernetes software and hosts containers
- A cluster typically consists of at least one **master** node and multiple **worker** nodes.
- The master node runs the Kubernetes **control plane**, which is responsible for managing and coordinating the worker nodes.
- The worker nodes run the Kubernetes **kubelet**, which is responsible for running and managing the containers on that node.



Kubernetes is an Executor

In the context of Data Science workflows

- A **Task** (with its input artifacts) can be delivered as a Containerized Application
- And Kubernetes can act as the **Executor**, running the Tasks on the worker nodes, abstracting the underlying computational infrastructure

- But Kubernetes is not an **Orchestrator** of workflows
  - Only sees/knows of individual Tasks

Orchestrators of Workflows

Workflow Orchestrators

- Orchestrators **deploy** Workflows on Executors (the execution infrastructure)
- Provide the features and tools to **monitor** and **manage** workflows

- Examples are Kubeflow and Airflow

Kubeflow

Apache Airflow

Kubeflow + Argo Workflows

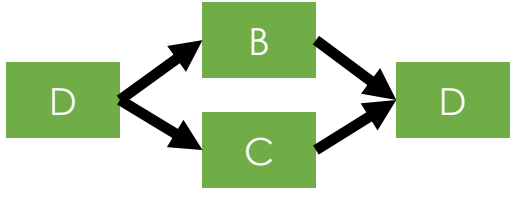
- Kubeflow is an open-source platform for deploying and managing workflows on Kubernetes
  - Created by Google
- Focused on machine learning (ML) workflows, provides a set of tools and components to make it easier to develop, train, and deploy ML models at scale
- Kubeflow works with Argo workflows, an open-source tool for orchestrating parallel and sequential workflows on Kubernetes
- Argo Workflows uses Kubernetes resources to execute the tasks in a workflow

Kubeflow + argo

Argo Workflows

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  generateName: dag-diamond-
spec:
  entrypoint: diamond
  templates:
  - name: echo
    inputs:
      parameters:
      - name: message
    container:
      image: alpine:3.7
      command: [echo, "{{inputs.parameters.message}}"]
  - name: diamond
    dag:
      tasks:
      - name: A
        template: echo
        arguments:
          parameters: [{name: message, value: A}]
      - name: B
        dependencies: [A]
        template: echo
        arguments:
          parameters: [{name: message, value: B}]
      - name: C
        dependencies: [A]
        template: echo
        arguments:
          parameters: [{name: message, value: C}]
      - name: D
        dependencies: [B, C]
        template: echo
        arguments:
          parameters: [{name: message, value: D}]
```

- Argo Workflows uses a YAML-based language
- Here a diamond-shaped **task graph** is defined



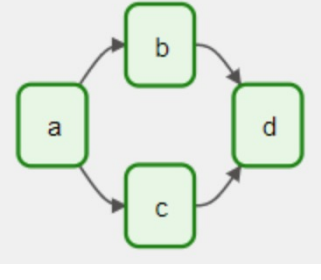
Airflow

- Apache Airflow is an open-source platform for scheduling and orchestrating workflows
  - Created by Airbnb, now open-source
- Airflow uses a DAG to represent the tasks in a workflow and their dependencies (task-graph)
- Airflow provides a **web-based** user interface for managing and monitoring the workflows, as well as a command-line interface and **Python-based API**
- Available at major Cloud Platforms
- Works with various Executors
  - including Kubernetes

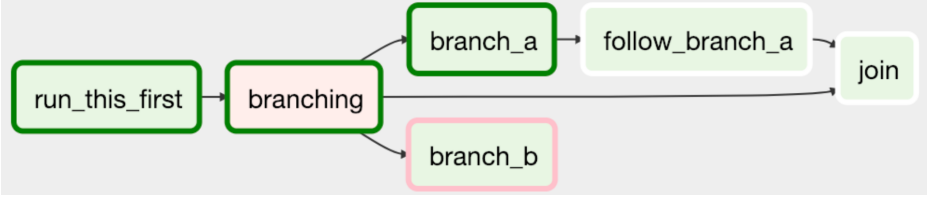
Airflow

- A DAG can be declared in Python, and viewed in the web-based interface
- Here's the diamond task graph

```
first_task >> [second_task, third_task]
third_task << fourth_task
```



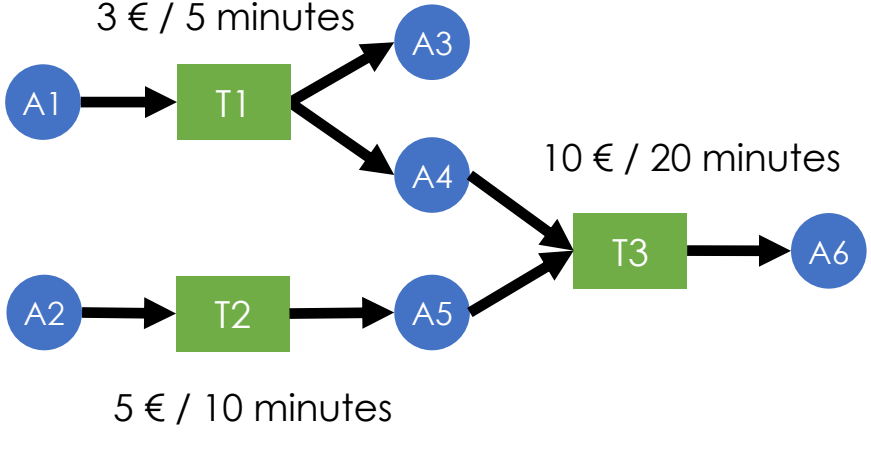
- Airflow also support conditional execution (branching)



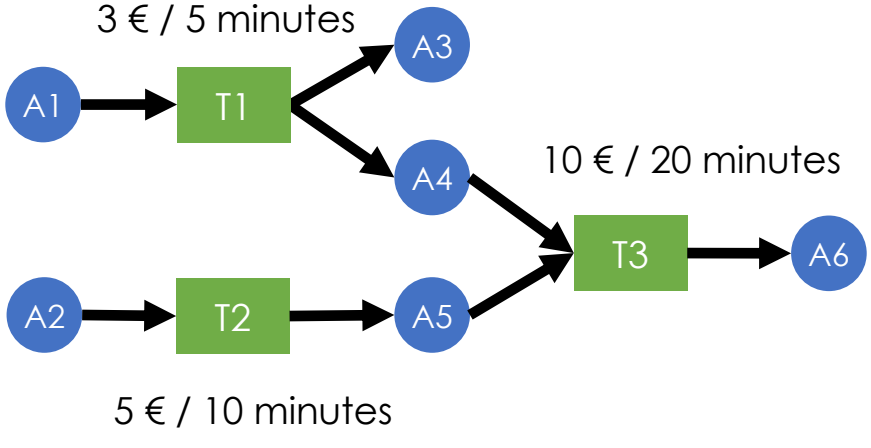
Workflow Optimization

Task Cost Measures

- Executing Tasks consumes resources
- Each task has two cost measures
  - the **cost**, e.g., in terms of money,
  - the **time** it takes to execute (response time, duration, cycle time)
- We can indicate these measures on the tasks in a dag



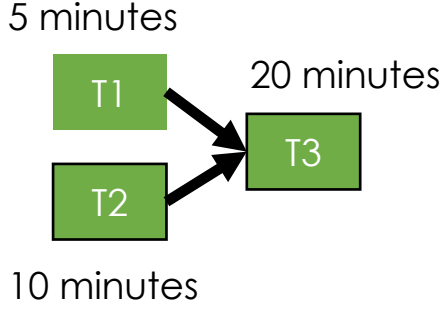
Workflow Cost Measures



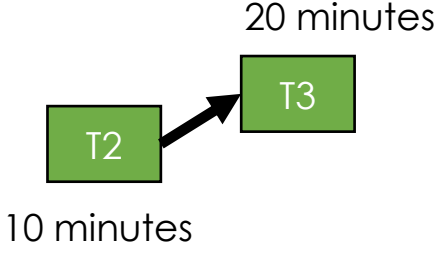
- What is the **total cost** of running this workflow?
  - No surprises, all tasks have to be executed, so 18 €
- What is the **total time** of running this workflow?
  - It depends on how you orchestrate it!
  - What tasks can run in parallel?

Critical Path in Workflows

- The **critical path** is the sequence of tasks that determines the **minimum time required** to execute the workflow
- It is called the "critical" path because if any of the tasks on this path are delayed, it will delay the entire workflow
- It corresponds to a **longest path** on the task graph
- We define the workflow's **total time** as the time of a critical path
  - On this workflow, the total time is 30 minutes



task graph



a critical path

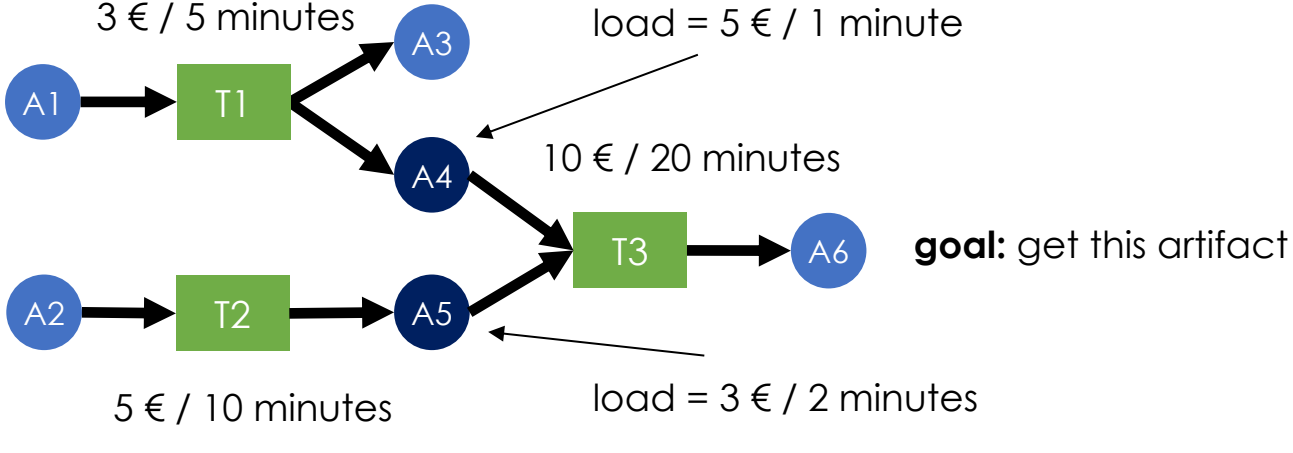
Reuse – Materialization

- Data Science Workflows are meant to be repeated and reused
  - e.g., in data explorative analysis, ML model selection
- Some of the artifacts may be used across workflows

- Suppose we **materialize**, i.e., store, them for future **reuse**. Can we reduce cost and time?

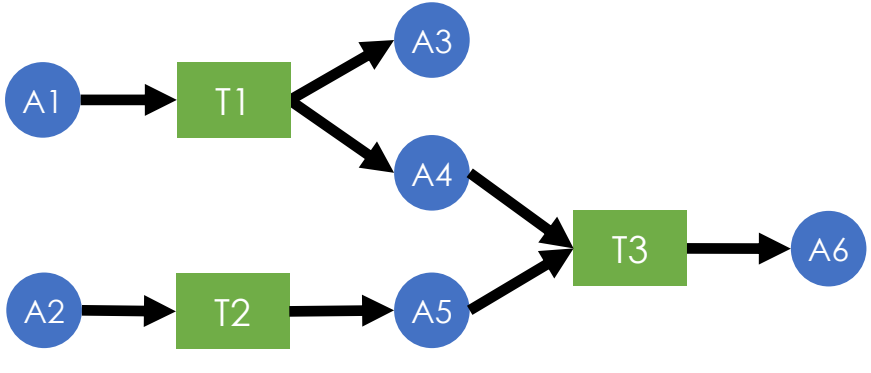
Reuse Problem

- Given a set of artifacts that are materialized, decide whether to **compute** or **reuse** them when executing a workflow
- What would you decide here?
  - If I cared about cost, I would load A5.
  - If I cared about time, I would load A4 and A5
- Solving the Reuse Problem is computationally hard
  - Equivalent to Max-Flow problem in graphs, about $O(n^3)$



Materialization Problem

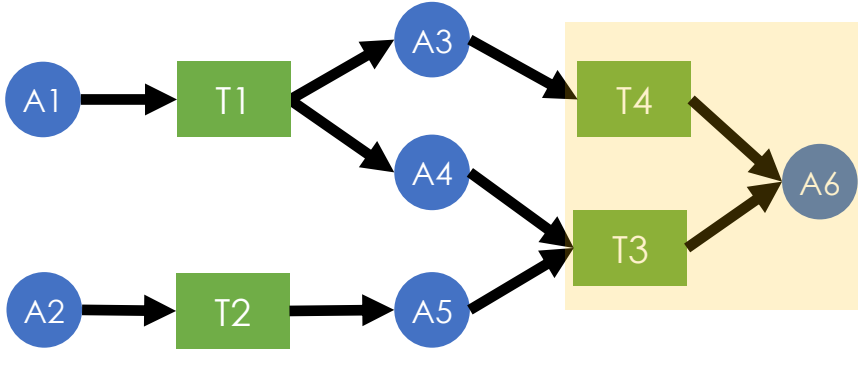
- Given a prediction about future workflows decide if and what artifacts to **materialize**
  - To store or not store A1, A2, …?
- Solving the Materialization Problem is computationally very hard
  - Even assuming complete knowledge of the future
  - NP-hard



goal: get this artifact

Alternatives in Task-Artifact Graphs

- So far in the task-artifact graphs, there is only one way to obtain an artifact
- In the **reuse problem**, there are two ways: **compute** (do the workflow) or **reuse** (load from store)
- More generally, there can be alternate ways to obtain an artifact. Which way to go?
- Should we compute A6 via T4 or via T3?

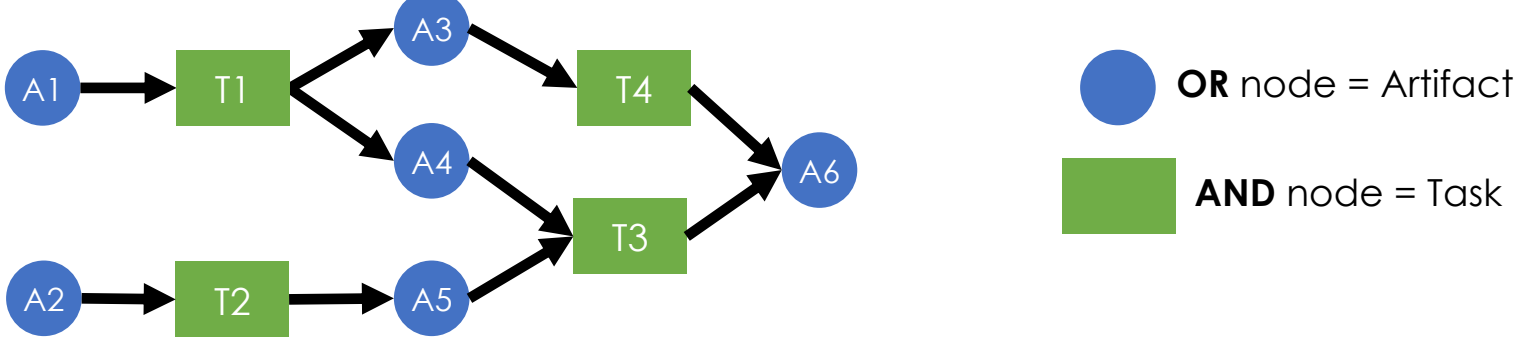


T4 and T3 are equivalent alternatives to get A6

goal: get this artifact

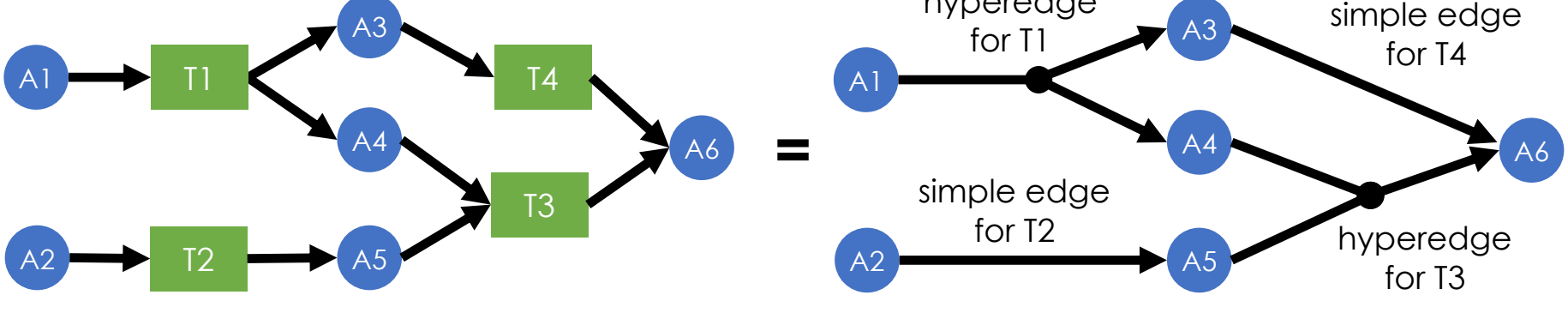
Task-Artifacts Graphs as AND/OR Graphs

- AND/OR graphs is an abstraction to represent alternatives in workflows
- An OR node means that there are alternative but equivalent ways to reach the node, and only **one in-edge** is necessary
  - All artifacts are OR nodes; most have only one in-edge
- An AND node means that the node requires **all in-edges**
  - All tasks are AND nodes



Artifact Hypergraph

- an AND/OR graph can be represented as a hypergraph
- a hypergraph has **hyperedges**
  - edge is from **one** node to **another** node
  - hyperedge is from a **set** of nodes to another **set** of nodes
- hyperedges represent tasks

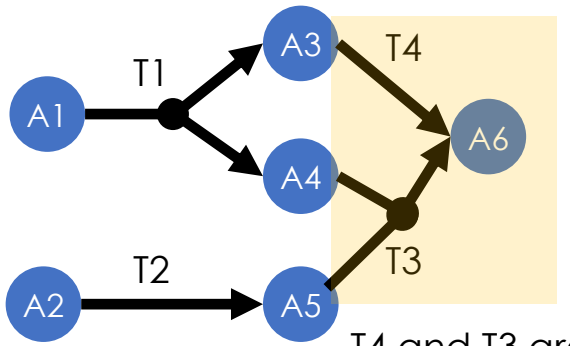


AND/OR Graph

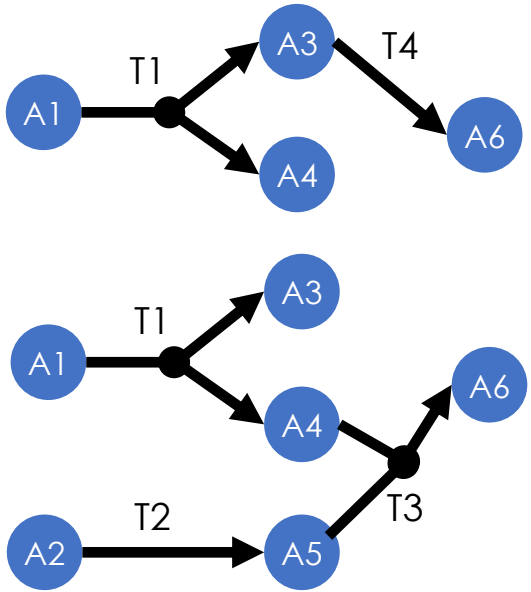
Hypergraph

A Plan for an Artifact Hypergraph

- hypergraph encodes alternative ways to retrieve artifacts
  - alternatives exists when an artifact has multiple in-edges
- a **plan** is a sub-hypergraph where each node has **a single in-edge**
- **reuse problem** = create a plan from an artifact hypergraph
  - computationally very hard (NP-hard) to find optimal plan



T4 and T3 are alternatives to get A6



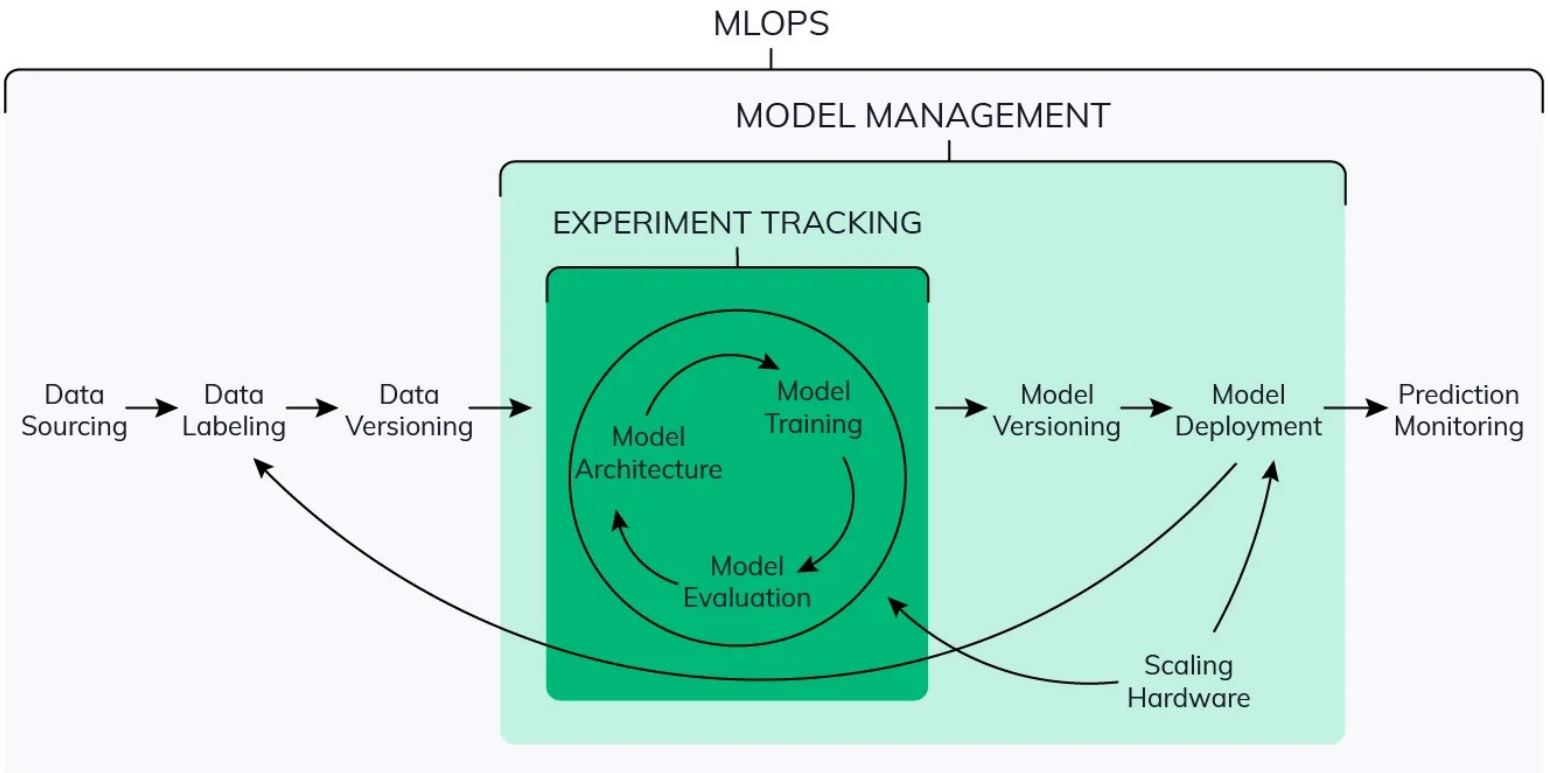
plan 1: keep T4

plan 2: keep T3

Data Science Workflows in Practice

ML Lifecycle

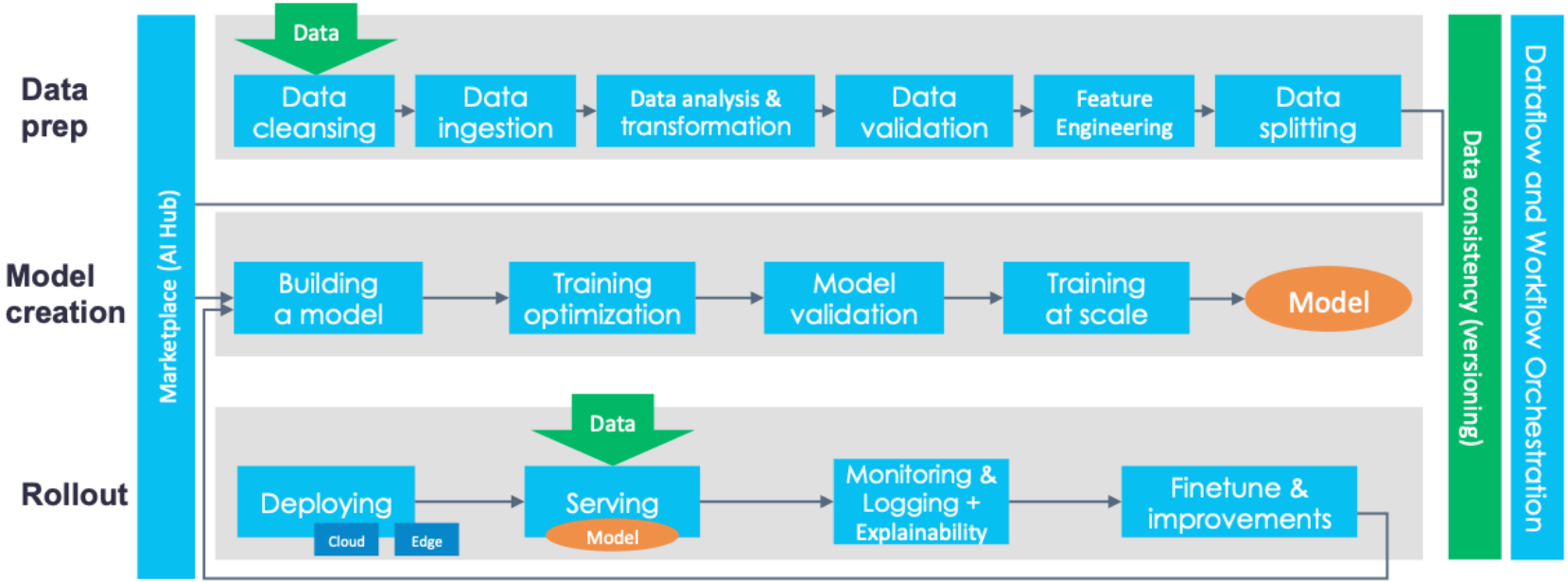
ML Operations (MLOps) = management of the entire ML lifecycle



https://neptune.ai/blog/mlops

ML Lifecycle

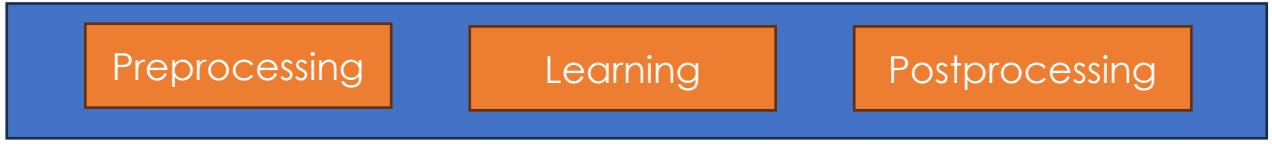
- several workflows within the lifecycle
  - experimentation
  - deployment



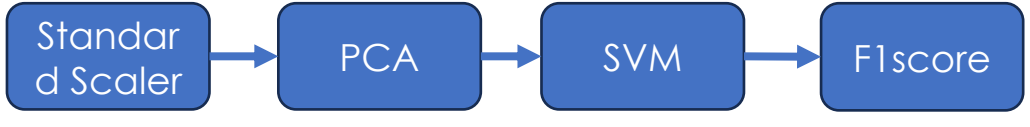
https://www.kdnuggets.com/2020/07/tour-end-to-end-machine-learning-platforms.html

Abstractions of ML Workflows

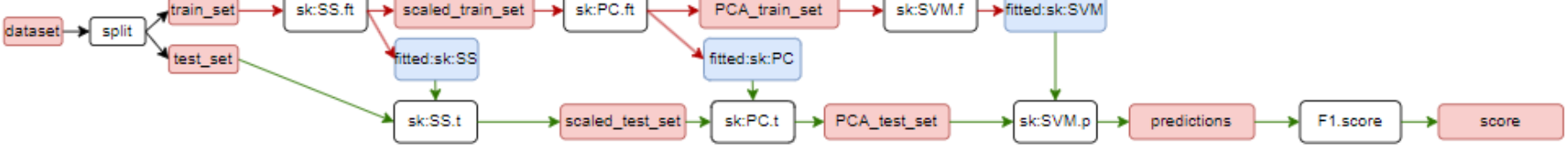
Common Machine Learning Strategy



Example Logical Pipeline



Example Physical Pipeline



Designing a Workflow

at the **physical level** (Implicit): the data scientist provides the code to be executed (e.g., on Jupyter Notebooks)

at the **logical level** (Explicit): the data scientist doesn't have to code

There are two main popular ways of implementing workflows at the logical level:

- Pipeline libraries like (SparkML, Sklearn, ML.Net)
- Data Analytics Platforms (DAPs) that allow the user to graphically construct a Logical pipeline

Combining both: Libraries and DAPs allow the user to add custom functionality and implement their own operations

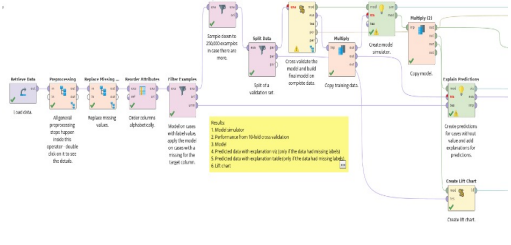
Data Analytics Platforms

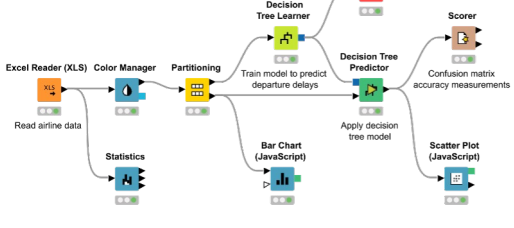
- Offer a graphical UI for describing logical pipelines, by dragging and dropping operators
- The user is not aware of the underneath code

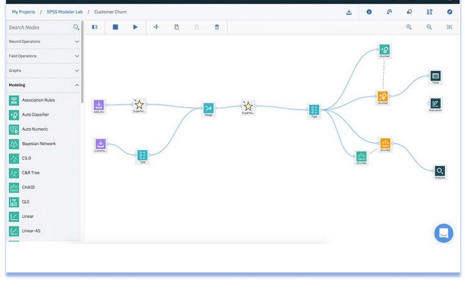
RAPIDMINER

Open for Innovation
KNIME

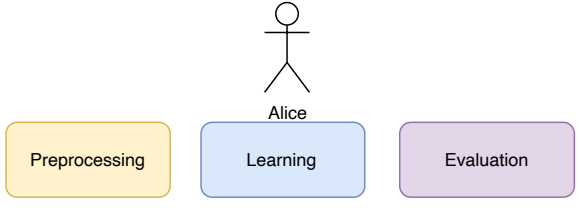
IBM Watson



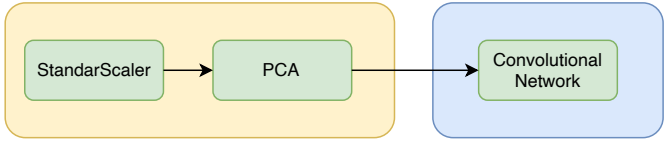




Every Data Scientist can approach the problem with a different strategy

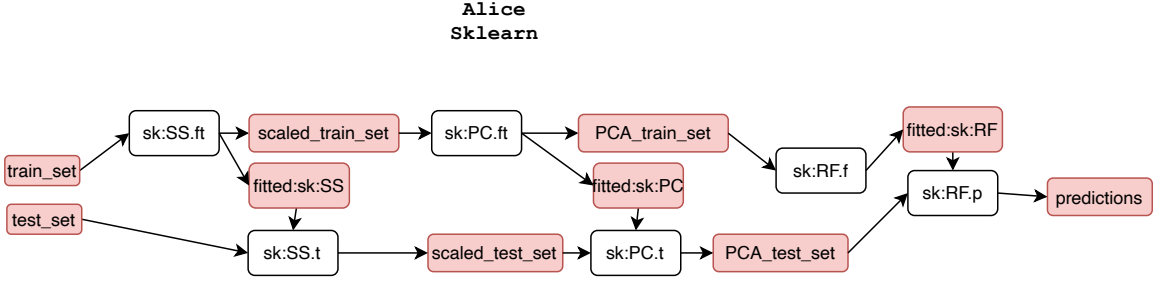


The logic underneath their strategy can be the same or different



Operator

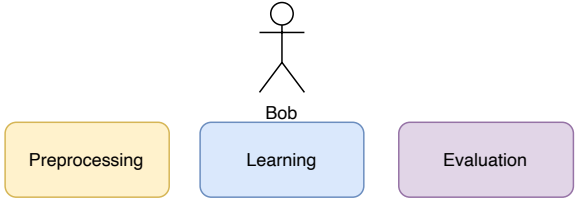
Similar their implementation underneath their logic can be the same or different



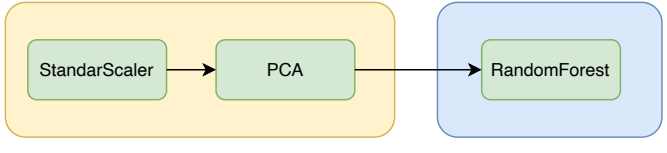
Artifacts

Task

Every Data Scientist can approach the problem with a different strategy

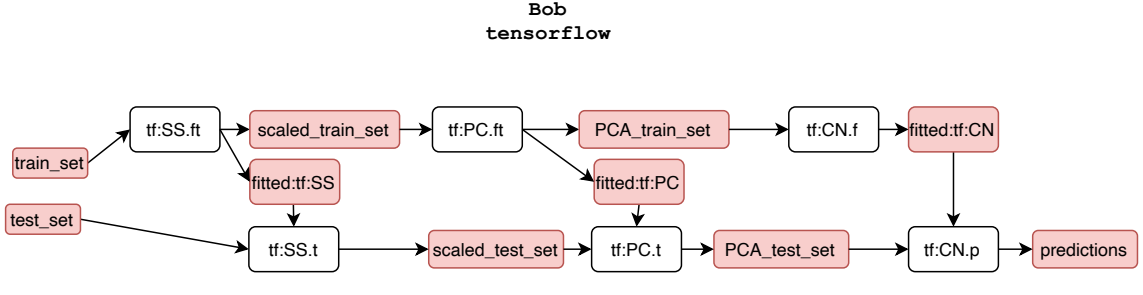


The logic underneath their strategy can be the same or different



Operator

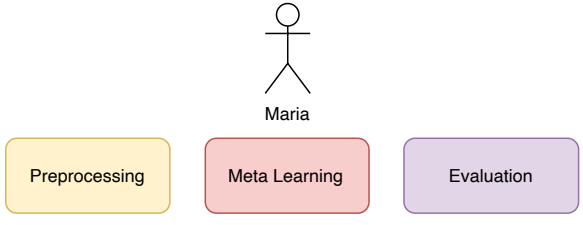
Similar their implementation underneath their logic can be the same or different



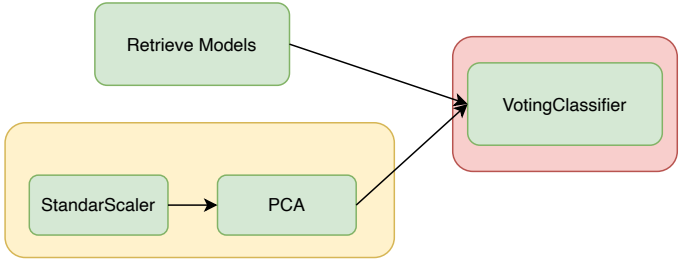
Artifacts

Task

Every Data Scientist can approach the problem with a different strategy

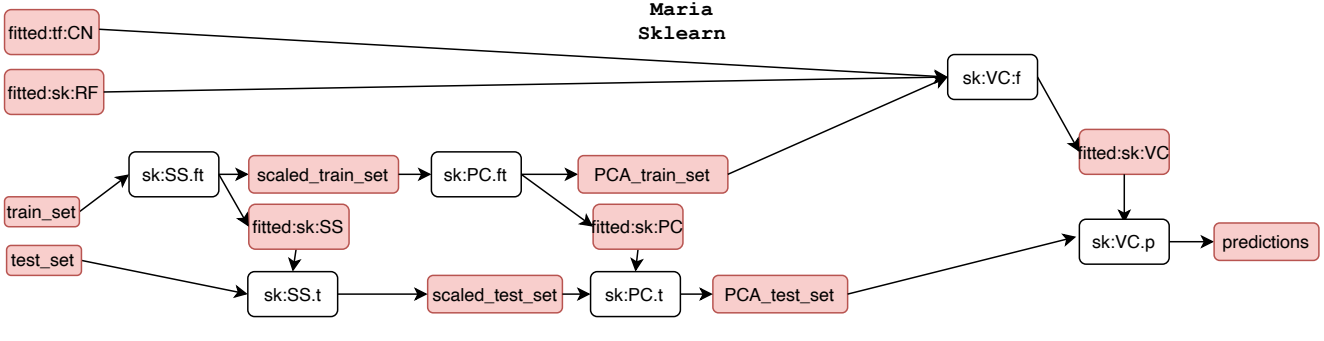


The logic underneath their strategy can be the same or different



Operator

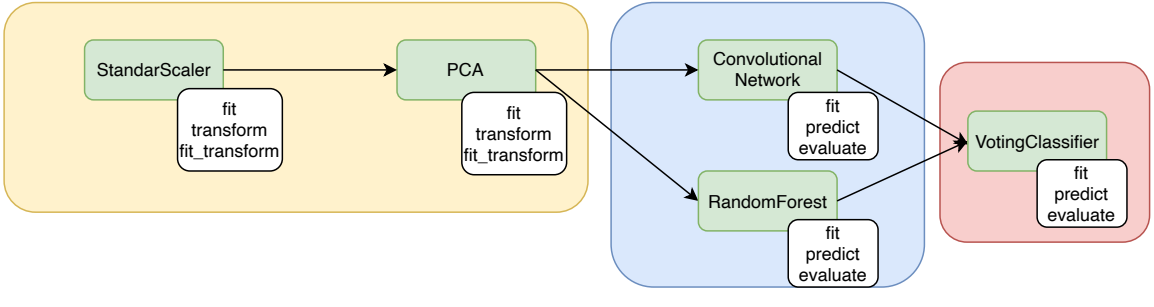
Similar their implementation underneath their logic can be the same or different



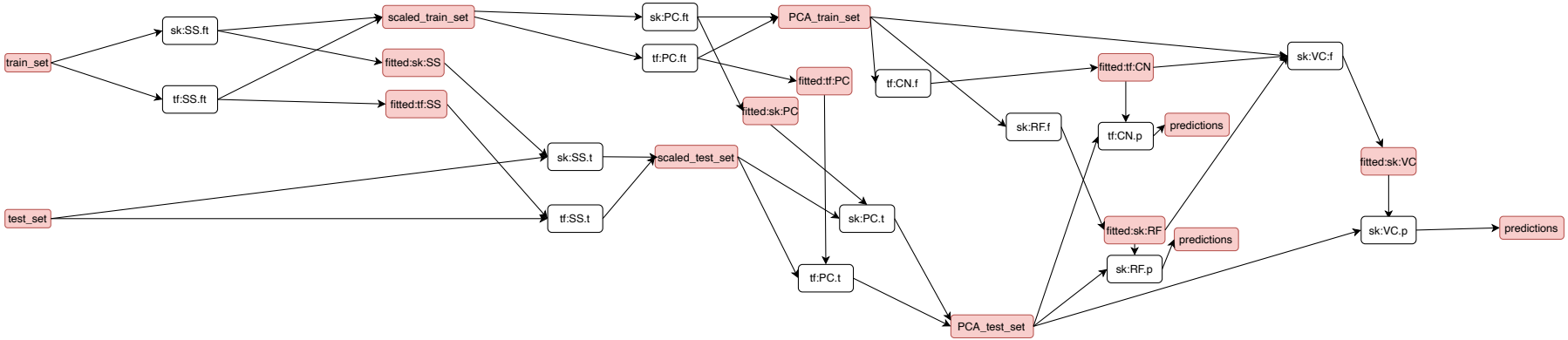
Artifacts

Task

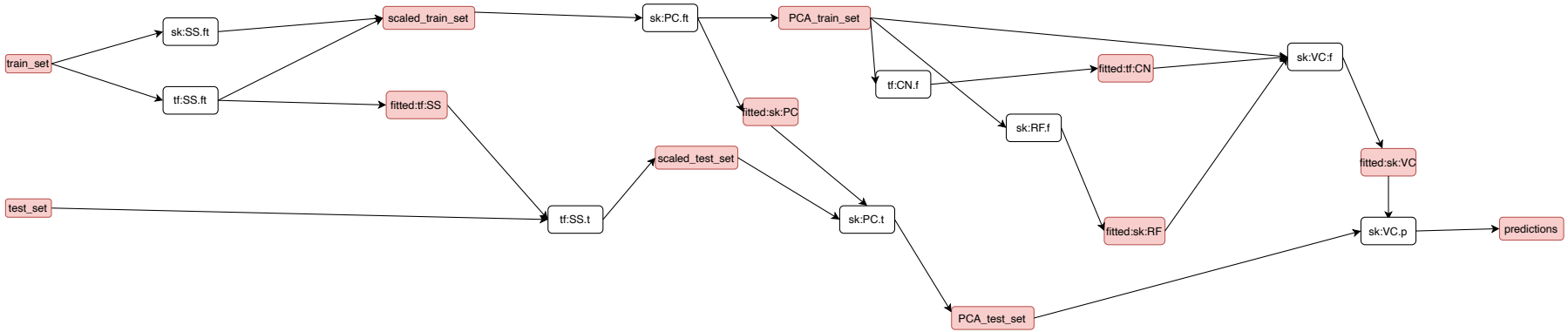
but we an create a unified logical graph



and a unified task-artifact graph



and finally a plan



Optimizing Data Science Workflows

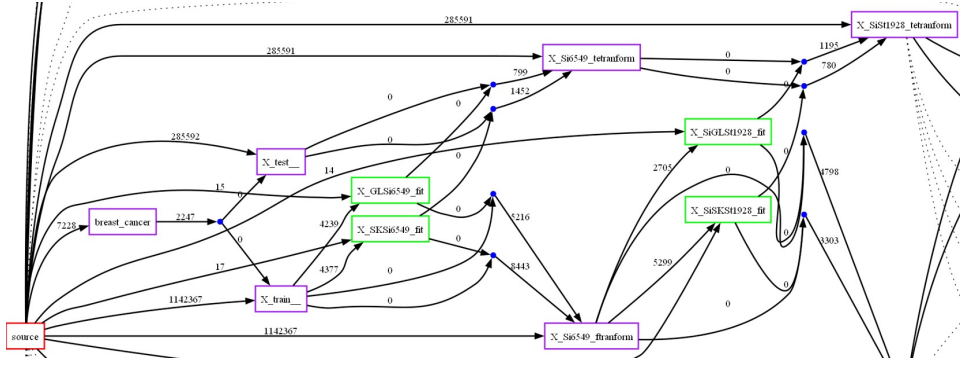
We can leverage:

- **Sharing Computation**: Identifying common subexpression in multiple pipelines so the results are only computed once
- **Materialization**: Storing results so they can be reused in the future
- **Equivalent Verification**: Discovering when the same results can be computed by different pipelines

Creating a Plan: there are trade-offs among when to share, what to materialize and which pipeline to choose.

task-artifact graphs can be huge

creating a plan is not easy



INFO-H420
Management of Data Science and Business Workflows

Part II
Data Privacy

Dimitris SACHARIDIS

2023-2024

Data Release

Data Release Goes Bad

NYC Taxi & Limousine Commission

Jessica Alba (actor)

Strava's fitness tracker heat map reveals the location of military bases

Geolocation isn't a new problem for the military

By Andrew Liptak | @AndrewLiptak | Jan 28, 2018, 3:51pm EST

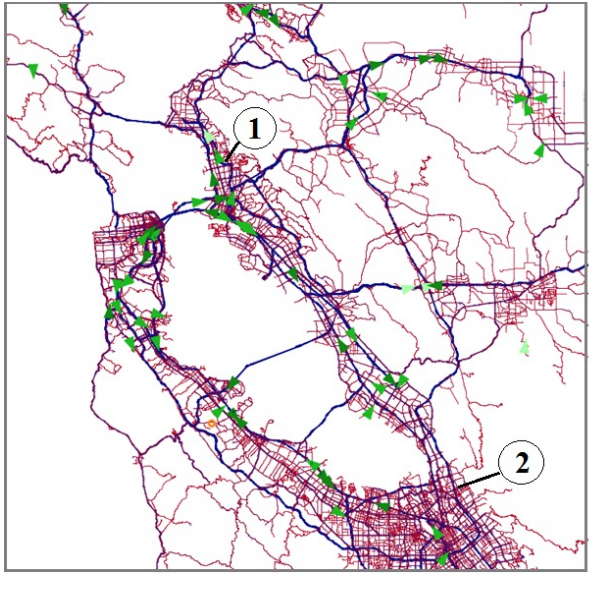
AOL

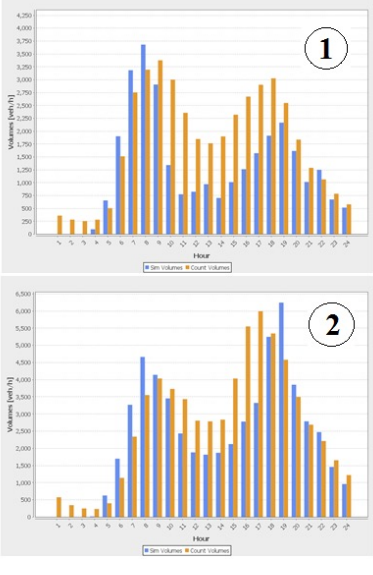
NETFLIX

Commonwealth of Massachusetts
Group Insurance Commission
Your Benefits Connection

We need to solve this data release problem …

But Why Release Data?





- Urban mobility modeling at scale
  - Smart traffic design is critical, congestion is getting worse

Big Data is Expected to Solve Big Problems

- The big data concept has taken off
  - Many organizations have adopted it
  - Entered the public vocabulary

- But this data is mostly about individuals
  - Individuals want privacy for their data

- How can researchers work on sensitive data?
  - The easy answer: anonymize it and share
  - The problem: we don't know how to do this

Data Release

- Several agencies, institutions, bureaus, organizations make (sensitive) data involving people publicly available
  - termed microdata (vs. aggregated macrodata) used for analysis
  - often required and imposed by law

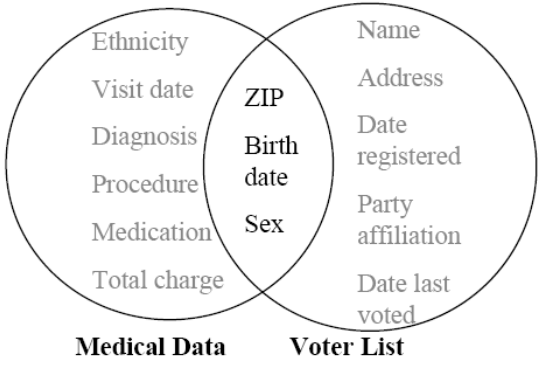
- To protect privacy microdata are anonymized
  - explicit identifiers (SSN, name, phone #) are removed

- is this sufficient for preserving privacy?
- no! susceptible to link attacks
  - publicly available databases (voter lists, city directories) can reveal the "hidden" identity

Link attack example

- Sweeney [S02a] managed to re-identify the medical record of the governor of Massachussetts
  - MA collects and publishes anonymized medical data for state employees (microdata) left circle
  - voter registration list of MA (publicly available data) right circle

- looking for governor's record
- join the tables:
  - 6 people had his birth date
  - 3 were men
  - 1 in his zipcode



- regarding the US 1990 census data
  - 87% of the population are unique based on (zipcode, gender, dob)

[S02a] Sweeney, L. *k*-Anonymity: A Model for Protecting Privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 2002.

Privacy in Data Publishing

- the role of attributes in published data
  - explicit identifiers are removed
  - quasi identifiers (QI) can be used to re-identify individuals
  - sensitive attributes (may not exist!) carry sensitive information

identifier	quasi identifiers			sensitive
Name	**Birthdate**	**Sex**	**Zipcode**	**Disease**
Andre	21/1/79	male	53715	Flu
Beth	10/1/81	female	55410	Hepatitis
Carol	1/10/44	female	90210	Bronchitis
Dan	21/2/84	male	02174	Sprained Ankle
Ellen	19/4/72	female	02237	AIDS

goal of privacy preservation (rough definition)
de-associate individuals from sensitive information

k-anonymity

- preserve privacy via k-anonymity, proposed by Sweeney and Samarati [S01, S02a, S02b]
- k-anonymity: intuitively, hide each individual among k-1 others
  - each QI set of values should appear at least k times in the released microdata
  - linking cannot be performed with confidence > 1/k
  - sensitive attributes are not considered

- how to achieve this?
  - generalization and suppression
  - value perturbation is not considered (we should remain *truthful* to original values)

- privacy vs utility tradeoff
  - do not anonymize more than necessary

[S01] Samarati, P. Protecting Respondents' Identities in Microdata Release. IEEE TKDE, 13(6):1010-1027, 2001.
[S02a] Sweeney, L. *k*-Anonymity: A Model for Protecting Privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 2002.
[S02b] Sweeney, L. *k*-Anonymity: Achieving *k*-Anonymity Privacy Protection using Generalization and Suppresion. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 2002.

k-anonymity example

tools for anonymization

- generalization
  - publish more general values, i.e., given a domain hierarchy, roll-up
- suppression
  - remove tuples, i.e., do not publish outliers
  - often the number of suppressed tuples is bounded

original data

Birthdate	Sex	Zipcode
21/1/79	male	53715
10/1/79	female	55410
1/10/44	female	90210
21/2/83	male	02274
19/4/82	male	02237

2-anonymous data

	Birthdate	Sex	Zipcode
group 1	*/1/79	person	5****
	*/1/79	person	5****
suppressed	1/10/44	female	90210
group 2	*/*/8*	male	022**
	//8*	male	022**

k-anonymity problems

- homogeneity attack: in the last group everyone has cancer
- background knowledge: in the first group, Japanese have low chance of heart disease
- we need to consider the sensitive values

Original data

id	Zipcode	Sex	National.	Disease
1	13053	28	Russian	Heart Disease
2	13068	29	American	Heart Disease
3	13068	21	Japanese	Viral Infection
4	13053	23	American	Viral Infection
5	14853	50	Indian	Cancer
6	14853	55	Russian	Heart Disease
7	14850	47	American	Viral Infection
8	14850	49	American	Viral Infection
9	13053	31	American	Cancer
10	13053	37	Indian	Cancer
11	13068	36	Japanese	Cancer
12	13068	35	American	Cancer

4-anonymous data

id	Zipcode	Sex	National.	Disease
1	130**	<30	*	Heart Disease
2	130**	<30	*	Heart Disease
3	130**	<30	*	Viral Infection
4	130**	<30	*	Viral Infection
5	1485*	≥40	*	Cancer
6	1485*	≥40	*	Heart Disease
7	1485*	≥40	*	Viral Infection
8	1485*	≥40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

l-diversity (ell diversity)

- make sure each group contains well represented sensitive values
  - protect from homogeneity attacks
  - protect from background knowledge

l-diversity (simplified definition)
a group is l-diverse if the most frequent sensitive value appears at most 1/l times in group

- l-diversity definition has monotonicity properties similar to k-anonymity
  - Easy to adapt existing algorithms to count frequencies of sensitive values

when value of k and l is increased, privacy increases

Differential Privacy

Differential Privacy

- Differential privacy protects the privacy of individuals by **adding noise** or perturbation in a **controlled** way
- Goal: to make it **difficult to determine the presence or absence of any particular individual**

- It is a mathematical framework that provides a way to **measure** the **privacy loss** of a given dataset
- It allows for the creation of **algorithms** that can **analyze datasets** while still preserving the privacy of individuals within the dataset
- It provides a way to **balance** the **trade-off** between **utility** and **privacy**

how truthful is information

Differential Privacy

A randomized algorithm K satisfies ε-differential privacy if:
Given two data sets that differ by one individual,
D and D', any property S, and $\varepsilon > 0$:

$$\Pr[\, K(D) \in S] \leq e^{\varepsilon} \Pr[\, K(D') \in S]$$

- Can achieve ε-DP for counts by adding a random noise value

- Uncertainty due to noise → plausible deniability
  - Hides whether someone is present in the data

Achieving ε-DP

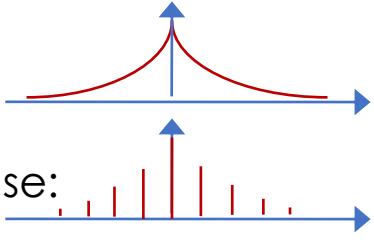
(Global) Sensitivity of publishing:

$s = \max_{D,D'} |F(D) - F(D')|$, D, D' differ by 1 individual

E.g., count individuals satisfying property P: $s = 1$

For every value that is output:

- Add Laplacian noise $Lap(s/\varepsilon)$:
- Or Geometric noise for discrete case:



Simple rules for composition of differentially private outputs:

Given output O_1 that is ε_1-DP and O_2 that is ε_2-DP

- (Sequential composition) If inputs overlap, result is $(\varepsilon_1 + \varepsilon_2)$-DP
- (Parallel composition) If inputs disjoint, result is $\max(\varepsilon_1, \varepsilon_2)$-DP

Trying to Reduce Trust

- Most work on differential privacy assumes a trusted party
  - Data aggregator (e.g., organizations) that sees the true, raw data
  - Can compute exact query answers, then perturb for privacy

- A reasonable question: can we reduce the amount of trust?
  - Can we remove the trusted party from the equation?
  - Users produce locally private output, aggregate to answer queries

- One approach is to use secure multiparty computation or homomorphic encryption encrypt data and then perform some computation
  - Merge encrypted data, and add noise for privacy inside encryption
  - Complex to get right, and very high computational overhead

Local Differential Privacy

Local Differential Privacy - Non centralized approach
Global Differential Privacy - Centralized approach

- What about having each user run a DP algorithm on their data?
  - Then combine all the results to get a final answer

- On first glance, this idea seems crazy
  - Each user adds noise to mask their own input
  - So surely the noise will always overwhelm the signal?

- But … noise can cancel out or be subtracted out
  - We end up with the true answer, plus noise which can be smaller
  - However, noise is still larger than in the centralized case

Local Differential Privacy: Example

- Each of N users has 0/1 value, estimate total population sum
  - Each user adds independent Laplace noise: mean 0, variance $2/\varepsilon^2$

- Adding user results: true answer + sum of N Laplace distributions
  - Error is random variable, with mean 0, variance $2N/\varepsilon^2$
  - Confidence bounds: ~95% chance of being within 2σ of the mean
  - So, error looks like $\sqrt{N}/\varepsilon$, but true value may be proportional to N

- Numeric example: suppose true answer is $N/2$, $\varepsilon = 1$, $N = 1M$
  - We see $500K \pm 2800$: about 1% uncertainty
  - Error in centralized case would be close to 1 (0.001%)

Local Differential Privacy

- We can achieve LDP, and obtain reasonable accuracy (for large N)
  - The error typically scales with $\sqrt{N}$
- Generic approach: apply centralized DP algorithm to local data
  - But error might still be quite large
  - Unclear how to merge private outputs (e.g., private clustering)
- So, we seek to design new LDP algorithms
  - Maximize the accuracy of the results
  - Minimize the costs to the users (space, time, communication)
  - Ensure that there is an accurate algorithm for aggregation

Privacy with a coin toss: Randomized Response

- Each user has a single bit of private information
  - Encoding e.g. political/sexual/religious preference, illness, etc.

- Randomize Response (RR): toss an unbiased coin [Warner 65]
  - If Heads (probability $p = ½$), report the true answer
  - Else, toss unbiased coin again: if Heads, report true answer, else lie

- Collect responses from N users, subtract noise
  - Error in the estimate is proportional to $1/\sqrt{N}$
  - Simple LDP algorithm with parameter $\varepsilon=\ln((¾)/(¼)) = \ln(3)$
  - Generalization: allow biased coins ($p \neq ½$)

Privacy in practice

- Differential privacy based on coin tossing is widely deployed!
  - In Google Chrome browser, to collect browsing statistics
  - In Apple iOS and MacOS, to collect typing statistics
  - In Microsoft Windows to collect telemetry data over time
  - This yields deployments of over 100 million users each

- All deployments are based on RR, but extend it substantially
  - To handle the large space of possible values a user might have

- Local Differential Privacy is state of the now
  - Randomized response invented in 1965: six decades ago!

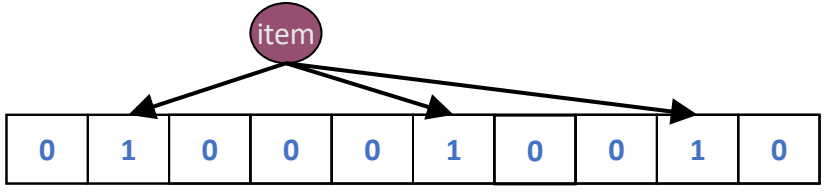
Google's RAPPOR

- Each user has one value out of a very large set of possibilities
  - E.g., their favourite URL, www.nytimes.com

- Attempt 1: run RR for all values: google.com? Bing.com? …
  - User has sparse binary vector with one 1: run RR on every bit
  - Satisfies 2ε-LDP: [recall ε=ln(3)] change user's choice, 2 bits change: 1 → 0, 0 → 1
  - Slow: sends 1 bit for every possible index in the vector
  - Limited: can't handle new possibilities being added

- Try to do better by reducing domain size through hashing

Bloom Filters

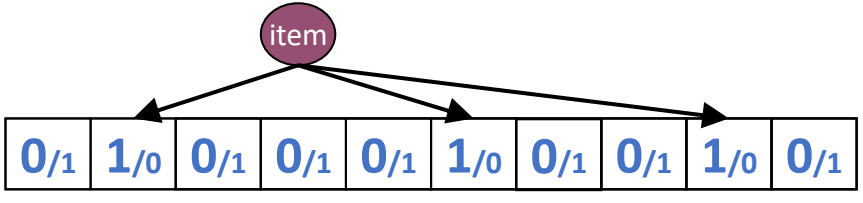


- Bloom filters [B70] compactly encode set membership
  - k hash functions map items to m-bit vector k times
  - Update: Set all k entries to **1** to indicate item is present
  - Query: Can lookup items, store set of size n in O(n) bits
  - Analysis: choose k and m to obtain small false positive probability

false positive possible
false negative - not possible

- Can be merged by OR-ing vectors (of same size)
  - Duplicate insertions do not change Bloom filters

Counting Bloom Filters & Randomized Response



- Idea: apply Randomized Response to the bits in a Bloom filter
  - Not too many bits in the filter compared to all possibilities
- Each user maps their input to at most k bits in the filter
  - Privacy guarantee with parameter $2k\varepsilon$
- Combine all user reports and observe how often each bit is set

Decoding Noisy Counting Bloom Filters

- We have a noisy Bloom filter, where each bit has a probability

- To estimate the frequency of a particular value:
  - Look up its bit locations in the Bloom filter
  - Compute the unbiased estimate of the probability each is 1
  - Take the minimum of these estimates as the frequency
  - More advanced decoding heuristics to decode all at once

- How to find frequent strings without knowing them in advance?
  - Subsequent work: build up frequent strings character by character

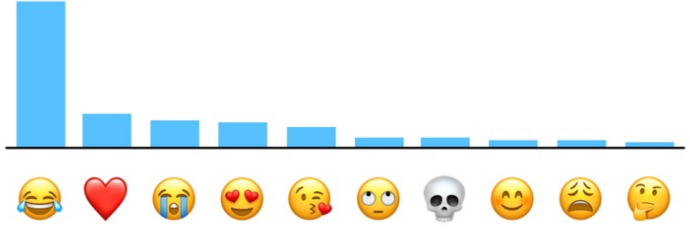
RAPPOR in practice

- The RAPPOR approach is implemented in the Chrome browser
  - Collects data from opt-in users, tens of millions per day
  - Open-source implementation available

- Tracks settings in the browser, e.g., home page, search engine
  - Many users unexpectedly change home page → possible malware

- Typical configuration:
  - 128 bit Bloom filter, 2 hash functions, privacy parameter ~0.5
  - Needs about 10K reports to identify a value with confidence

Apple's Differential Privacy in Practice



The Count Mean Sketch technique allows Apple to determine the most popular emoji to help design better ways to find and use our favorite emoji. The top emoji for US English speakers contained some surprising favorites.

- Apple uses their system to collect data from iOS and MacOS users
  - Popular emojis: (heart) (laugh) (smile) (crying) (sadface)
  - "New" words: bruh, hun, bae, tryna, despacito, mayweather
  - Which websites to mute, which to autoplay audio on!

- Deployment settings: w=1000, d=1000, ε=2-8 (some criticism)

Acknowledgements

- DP content from "*Privacy at Scale: Local Differential Privacy in Practice*" Tutorial in SIGMOD 2018 by Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang

INFO-H420
Management of Data Science and Business Workflows

Part II
Fairness

Dimitris SACHARIDIS

2023-2024

How do people feel about interacting with AI systems?

survey of 5,000 consumers by Pegasystems about people interacting with Artificial Intelligence (AI) systems

- 34% say they interact with AI systems
- machines **not trustworthy**
  - only 9% very comfortable interacting
- machines **can be biased**
  - 53% say it's possible for systems to show bias in their decisions
- machines **cannot be moral**
  - 56% don't believe it is possible to develop systems that behaves morally

How comfortable are you/would you be with a business using Artificial Intelligence to interact with you?

Responsible Data Science

- Responsible data science is the practice of using data and data-driven techniques in a way that is **ethical**, **transparent**, and **respectful of the rights** and interests of **individuals** and **society**

- Key concepts:
  - **Data privacy** and security, to protect sensitive data of individuals
  - **Transparency** about data sources, methods, and limitations
  - Exhibit desirable **ethical** principles
  - Put **human in control**, allow them to understand and control

ChatGPT, December 2022

Responsible AI/ML/Data Science

- Algorithmic Decisions (e.g., from ML/AI models) are ubiquitous, permeate many aspects of **society**
- more and more questions of **responsible use** arise

  Fairness Accountability Transparency Ethics

- dedicated conferences
  - ACM Conference on Fairness, Accountability, and Transparency (FAccT)
  - AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society (AIES)
- topics appear in almost all CS venues
- courses: in Berkeley, Cornell, Princeton
- EU high-level expert group on AI: Ethics Guidelines for Trustworthy AI
- EU upcoming regulation: AI Act

Irresponsible AI

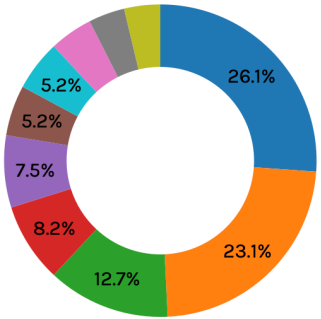
Welcome to the

AI Incident Database

Search over 2000 reports of AI harms

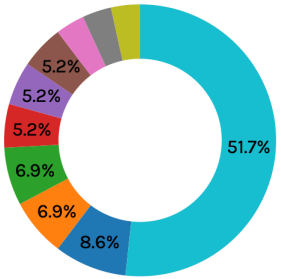
Search | Discover

Harm Distribution Basis



none, race, sex, nation of origin, citizenship, immigrant status, religion, sexual orientation or gender identity, geography, financial means, disability, All Others

Known AI Goal



Chatbot, Autonomous Driving, Visual Art Generation, Hate Speech Detection, Question Answering, Automatic Skill Assessment, Market Forecasting, Face Recognition, Deepfake Video Generation, All Others

https://incidentdatabase.ai

Ethical Principles

- review of 84 AI ethics guidelines
- trust is the "end goal"

Ethical principle	Number of documents	Included codes
Transparency	73/84	Transparency, explainability, explicability, understandability, interpretability, communication, disclosure, showing
Justice & fairness	68/84	Justice, fairness, consistency, inclusion, equality, equity, (non-)bias, (non-)discrimination, diversity, plurality, accessibility, reversibility, remedy, redress, challenge, access and distribution
Non-maleficence	60/84	Non-maleficence, security, safety, harm, protection, precaution, prevention, integrity (bodily or mental), non-subversion
Responsibility	60/84	Responsibility, accountability, liability, acting with integrity
Privacy	47/84	Privacy, personal or private information
Beneficence	41/84	Benefits, beneficence, well-being, peace, social good, common good
Freedom & autonomy	34/84	Freedom, autonomy, consent, choice, self-determination, liberty, empowerment
Trust	28/84	Trust
Sustainability	14/84	Sustainability, environment (nature), energy, resources (energy)
Dignity	13/84	Dignity
Solidarity	6/84	Solidarity, social security, cohesion

[2019 Nature Machine Intelligence A. Jobin et al.] *Artificial Intelligence: the global landscape of ethics guidelines*

Fairness

Bias in Data-Driven Decisions

- COMPAS **risk assessment** software by Northpointe
- estimates the likelihood of a criminal to **reoffend**
  - based on a detailed question-based profiling of the people
- used by **judges** in the US to guide their decisions
  - about sentences, bail amounts

- ProPublica, a news organization, analyzed the predictions of COMPAS
- looked at more that 10,000 offenders
  - compared the predictions of COMPAS
  - with what actually happened (did they reoffend?)

https://incidentdatabase.ai/cite/40/

Bias in Data-Driven Decisions

results of the study:

- correctly predicts reoffending for black and white at the same rate
  - same **recall** (true positive rate, sensitivity)

but **mistakes** paint a different picture:

- blacks are almost **twice as likely** as whites to be labeled a **higher risk** but not actually reoffend
  - more **false positives** for blacks
- black are **much less likely** than whites to be labeled **lower risk**, but actually reoffend
  - less **false negatives** for blacks

Algorithmic Decisions

- main result of the study:

discrimination against blacks

- but there is some controversy to the study results
  - https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing
  - https://www.documentcloud.org/documents/2998391-ProPublica-Commentary-Final-070616.html
  - https://www.propublica.org/article/technical-response-to-northpointe
  - http://www.crj.org/assets/2017/07/9_Machine_bias_rejoinder.pdf

Algorithmic Decisions

What are the issues here?

- **fairness** – is the system actually fair to racial groups?
- **transparency** – we don't know what the algorithm does
- **accountability** – who is to hold responsible?

What is Fairness?

- Dictionary: "***the state of being free from bias or injustice***"

- Political Science: "*distributive justice discusses* ***fair allocation of resources*** *among diverse members of a community*"
  - "A Theory of Justice" by J. Rawls (American philosopher)
    - "justice as fairness"; "social cooperation should be fair to all citizens regarded as free and as equals"
  - but what is a *fair allocation*?
    - ***equality of outcome***: each person get the same amount
    - ***equality of opportunity***: equal grounds for competing for resources
    - ***social welfare***: what benefits the society the most

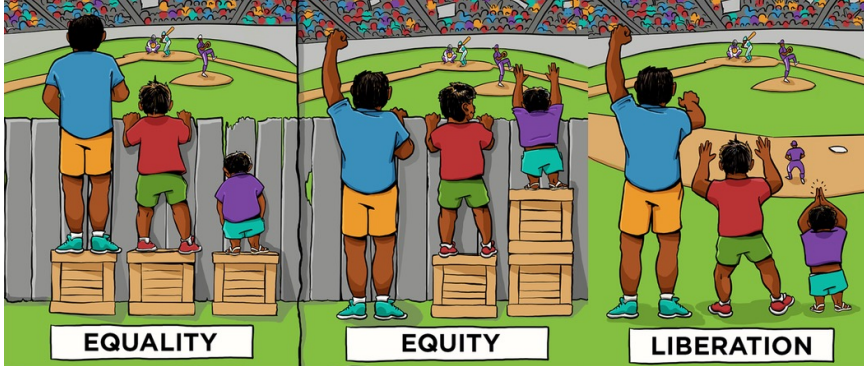
What is Fairness?

- Legal Systems: ***"fairness as non-discrimination"***
  - ***disparate treatment***: intentional discrimination on *protected groups* (defined on race, color, sex, etc.); not "*color-blind*"
    - e.g., only African American applicants are required to take a pre-employment assessment test
  - ***disparate impact***: a procedure that has disproportionate impact on protected groups
    - e.g., all applicants are tested but only African Americans are eliminated based on the results of the assessment.
  - ***affirmative action***: promote non-discrimination and support historically disadvantaged groups; *quota systems*
    - e.g., to address gender imbalance in STEM

Algorithmic Fairness

- Data-Driven Systems often make **decisions** on behalf of humans
  - high-stake decisions: likelihood to reoffend; grant a loan application
  - innocuous decisions: which article to read; what to buy in a store
    - or not so innocuous? fake news, filter bubbles

Algorithmic Fairness: machine made decisions should *not discriminate* against individuals



Group vs Individual Fairness

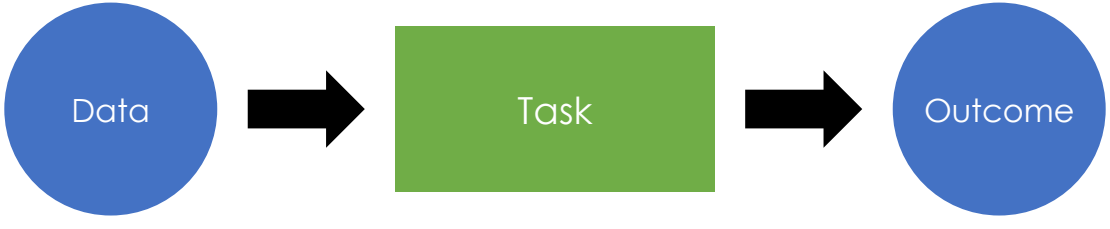
Group Fairness

- a specific **protected** group defined by a **sensitive** attribute (e.g., race, gender) should be treated/impacted similarly to the other (non-protected) groups
    - typically two groups protected (disadvantaged) and non-protected (advantaged)

Individual Fairness

- a specific **individual** should be treated/impacted similarly **to a similar other** individual

Classification based on Tasks



Fairness depends on the type of task and its outcomes

- Classification
- Ranking
- Recommendation

Fairness in Classification

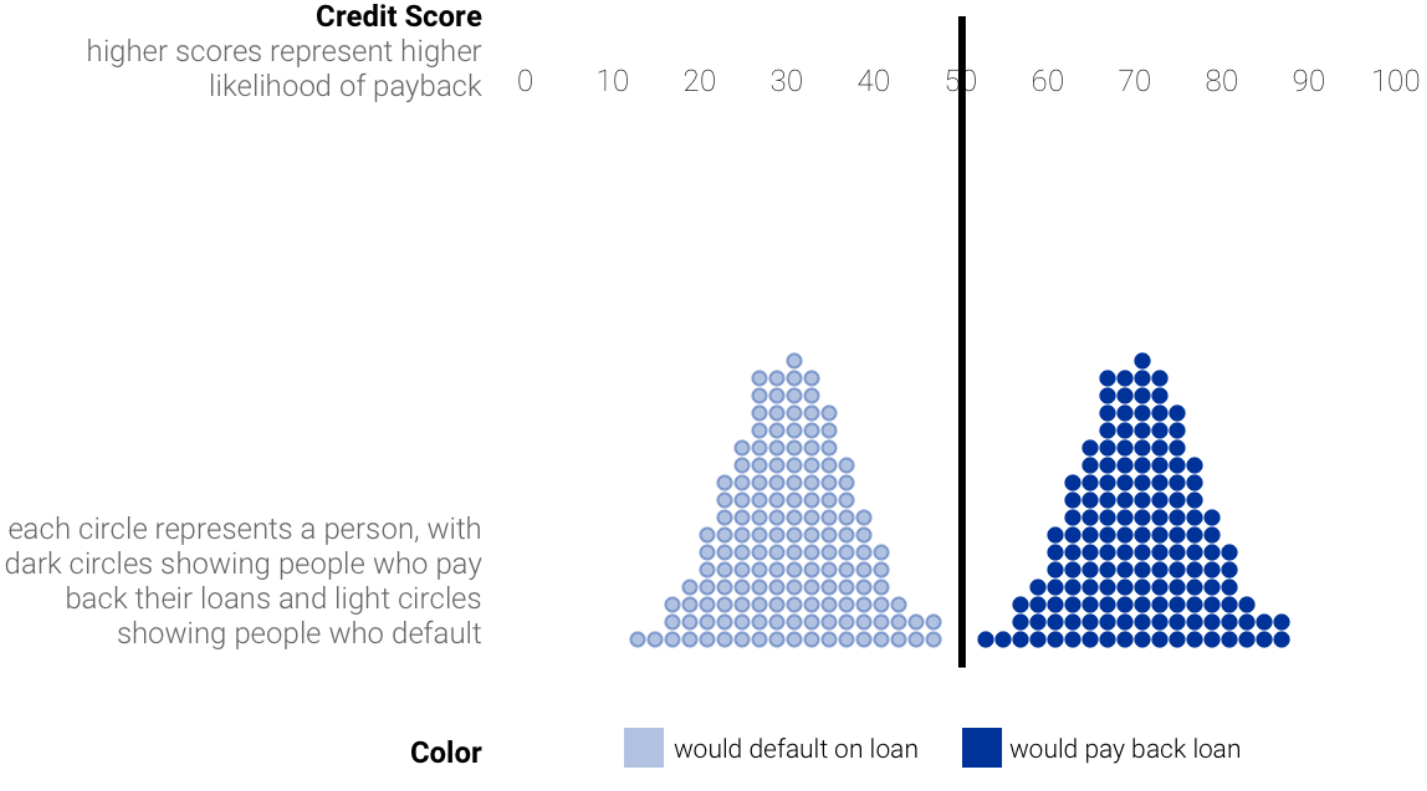
Classification

- assume **binary** classification (positive or negative classes)
  - e.g., accept a loan
- assume group fairness: **protected** vs. **non-protected** group

- assume that some **score** predicts the likelihood of being in the positive class
  - e.g., a credit score

- a score **threshold** determines the classification outcome
  - e.g., scores above threshold mean positive

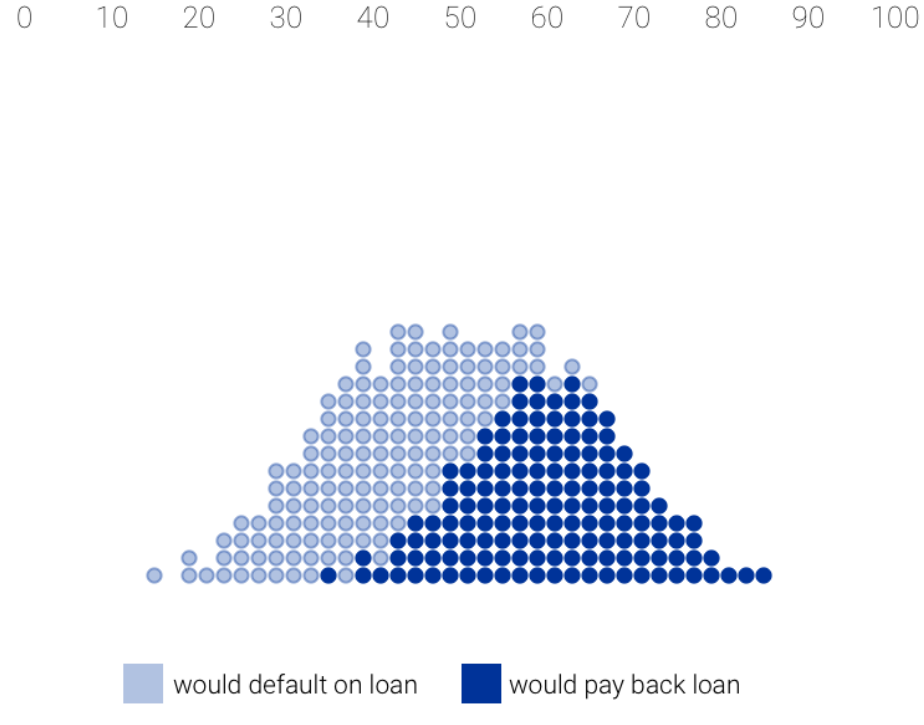
How should we set the threshold?



http://research.google.com/bigpicture/attacking-discrimination-in-ml/

How about now?

Credit Score
higher scores represent higher likelihood of payback

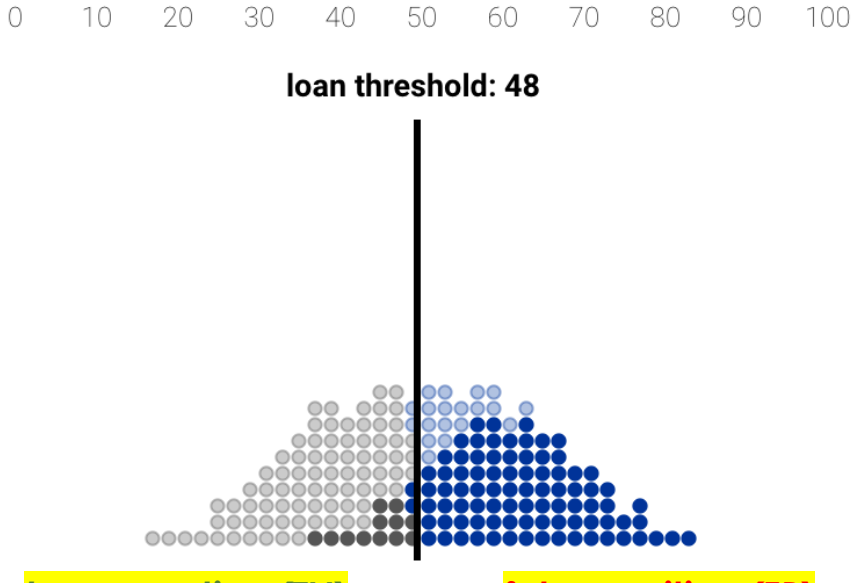


each circle represents a person, with dark circles showing people who pay back their loans and light circles showing people who default

Color

Recall -> % of positive tuples did the classifier label as positive TP/(TP+FN)
Precision -> what is the % of tuples that the classifier labelled as positive are actually positive TP/(TP+FP)

How should we set the threshold?



true negative (TN)

false positive (FP)

denied loan / would default

granted loan / defaults

denied loan / would pay back

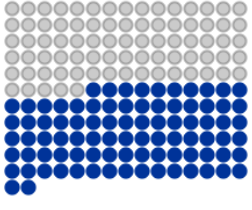
granted loan / pays back

false negative (FN)

true positive (TP)

accuracy

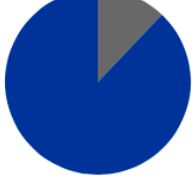
Correct 84%
loans granted to paying applicants and denied to defaulters



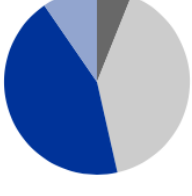
Incorrect 16%
loans denied to paying applicants and granted to defaulters

recall

True Positive Rate88%
percentage of paying applications getting loans

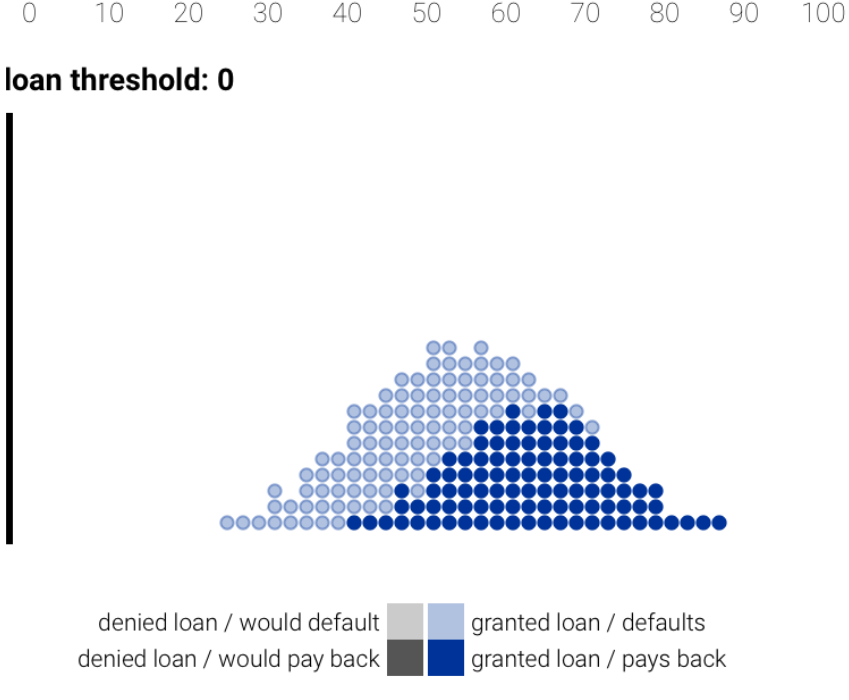


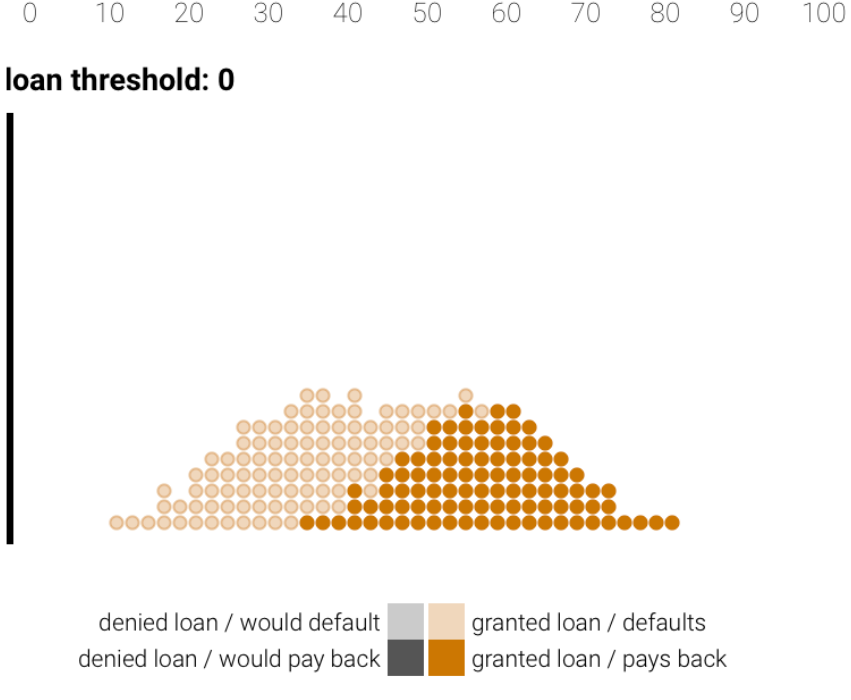
Positive Rate54%
percentage of all applications getting loans



Two groups: Blue and Orange

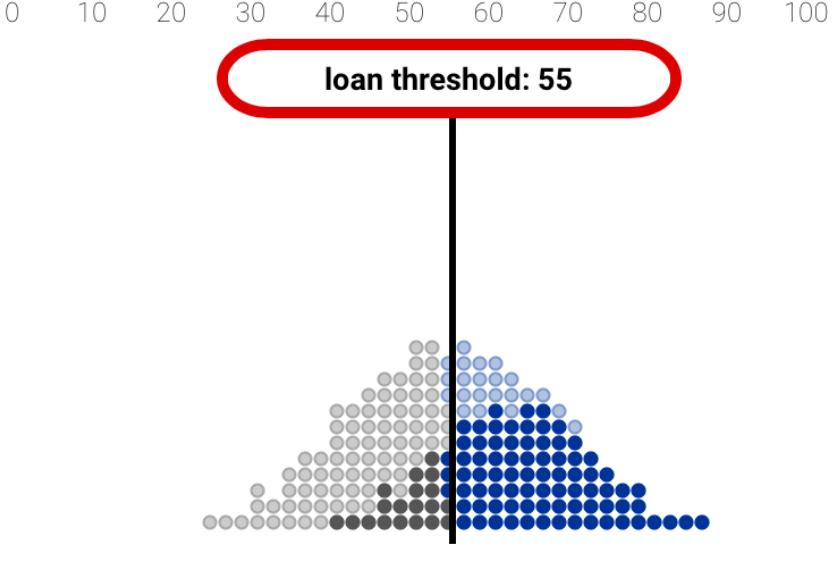
How should we set the thresholds for the two groups?

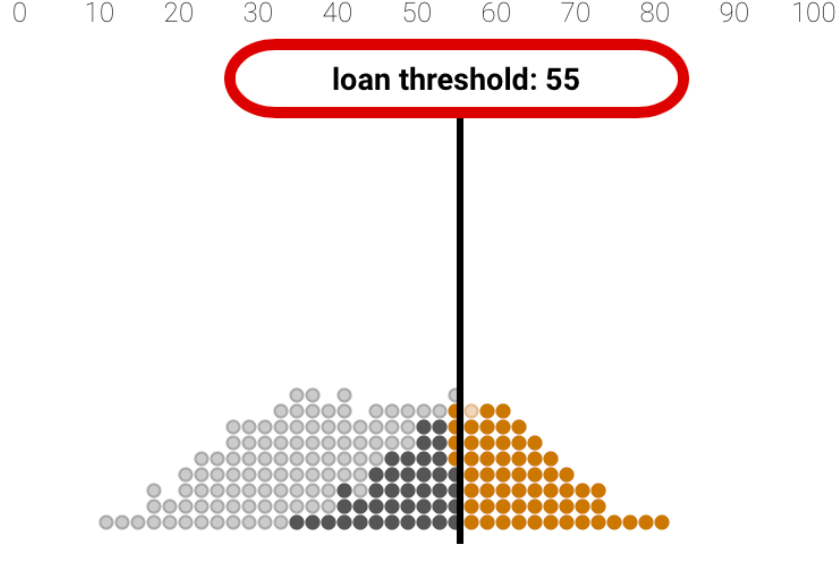




Equal Treatment (Color-Blindness)

We don't look at color, set the **same threshold.**



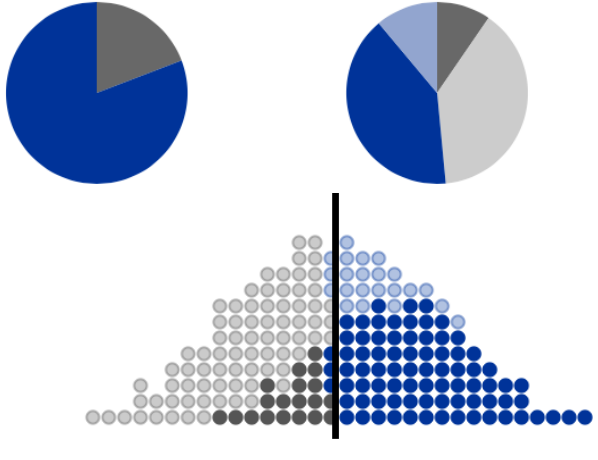


Equal Treatment (Color-Blindness)

We don't look at color, set the **same threshold.**

True Positive Rate81%
percentage of paying applications getting loans

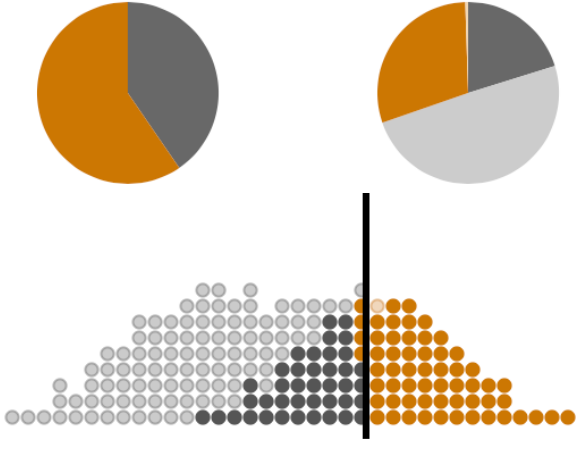
Positive Rate52%
percentage of all applications getting loans



denied loan / would default
denied loan / would pay back
granted loan / defaults
granted loan / pays back

True Positive Rate60%
percentage of paying applications getting loans

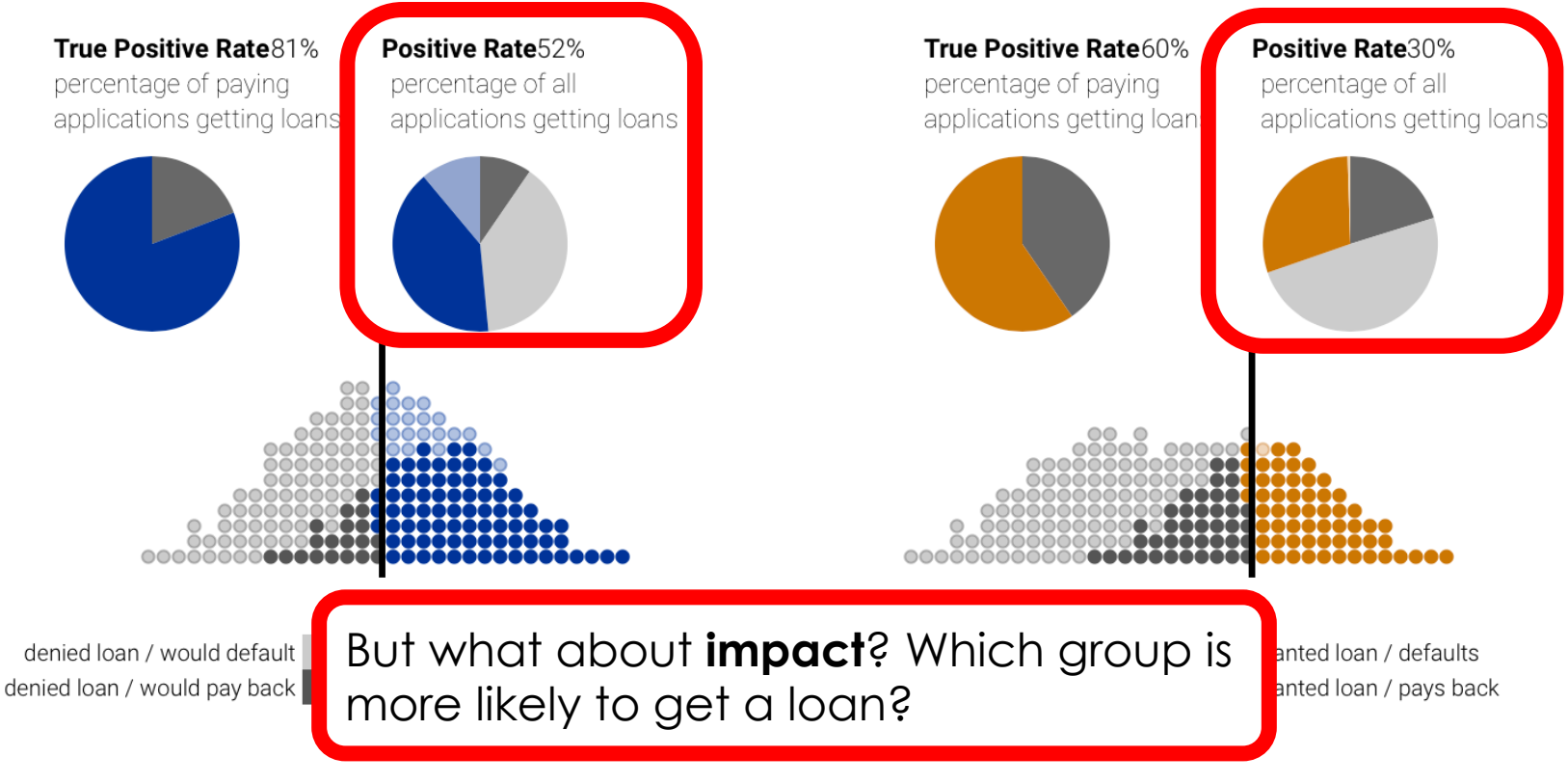
Positive Rate30%
percentage of all applications getting loans



denied loan / would default
denied loan / would pay back
granted loan / defaults
granted loan / pays back

Equal Treatment (Color-Blindness)

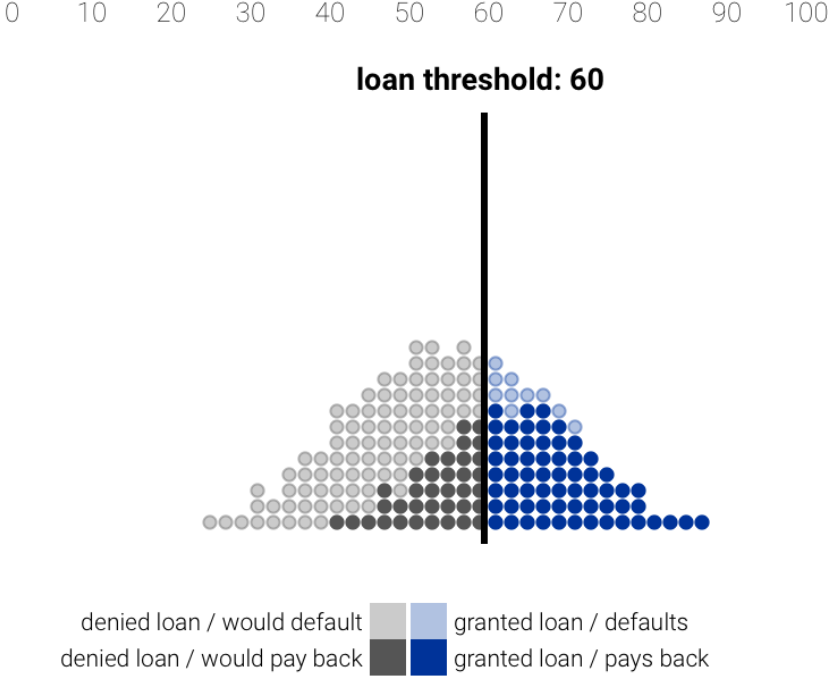
We don't look at color, set the **same threshold.**

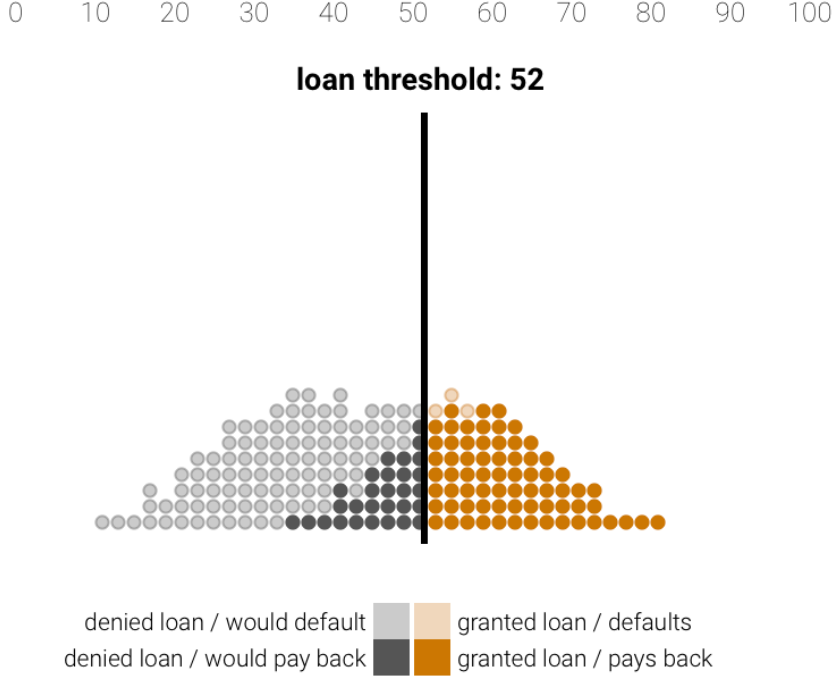


But what about **impact**? Which group is more likely to get a loan?

Group (Demographic/Statistical) Parity

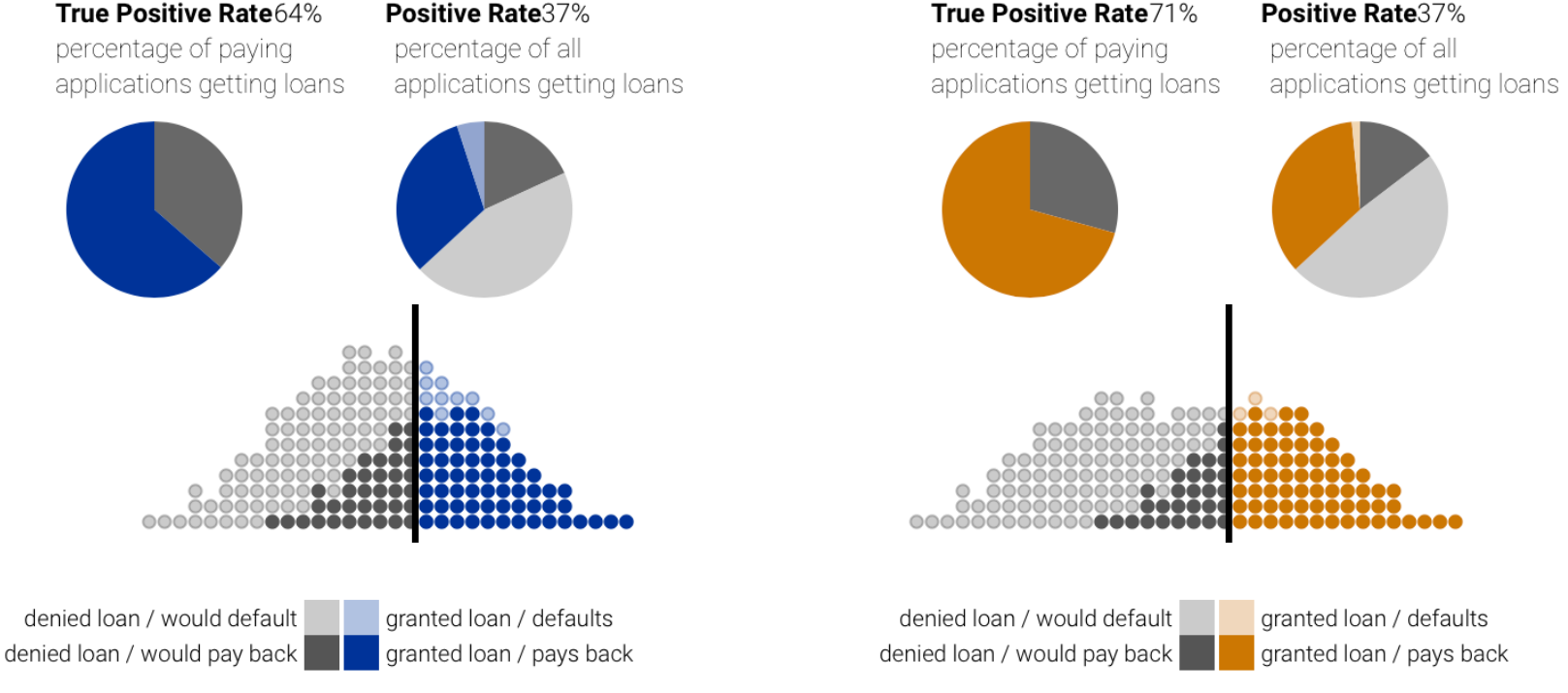
Set **different thresholds** so that groups have equal chance of getting a loan (*equal impact*)





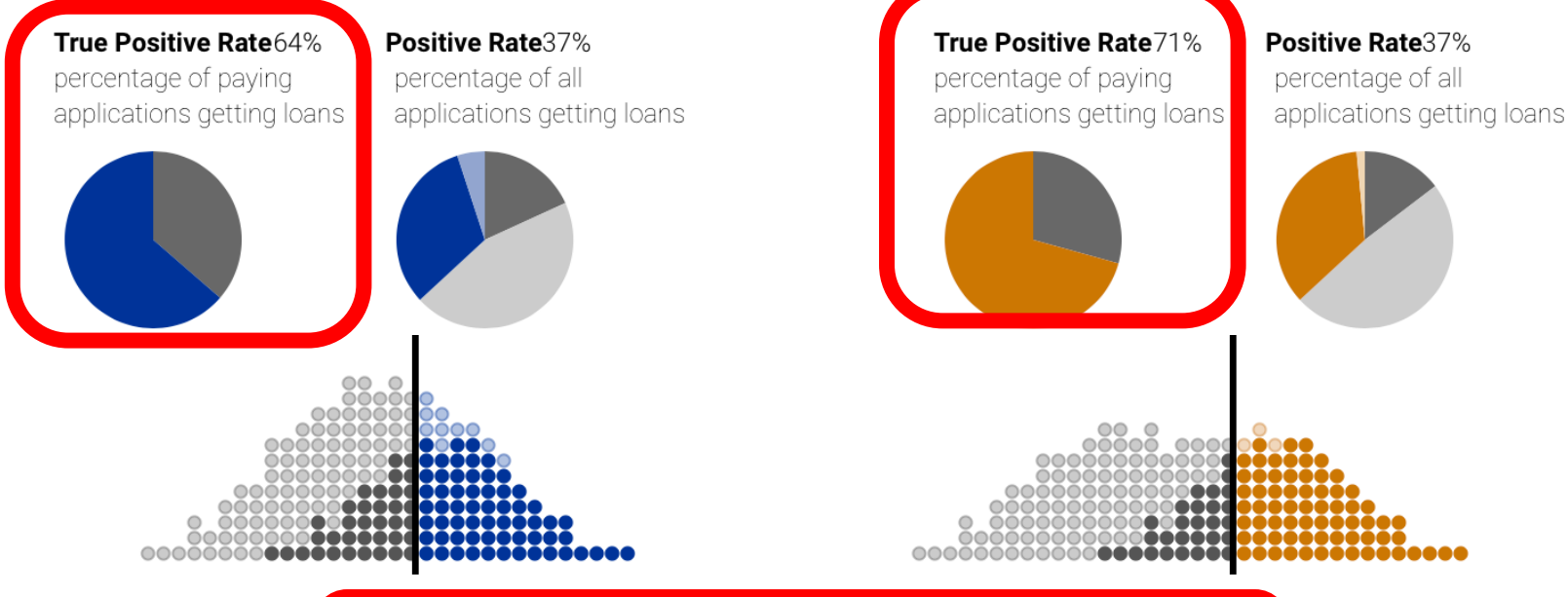
Group (Demographic/Statistical) Parity

Set **different thresholds** so that groups have equal chance of getting a loan (*equal impact*)



Group (Demographic/Statistical) Parity

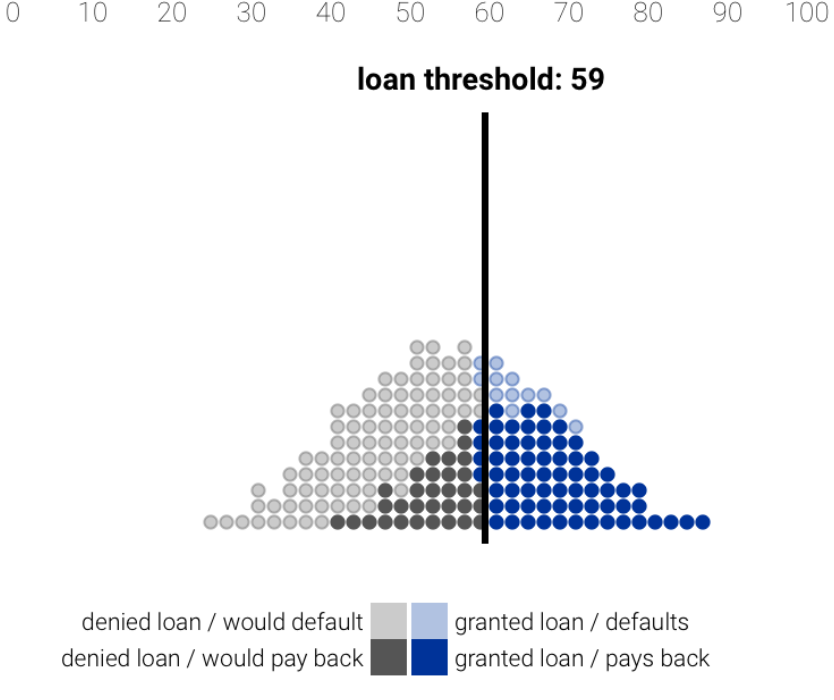
Set **different thresholds** so that groups have equal chance of getting a loan *(equal impact)*

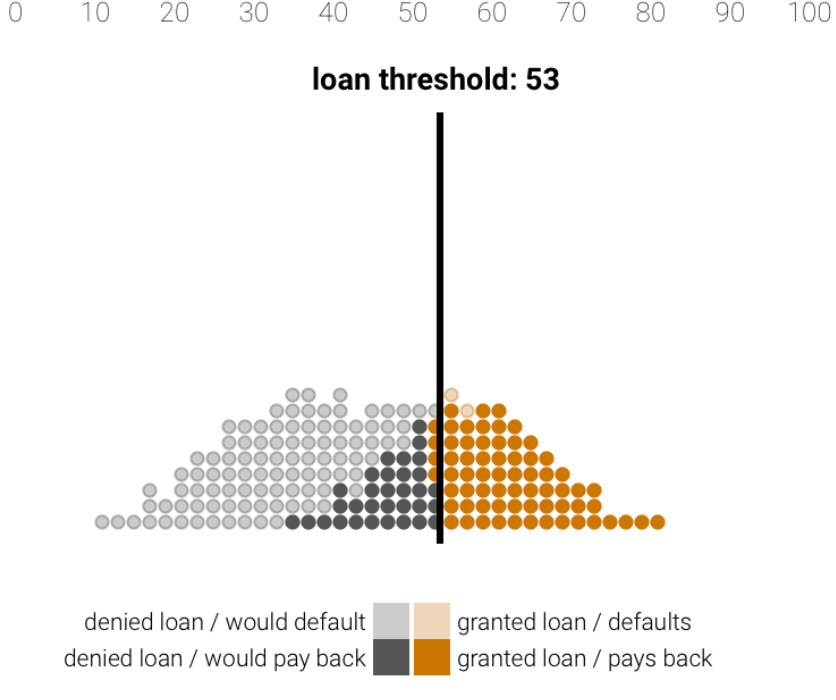


But look at those who would pay back the loan; are they impacted the same?

Equal Opportunity

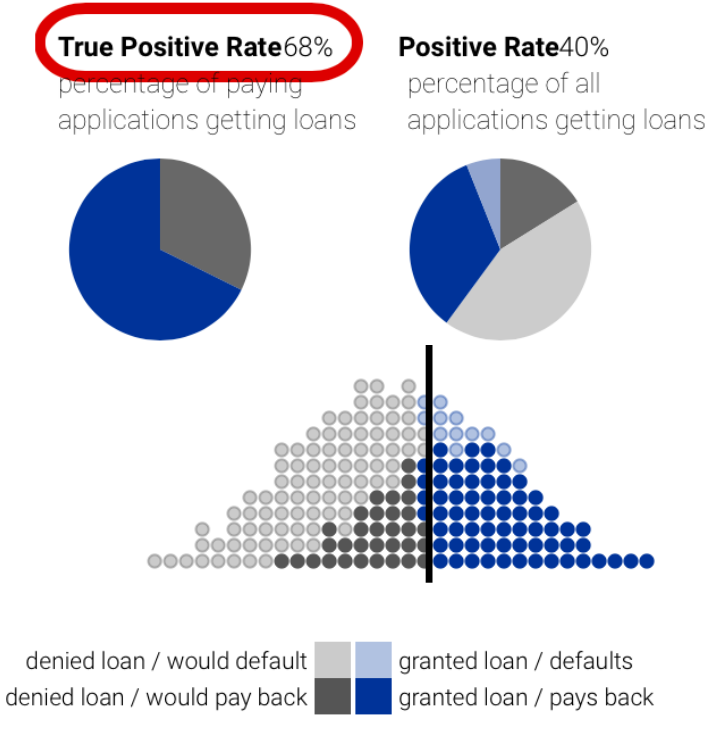
Aim to be fair for those that would pay back the loan

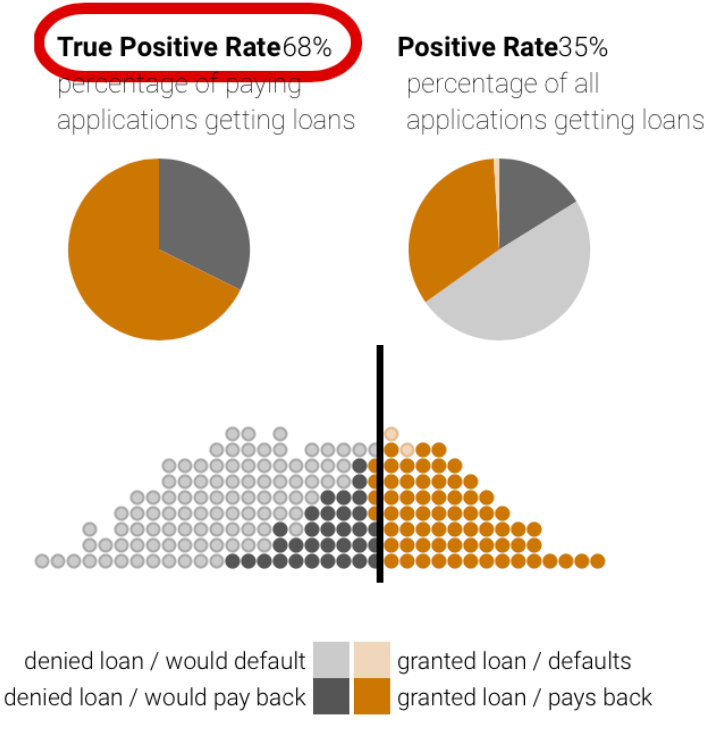




Equal Opportunity

Aim to be fair for those that would pay back the loan





Fairness Notions

- **equal treatment** (color-blindness, fairness by unawareness): same threshold
- **group parity** (demographic or statistical parity): same positive rate
- **equal opportunity**: same true positive rate (recall)

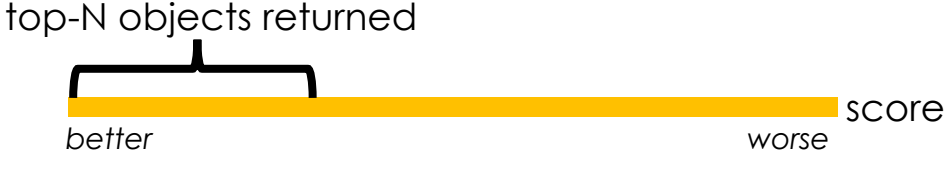
- you cannot have it all!
- there are various impossibility theorems related to fairness in classification

Fairness in Ranking

Ranking

- algorithm computes some **score** that encodes/predicts *goodness*
  - e.g., **merit**, relevance
- we want to **rank** objects (usually people) based on that score
  - e.g., candidates for a job position

- similarity with classification task: algorithm computes scores
- difference with classification task: there is **no threshold**; just the length (cut-off) of the ranked list



- so to achieve fairness we can only change the **scoring function**

Group Parity

- how does **group parity** translate in ranking?
- remember for classification: group parity = achieve equal positive rate in each group
  - positive (predicted as relevant) = you appear in the ranked list

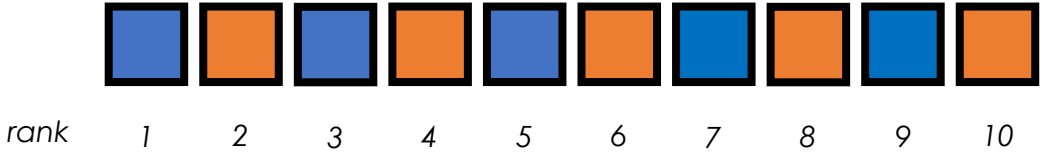
- so if we look at the top-N objects:
- group parity = *half* from protected, *half* from non-protected

example: rank 10 people from two groups

- is this fair? how can we fix it?

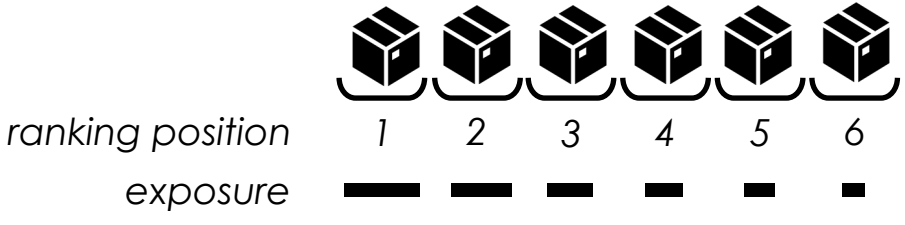
Group Parity

- stronger requirement: **for every prefix** of the ranked list, there should be group parity



- generalize a bit: assume we have a **target ratio**/mix of groups
  - e.g., 40% blue, 60% orange
- how can you achieve this?

Exposure in rankings



- a ranked list **exposes** items to the user
- the amount of exposure depends on the **ranking position**
  - top positions give higher exposure (aka *position bias*)

- the exposure of an item **does not depend** on relevance, utility, user satisfaction, etc.; just on ranking position

Fairness of Exposure

- a ranking **exposes** objects at varying degrees
  - **position bias**: higher ranked items are exposed more

mero lagi 10th page relevant cha bhane tyo first ma dekhauna paryo

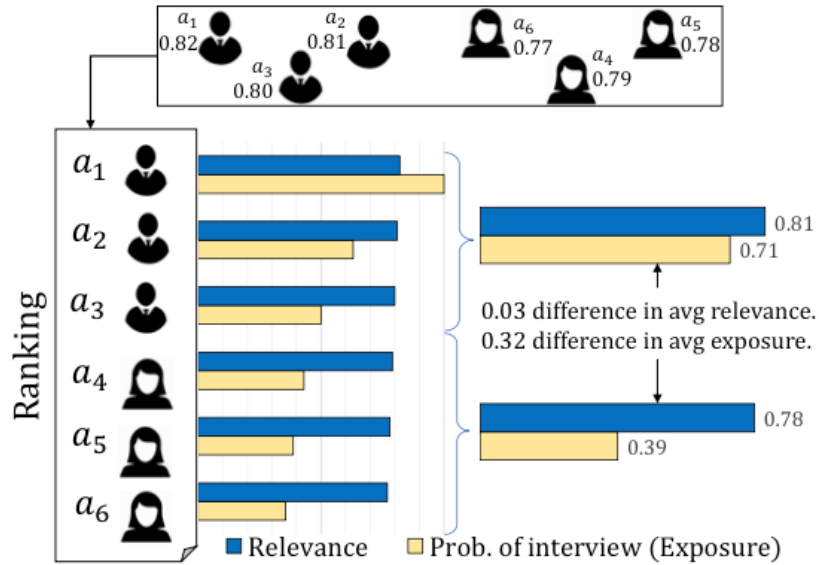
- fairness aspect: **exposure** should be **proportional** to **merit**
  - if an item is x times better than another, it should receive x times more exposure

Fairness of Exposure

- fairness = exposure ~ merit (relevance)
  ranking
- can we achieve this? almost never
  - exposure is **fixed**
  - merit **depends** on ranking scores

job applicants

ranking by merit



high discrepancy in group exposure

[2018 KDD A. Singh T. Joachims] *Fairness of Exposure in Rankings*

Fairness of Exposure

Amortized fairness acknowledges that achieving perfect fairness at every moment may be challenging. Instead, it aims for fairness over the long term, allowing for some fluctuations in exposure over short periods.

- what can we do then?
  - be fair in the long-term: **amortized fairness**
  - or be fair **probabilistically**: probabilistic rankings

Rather than strictly adhering to a fixed ranking, the system assigns probabilities to items' positions.
This allows for a degree of randomness in exposure while still favoring higher-ranked items.

- formulate an optimization problem

there can be trade off between relevance and exposure

  - **maximize** relevance **subject to** equal exposure, or
  - **minimize** exposure discrepancy **subject to** relevance quality drop

[2018 SIGIR A. Biega et al.] *Equity of Attention: Amortizing Individual Fairness in Rankings*
[2018 KDD A. Singh T. Joachims] *Fairness of Exposure in Rankings*

Fairness in Recommendations

What are recommender systems?

amazon

We Have Recommendations for You

Sign in to see personalized recommendations

Customers who bought this item also bought

Ultimate Ears Power Up Charging Dock for BOOM 3, MEGABOOM 3, BLAST and MEGABLAST
15
$39.99

LTGEM Case Compatible for Ultimate Ears UE Megaboom Wireless Bluetooth Speaker. Fits...
203
$12.99

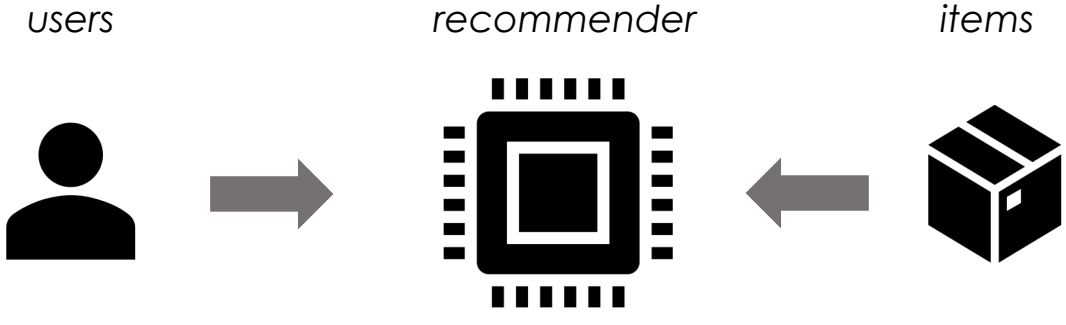
What other items do customers buy after viewing this item?

Ultimate Ears MEGABOOM Charcoal Wireless Mobile Bluetooth Speaker Waterproof and Shockproof (2015)
1,086
$102.99 prime

Ultimate Ears Power Up Charging Dock for BOOM 3, MEGABOOM 3, BLAST and MEGABLAST
15
$39.99 prime

Stakeholders in recommender systems

recommenders = match **users** with **items**



consumers **system provider** **producers**

a multi-sided market with possibly conflicting interests and various **fairness concerns**

Fairness in recommendations

fairness is the **fair distribution** of a **resource**

to make it less abstract and cyclic, answer three questions:

- fair **for whom?** (which are the *protected groups*)
  - **consumers** (end users, buyers)
  - **producers** (item providers, creators, sellers)

- distribution **of what** resource?
  - **accuracy** (effectiveness, utility, satisfaction, quality of service, etc.)
    - e.g., items are good matches for users; users are good matches to items
  - **exposure** (attention)
    - e.g., what users see; to whom items are recommended

- **when** is the distribution fair?
  - specify the **optimal state** of fairness
  - quantify how far the optimal state is = define some (**un-**)**fairness measure**

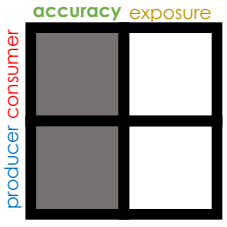
Taxonomy of fairness definitions in recommenders

of what?

for whom?	**accuracy**	**exposure**
consumer	prediction accuracy [2017 NeurIPS S. Yao et al.] ranking accuracy [2018 FAT M. Ekstrand et al.]	equal exposure [2018 FAT R. Burke et al.]
producer	ranking accuracy [2019 KDD A. Beutel et al.]	exposure coverage [2018 CIKM, 2019 RecSys] equal exposure [2018 FAT R. Burke et al.] calibrated exposure [2018 RecSys]

some representative works

Accuracy in recommender systems

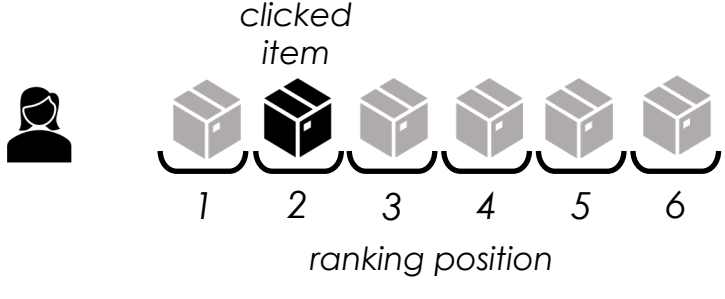


view 1: the recommender **matches** users with items

prediction accuracy: how good is the matching (regression task)

- metrics: absolute error (MAE), root mean square error (RMSE)

view 2: the recommender **ranks** items for a user

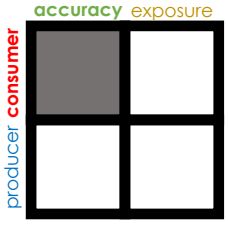


reciprocal rank (RR) = 1/2

ranking accuracy: how good is the ranking (ranking task)

- metrics: reciprocal rank (RR), discounted cumulative gain (DCG)

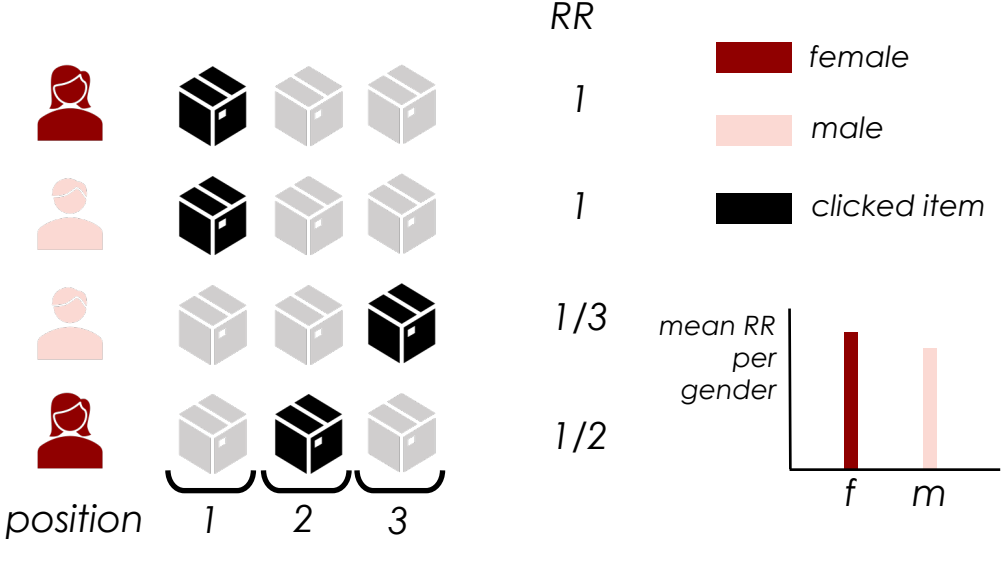
Fairness of **accuracy** for **consumers**



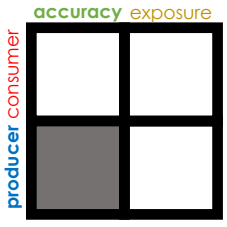
*"**Males** and **females** should experience the same quality of service."*

operationalized as:

1. **group** users **by gender**
2. measure **total accuracy** per gender
3. **assess** if distribution is **fair**



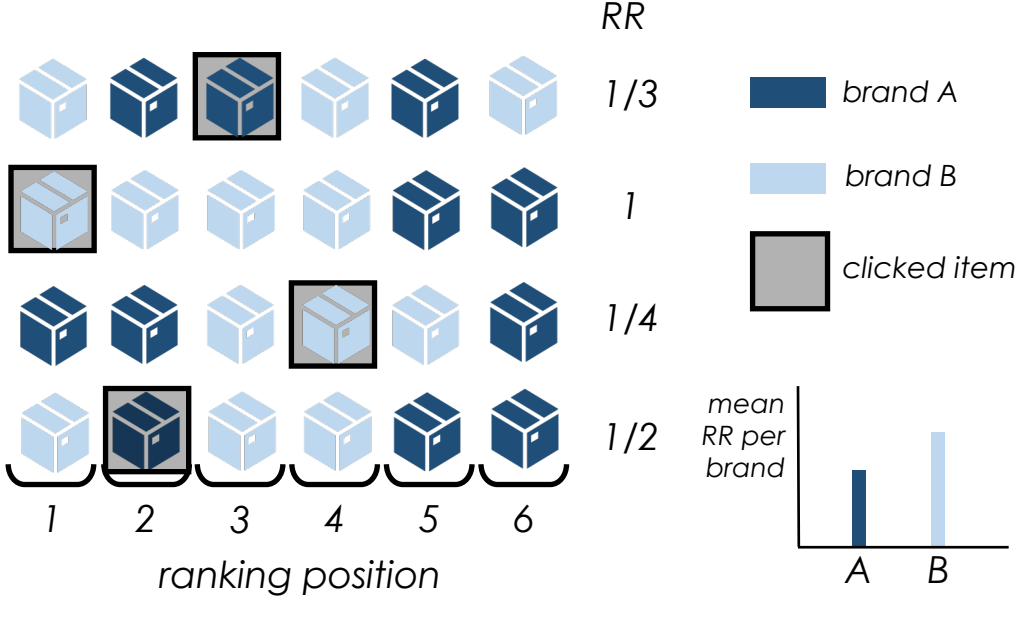
Fairness of **accuracy** for **producers**

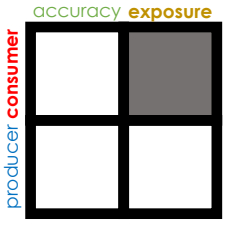


*"When recommending products, **all brands** should have similar accuracy."*

operationalized as:

1. **group** items **by brand**
2. measure **total accuracy** per brand
3. **assess** if distribution is **fair**



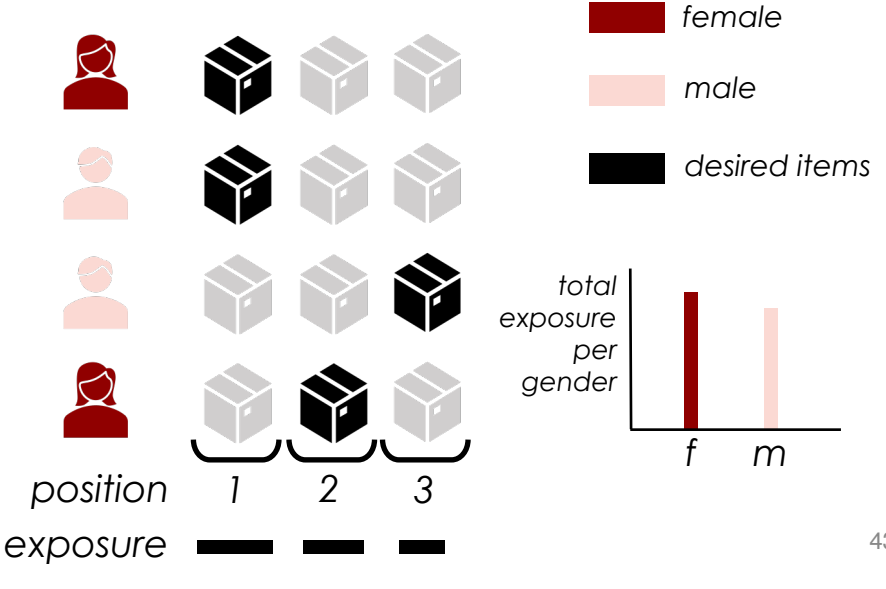


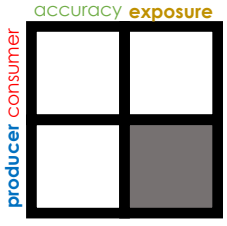
Fairness of **exposure** for **consumers**

*"When recommending jobs, **males** and **females** should see the same number of executive openings."*

operationalized as:

1. **group** users **by gender**
2. measure **total exposure** per gender
3. **assess** if distribution is **fair**



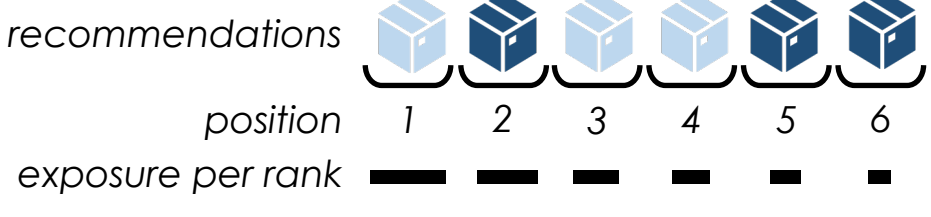


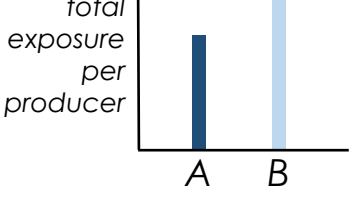
Fairness of **exposure** for **producers**

*"When recommending products, **all brands** should be fairly exposed."*

operationalized as:

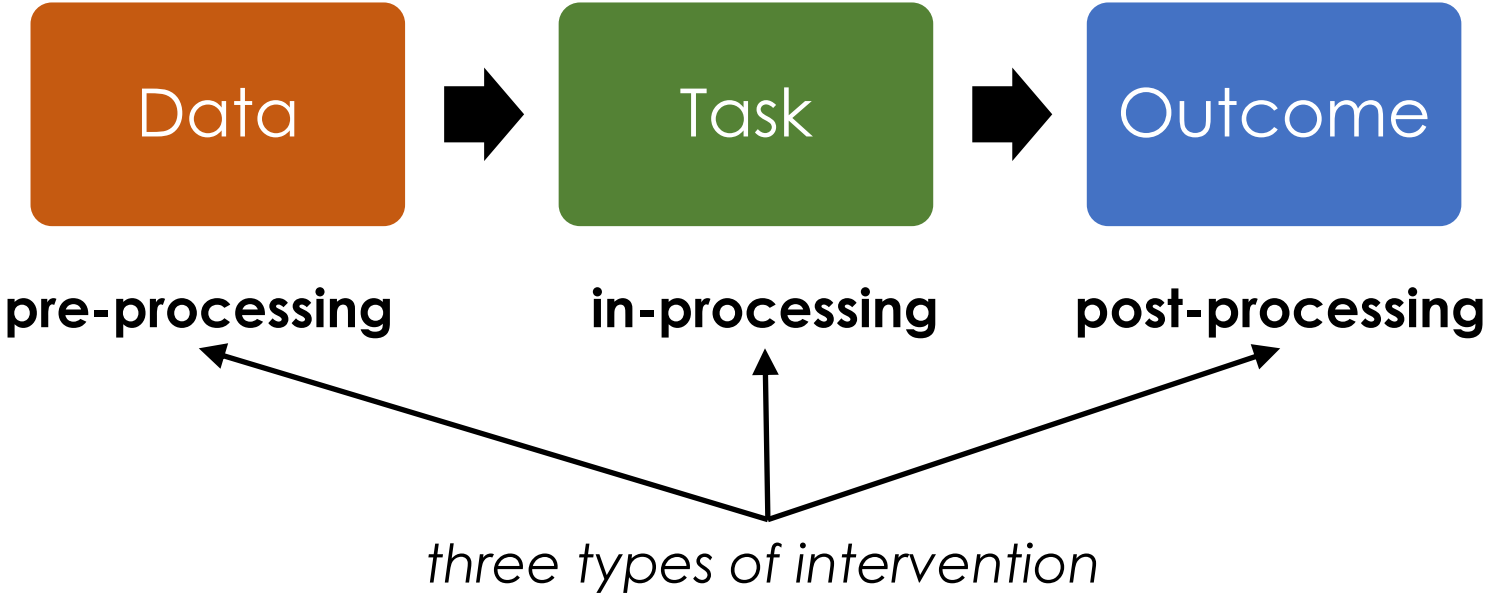
1. **group** items **by producer**
2. measure **total exposure** per producer
3. **assess** if the distribution is **fair**





How to Achieve Fairness

How to Achieve Fairness



Post-Processing for Group Parity in Ranking

Step 1. Determine **minimum representation** from the **protected** group in each **prefix**

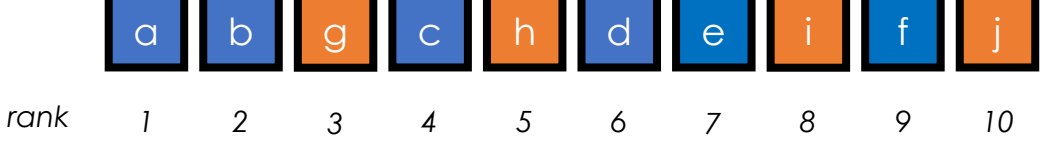
- Suppose we want at least $\lfloor 40\% N \rfloor$ orange at every top-N

N	1	2	3	4	5	6	7	8	9	10
# orange	0	0	1	1	2	2	2	3	3	4

Step 2. Create two sublists of objects (blue and orange), sorted on score

Step 3. Build the ranked list incrementally

- at each rank choose the **best** from either group (why best?),
- **unless** you *must* choose from the protected (to ensure minimum representation)

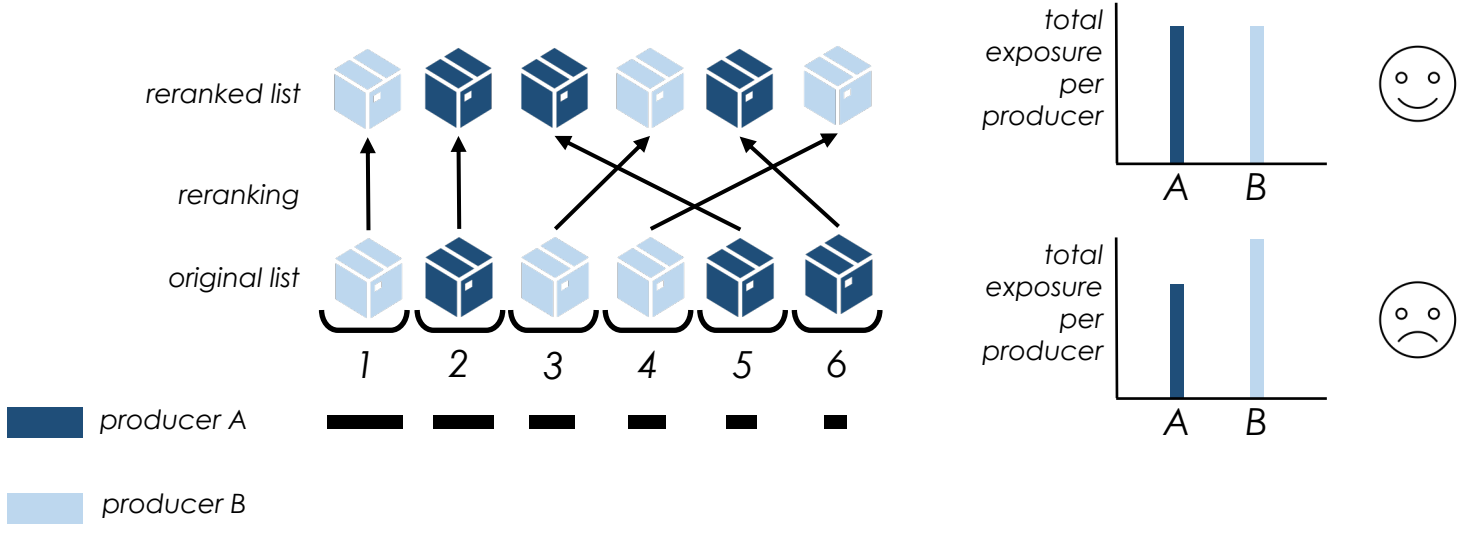


Post-Processing for Fair Exposure in Ranking

change the position of items to improve fairness

trade-off between **fairness** and **accuracy**

***reranking goal**: equalize exposure*



Post-Processing for Ranking

the **original list** optimizes some **internal measure** of **utility**

- e.g., relevance, click-through rate

but to increase **fairness**, the list has to be reranked

- at the expense of **utility**

"utility" typically refers to a measure of how valuable or relevant an item is to a user

reranking must **trade-off** two objectives: **utility** and **fairness**

eti jana fair bhanera fix garne like 2 jana female hunai parcha tespachi utility herne

possibilities:

maxU	maximize **utility** given a constraint on **fairness**
maxF	maximize **fairness** given a constraint on **utility**
U+F	maximize a combination of **utility** and **fairness**

Other Fairness Challenges

Continuous Protected Attributes

- All definitions compare a **protected group** (e.g., blacks, women) against the **non-protected**

- What happens in the case of continuous protected attributes, e.g., **age**, **income**, ***location***

- Groups are not defined beforehand, and defining them can lead to **gerrymandering**
  - purposefully setting the group boundaries to **hide discrimination**

Continuous Protected Attributes

- All definitions compare a **protected group** (e.g., blacks, women) against the **non-protected**

- What happens in the case of continuous protected attributes, e.g., **age**, **income**, ***location***

- Groups are not defined beforehand, and defining them can lead to **gerrymandering**
  - purposefully setting the group boundaries to **hide discrimination**

Spatially Fair Distribution of Outcomes

X	O	X	O
O	X	O	X
X	O	X	O
O	X	O	X

Spatially Unfair Distribution of Outcomes

X	X	O	O
X	O	X	O
X	X	O	O
X	X	O	O

Continuous Protected Attributes

- All definitions compare a **protected group** (e.g., blacks, women) against the **non-protected**

- What happens in the case of continuous protected attributes, e.g., **age**, **income**, ***location***

- Groups are not defined beforehand, and defining them can lead to **gerrymandering**
  - purposefully setting the group boundaries to **hide discrimination**

Spatially Fair Distribution of Outcomes

X	O	X	O
O	X	O	X
X	O	X	O
O	X	O	X

X	O	X	O
O	X	O	X
X	O	X	O
O	X	O	X

X	O	X	O
O	X	O	X
X	O	X	O
O	X	O	X

Spatially Unfair Distribution of Outcomes

X	X	O	O
X	O	X	O
X	X	O	O
X	X	O	O

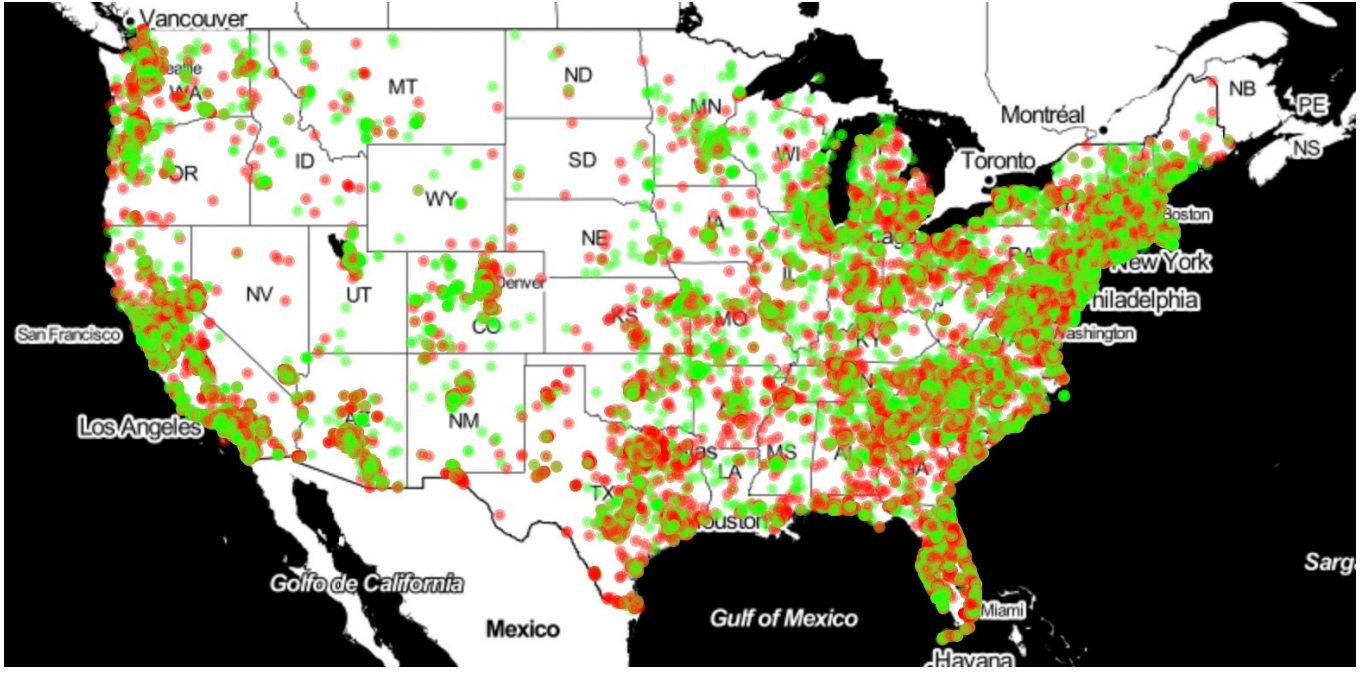
X	X	O	O
X	O	X	O
X	X	O	O
X	X	O	O

X	X	O	O
X	O	X	O
X	X	O	O
X	X	O	O

1 x 1 *2 x 2* *1 x 4*

Location Fairness

- We want **outcomes** to be **independent of location**
- Consider **Loan Application** **Accepts** and **Rejects** per location
- Is this map **fair**?



Discover Unfairness via Explanations

- Previous definitions require the groups (protected vs. non-protected) to be fixed and known in advance
- Can you **discover** groups (or sub-groups) where discrimination occurs?

Counterfactual Explanations

- A person x **does not receive** a loan
  - x = {Race=Caucasian, Gender=Female, Income=Low}
- What are the **minimal changes** to **receive** the loan?
  - {Gender=Female} → {Gender=Male}

- This shows **discrimination** based on gender!

INFO-H420
Management of Data Science and Business Workflows

Part II
Explainability

Dimitris SACHARIDIS

2023-2024

Explanations

- recommendations are everywhere – explanations are there as well!

amazon

Related to items you've viewed

Frequently Bought Together

Customers Who Bought This Item Also Bought

Recommended for You Based on

NETFLIX

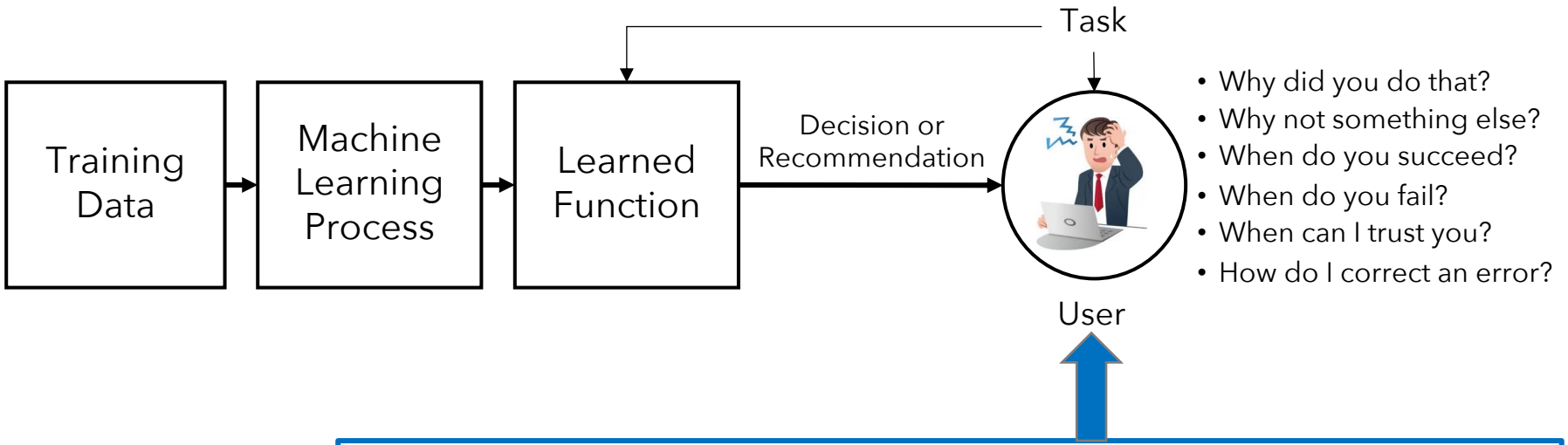
Popular on Netflix

Trending Now

Watch It Again

Because you watched Dark

Explainable AI



- The target of XAI is an end user who:
  - depends on decisions, recommendations, or actions of the system
  - needs to understand the rationale for the system's decisions to understand, appropriately trust, and effectively manage the system

[DARPA] Explainable Artificial Intelligence (XAI)

Explainable AI

Human-understandable explanations of **outcomes** of algorithmic decision-making systems

Explanations that

- enable understanding of overall strengths & weaknesses
- convey an understanding of how the system will behave in the future
- convey how to correct the system's mistakes

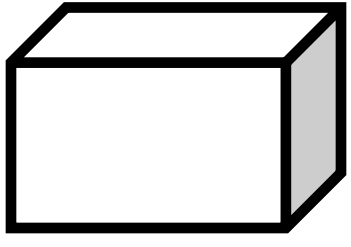
XAI is used to:

- detect biases and prevent discrimination
- detect training data errors, debug the model
- comply with regulations for transparency

[DARPA] Explainable Artificial Intelligence (XAI)

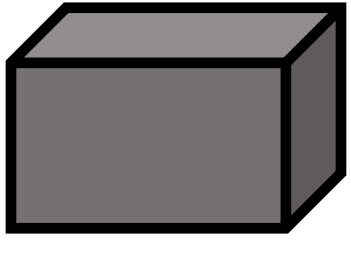
White-Box vs Black-Box

- a **white box** is a transparent AI system
  - complete knowledge of inner workings
  - often interpretable by humans = interpretation



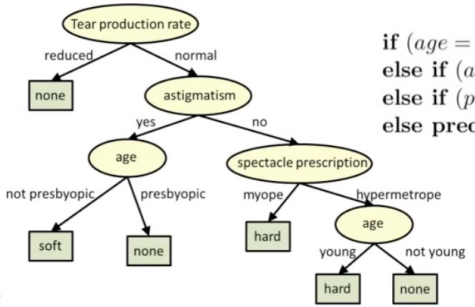
white box

- a **black box** is an opaque, complex AI system
  - no knowledge of inner workings
  - we can only try to make an educated guess about how it reaches a decision = explanation



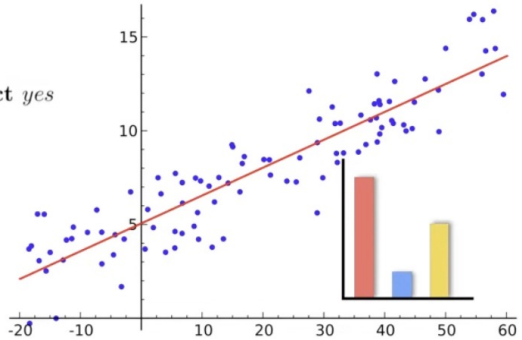
black box

How to achieve Explainability

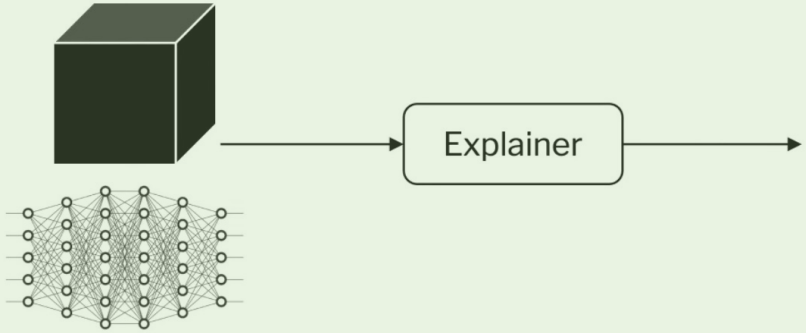


if $(age = 18 - 20)$ **and** $(sex = male)$ **then predict** yes
else if $(age = 21 - 23)$ **and** $(priors = 2 - 3)$ **then predict** yes
else if $(priors > 3)$ **then predict** yes
else predict no

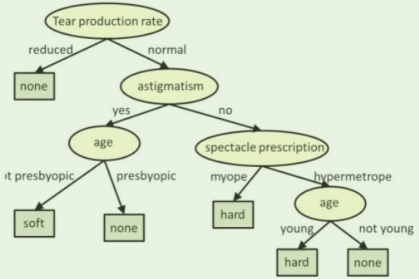
Intrinsic



Post hoc



if $(age = 18 - 20)$ **and** $(sex = male)$ **then predict** yes
else if $(age = 21 - 23)$ **and** $(priors = 2 - 3)$ **then predict** yes
else if $(priors > 3)$ **then predict** yes
else predict no



[2022 ACM SIGMOD Pradhan et al.] Explainable AI: Foundations, Applications, and Opportunities for Data Management Research https://explainable-ai-tutorial.github.io

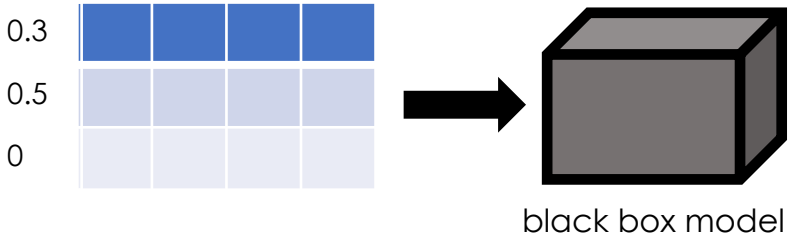
Types of Explanations

Name	Age	Saving	Month	...	Credit History
Maeve	<25	<100 DM	24	...	Paid duly

Why was my loan application rejected?

Feature-based Explanations

Data-based Explanations



[2022 ACM SIGMOD Pradhan et al.] Explainable AI: Foundations, Applications, and Opportunities for Data Management Research https://explainable-ai-tutorial.github.io

Why Explain?

Purpose	Description
Transparency	Explain how the system works
Effectiveness	Help users make good decisions
Trust	Increase users' confidence in the system
Persuasiveness	Convince users to try or buy
Satisfaction	Increase the ease of use or enjoyment
Education	Allow users to learn something from the system
Scrutability	Allow users to tell the system it is wrong
Efficiency	Help users make decisions faster
Debugging	Allows users to identify that there are defects in the system

[1984 B. G. Buchanan et al.] *Explanations as a Topic of AI Research, in Rule-based Systems*

[2017 UMUAI I. Nunes et al.] *A systematic review and taxonomy of explanations in decision support and recommender systems.*

[2007 ICDE_w N. Tintarev et al.] *A survey of explanations in recommender systems.*

What is a good Explanation?

	Co-12 Property	Description
Content	**Correctness**	Describes how faithful the explanation is w.r.t. the black box.
		Key idea: Nothing but the truth
	Completeness	Describes how much of the black box behavior is described in the explanation.
		Key idea: The whole truth
	Consistency	Describes how deterministic and implementation-invariant the explanation method is.
		Key idea: Identical inputs should have identical explanations
	Continuity	Describes how continuous and generalizable the explanation function is.
		Key idea: Similar inputs should have similar explanations
	Contrastivity	Describes how discriminative the explanation is w.r.t. other events or targets.
		Key idea: Answers "why not?" or "what if?" questions
	Covariate complexity	Describes how complex the (interactions of) features in the explanation are.
		Key idea: Human-understandable concepts in the explanation
Presentation	**Compactness**	Describes the size of the explanation.
		Key idea: Less is more
	Composition	Describes the presentation format and organization of the explanation.
		Key idea: *How* something is explained
	Confidence	Describes the presence and accuracy of probability information in the explanation.
		Key idea: Confidence measure of the explanation or model output
User	**Context**	Describes how relevant the explanation is to the user and their needs.
		Key idea: How much does the explanation matter in practice?
	Coherence	Describes how accordant the explanation is with prior knowledge and beliefs.
		Key idea: Plausibility or reasonableness to users
	Controllability	Describes how interactive or controllable an explanation is for a user.
		Key idea: Can the user influence the explanation?

[2023 ACM Comp. Surveys M. Nata et al.] *From Anecdotal Evidence to Quantitative Evaluation Methods: A Systematic Review on Evaluating Explainable AI*

A use case: the Adult Dataset

Adult

Donated on 4/30/1996

Predict whether income exceeds $50K/yr based on census data. Also known as "Census Income" dataset.

Dataset Characteristics
Multivariate

Subject Area
Social Science

Associated Tasks
Classification

Feature Type
Categorical, Integer

 # Instances
48842

 # Features
14

Name	Age	Education	Marital	…	Race	Gender	Hours	Income
Nazia	36	Bachelors	Unmarried	…	Asian	female	40	<50K
Matt	26	Bachelors	Married	…	White	male	40	≥50K
Yeji	50	Masters	Married	…	Asian	male	16	≥50K
Neel	45	Masters	Unmarried	…	Black	male	28	<50K
…	…	…	…	…	…	…	…	…

https://archive.ics.uci.edu/dataset/2/adult

A use case: the Adult Dataset

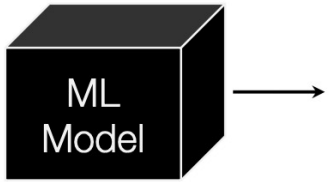
Bank

Rosa

Other customers of the bank

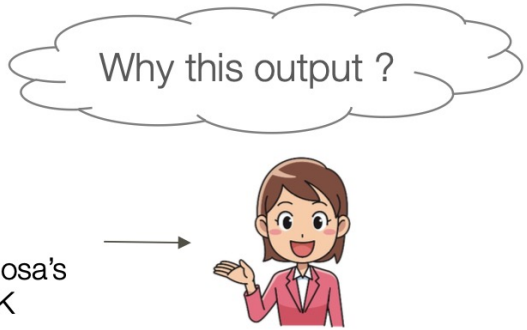
Name	Age	Education	Marital	…	Race	Sex	Hours
Rosa	34	Bachelors	Unmarried	…	White	female	40

Rosa's data



83%

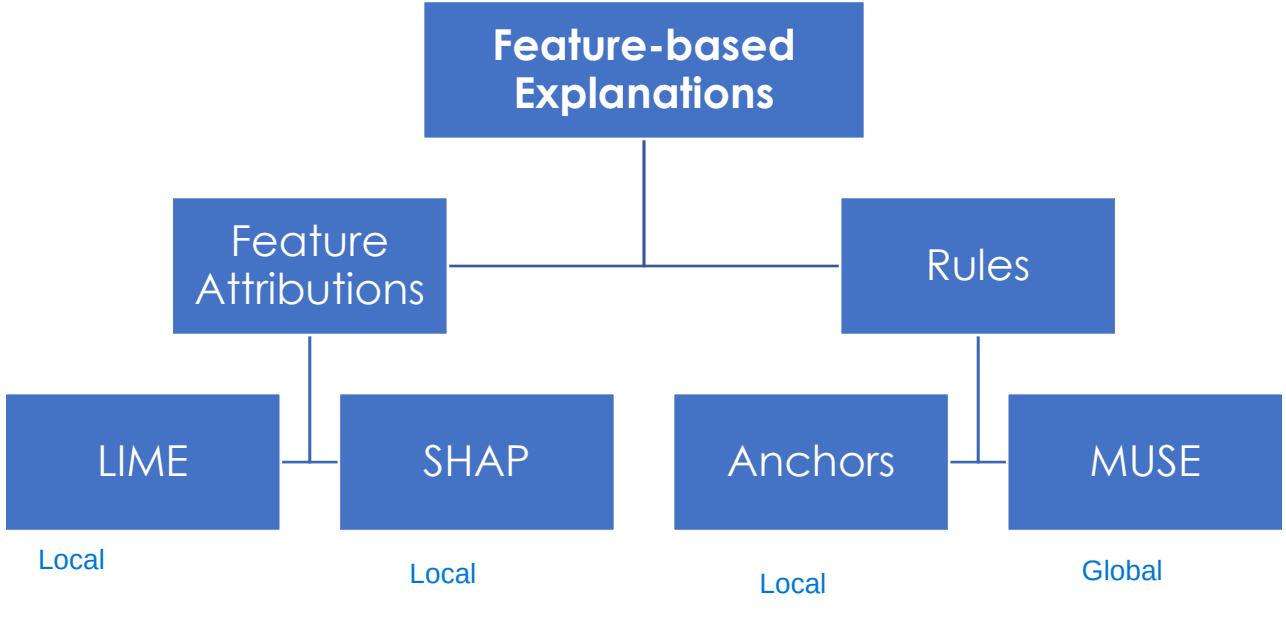
Probability that Rosa's income < 50K

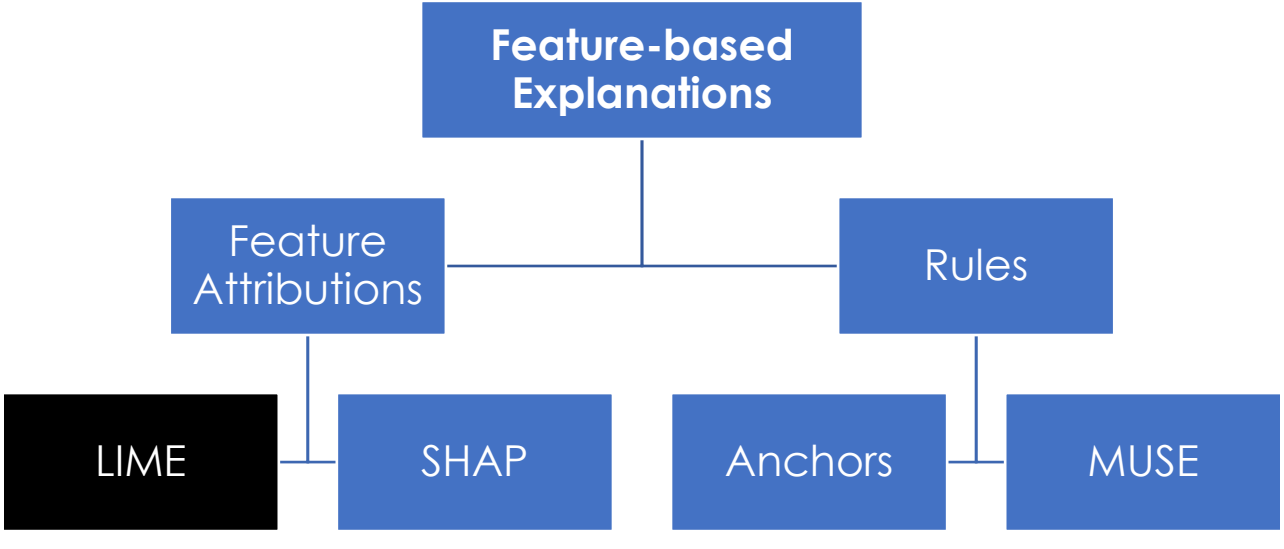


Manager

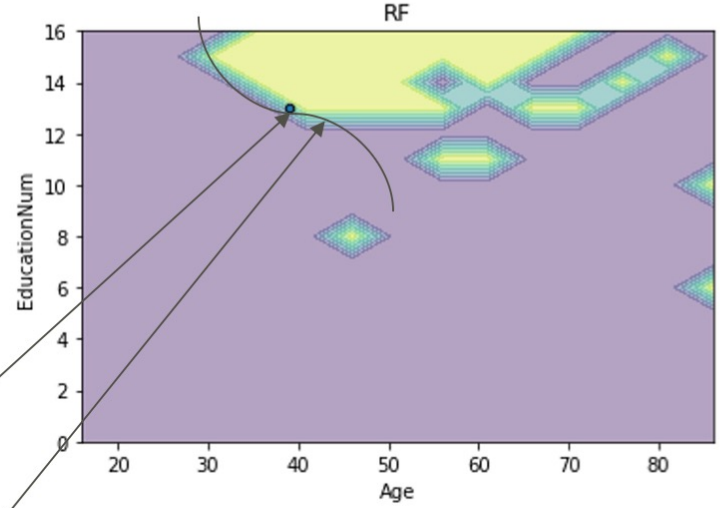
[2022 ACM SIGMOD Pradhan et al.] Explainable AI: Foundations, Applications, and Opportunities for Data Management Research https://explainable-ai-tutorial.github.io

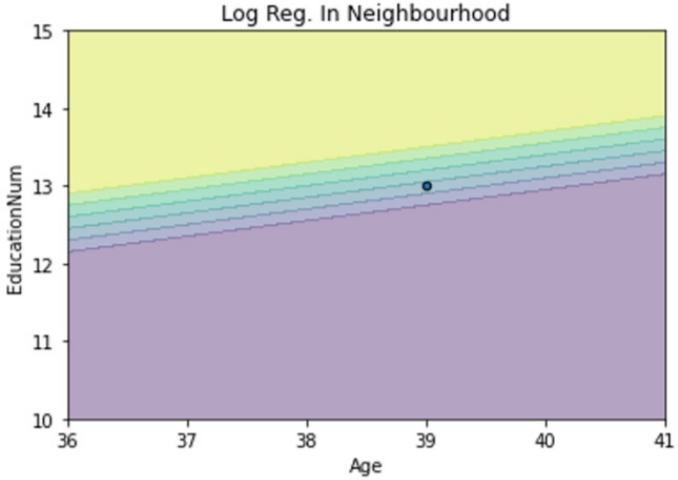
Feature-based Explanations


Feature-based Explanations
Feature Attributions
Rules
LIME
SHAP
Anchors
MUSE
Local
Local
Local
Global


Feature-based Explanations
Feature Attributions
Rules
LIME
SHAP
Anchors
MUSE

Complex Decision Boundaries





Want explanation for this point

2D Decision Boundary of a Complex Classifier

… are simpler locally

Take neighbouring points and fit linear model

[2022 ACM SIGMOD Pradhan et al.] Explainable AI: Foundations, Applications, and Opportunities for Data Management Research https://explainable-ai-tutorial.github.io

LIME – Algorithm

LIME perturbs instances around a specific data point, observes the changes in predictions, and fits a simple, interpretable model (e.g., linear model) to approximate the behavior of the black-box model locally.

Local Interpretable Model-Agnostic Explanations (LIME)

- fit an interpretable model locally
- get feature weights

Neighborhood generation

Perturb random sample

~ N(0,1)

normally distributed data with mean 0 std 1

→

Weighting function

Weigh instances by proximity to target

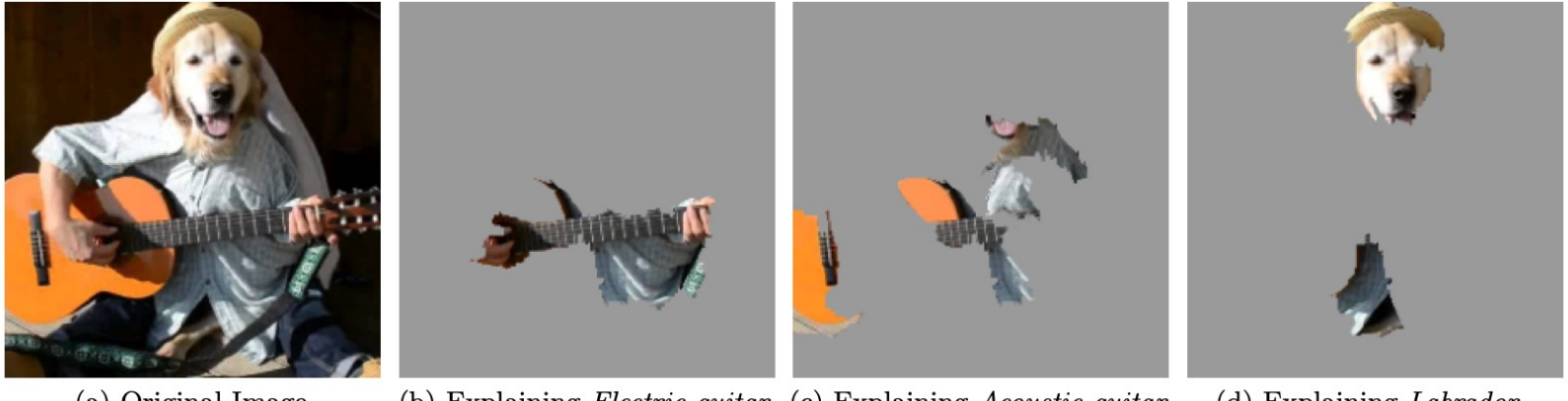
Exponential smoothing

→

Surrogate Training

Train a weighted model.

Sparsity using Lasso.



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

[2016 ACM SIGKDD Ribeiro et al.] "Why should I trust you?" Explaining the predictions of any classifier

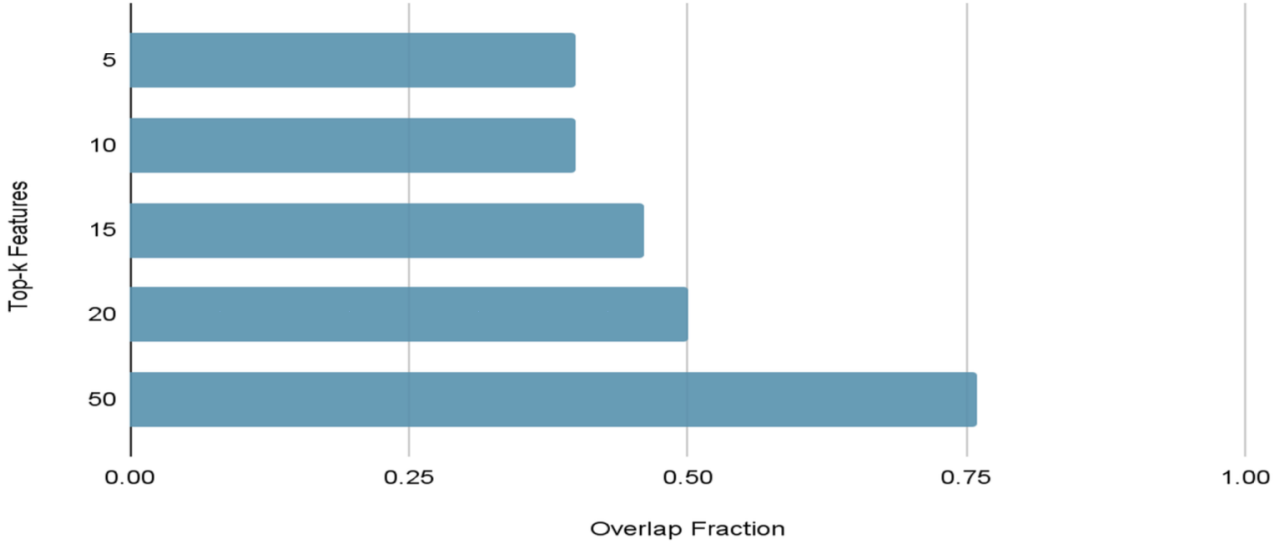
LIME – Issues

Rosa

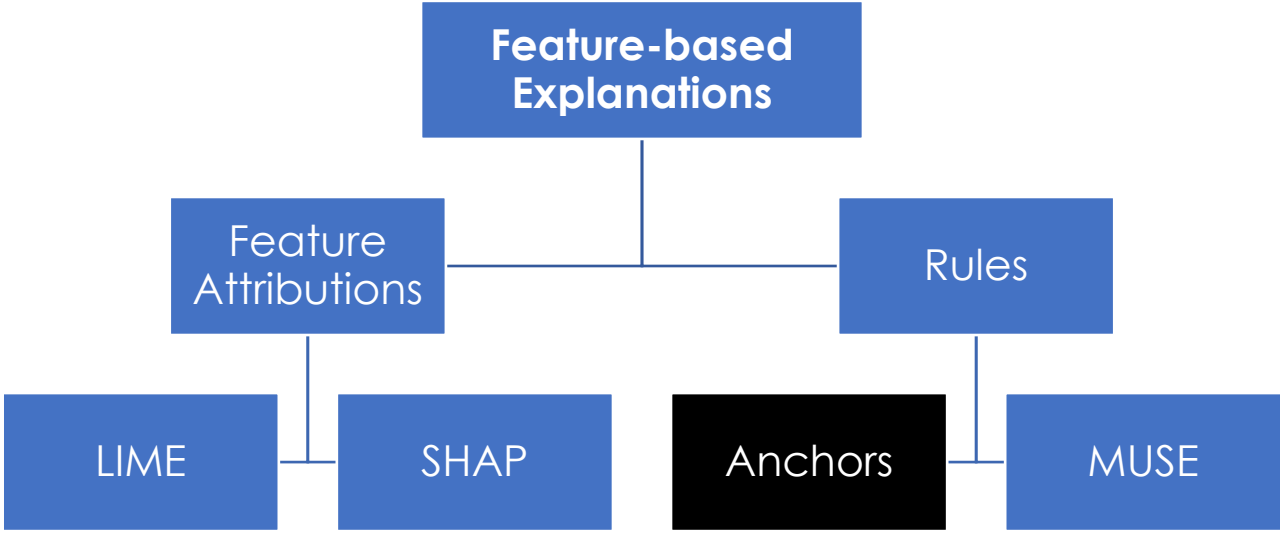
Name	Age	Race	…	Hours
Rosa	34	White	…	**40**

Emma

Name	Age	Race	…	Hours
Emma	34	White	…	**30**



[2022 ACM SIGMOD Pradhan et al.] Explainable AI: Foundations, Applications, and Opportunities for Data Management Research https://explainable-ai-tutorial.github.io


Feature-based Explanations
Feature Attributions
Rules
LIME
SHAP
Anchors
MUSE

Rule-based Explanations

Rosa

Name	Age	Education	Marital	…	Race	Sex	Hours
Rosa	34	Bachelors	Unmarried	…	White	female	40

Rosa's data

ML Model → < 50K Rosa's income

IF Hours<50 **AND** Marital != "Married", **THEN** <50K

- Local sufficient conditions explaining the output.
- Aims for clear coverage of explanation with precise rules

[2018 AAAI Ribeiro et al.] Anchors: High-precision model-agnostic explanations

LIME vs Anchors

Rosa

Name	Age	Race	...	Hours
Rosa	34	White	...	**40**

Emma

Name	Age	Race	...	Hours
Emma	34	White	...	**30**

Manager

Top 3 features
Age
Education
Hours

Top 3 features

- The scope or validity of LIME is unclear.
- ANCHORS provides clear conditions where explanations hold.

IF Hours<50 **AND** Marital != “Married”, **THEN** <50K

Anchors – Quality of Rules

Rosa

Name	Age	Education	Marital	…	Race	Sex	Hours
Rosa	34	Bachelors	Unmarried	…	White	female	40

Rosa's data

PRECISION

IF Hours<50, **THEN** <50K

IF Hours<50 **AND** Marital != "Married", **THEN** <50K

IF Hours<50 **AND** Marital != "Married" **AND** Race = "White", **THEN** <50K

COVERAGE

Precision = # of instances where provided explanation is correct.
Coverage = # of input instances where explanation clause is valid.

Anchors – Algorithm

Multi Armed Bandit Problem

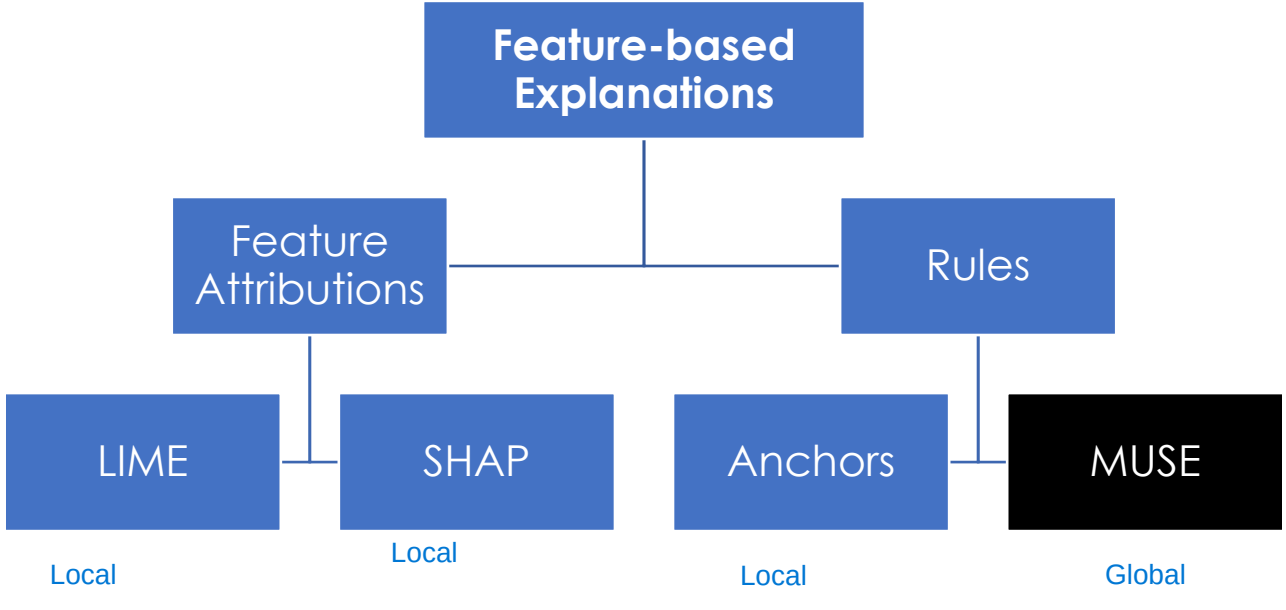
Beam Search

Idea: Create Rules, and Pick the most promising to expand

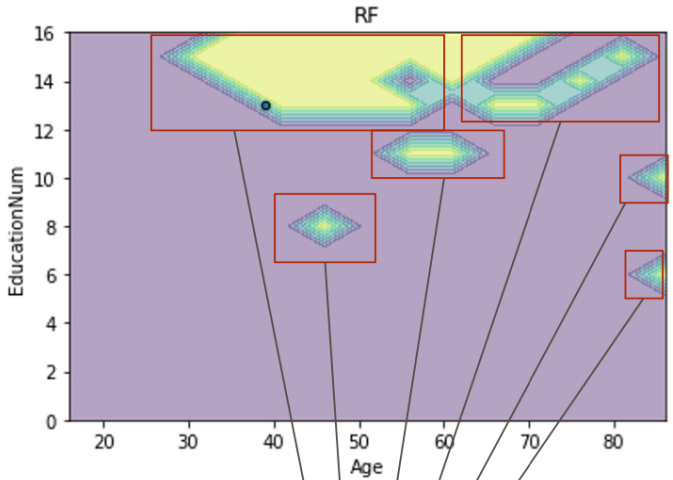
IF Hours<50

IF Hours<50 **AND** Marital != "Married"

IF Hours<50 **AND** Marital != "Married" **AND** Race = "White"


Feature-based Explanations
Feature Attributions
Rules
LIME
SHAP
Anchors
MUSE
Local
Local
Local
Global

Global Explainability



2D Decision Boundary of a Complex Classifier

Global Subspace Level Explanations

[2019 AAAI/ACM AES Lakkaraju et al.] Faithful and customizable explanations of black box models

MUSE – Global Explanations

Name	Age	Education	Marital	…	Race	Gender	Hours	Income
Nazia	36	Bachelors	Unmarried	…	Asian	female	40	<50K
Matt	26	Bachelors	Married	…	White	male	40	≥50K
Yeji	50	Masters	Married	…	Asian	male	16	≥50K
Neel	45	Masters	Unmarried	…	Black	male	28	<50K
…	…	…	…	…	…	…	…	…

Entire dataset

If Age < “34”

If Education==“Bachelors” AND Marital!=“Married” , THEN <50K

If Education==“Bachelors“ AND Age > “40” , THEN >50K

If Hours > 40

If Education==”Masters” AND Marital!=“Married” , THEN <50K

Default

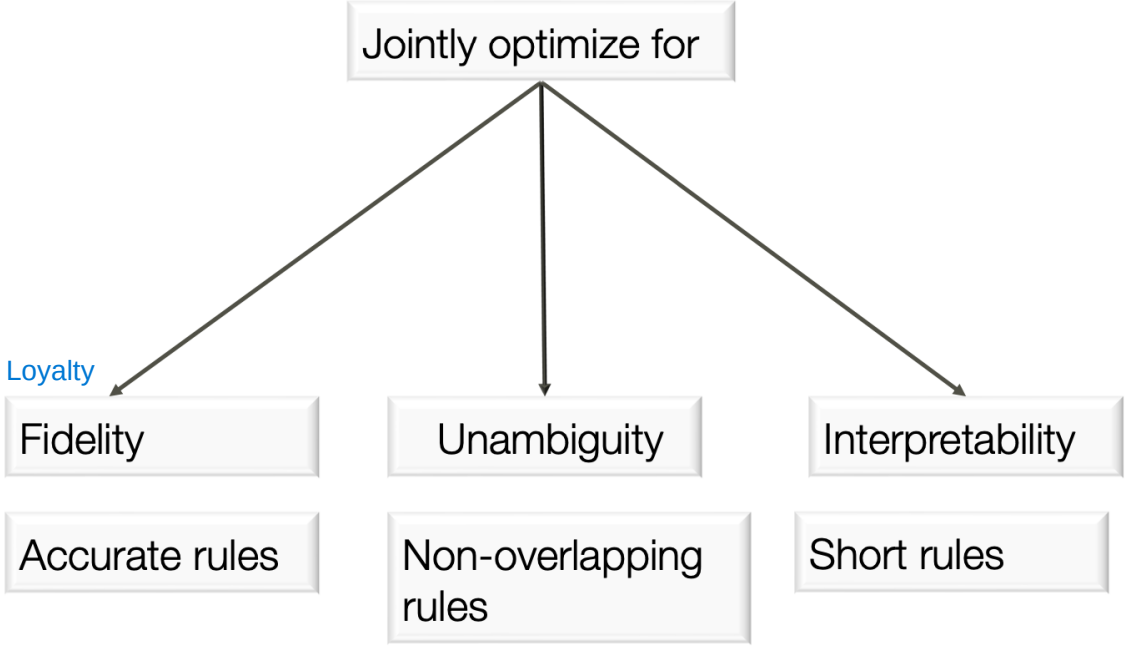
If Education==“Bachelors” AND Marital==“Married” , THEN >50K

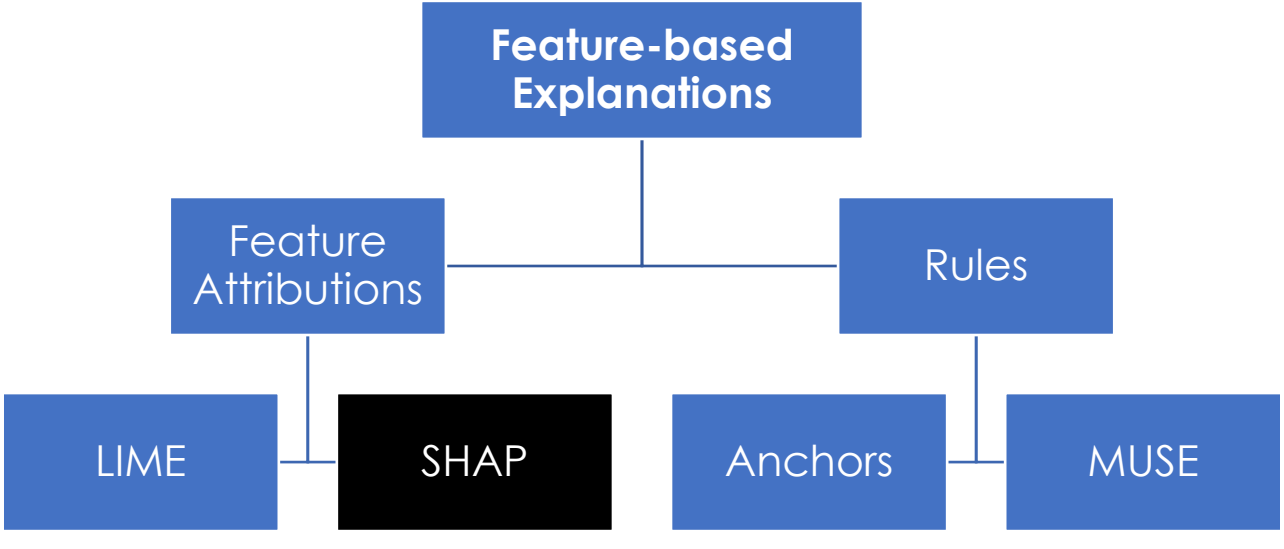
If Race==“Black” AND Hours<“20” , THEN <50K

Global, two-level rule-based explanations.

MUSE – Algorithm

Find a **good set** of rules that explain the model globally



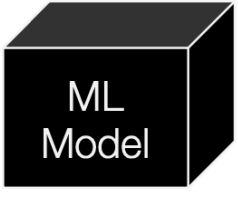

Feature-based Explanations
Feature Attributions
Rules
LIME
SHAP
Anchors
MUSE

Motivation

Rosa

Name	Age	Education	Marital	…	Race	Sex	Hours
Rosa	34	Bachelors	Unmarried	…	White	female	40

Rosa's data



< 50K

Rosa's income

p=0.83

prediction probability

find feature importance weights such that:

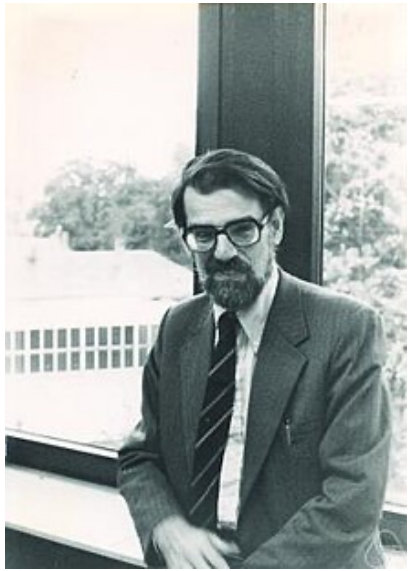
$\phi_{education} + \phi_{age} + \phi_{race} \ldots\ldots = 0.83$

Properties:

- Feature weights add up to prediction (Efficiency)
- Guarantees for when $\phi_{feature} = 0$ (Null)
- If feature1 ~ feature2 then $\phi_{feature1} = \phi_{feature2}$ (Symmetry)

Shapley Values – Game Theory

Shapley Values is the **unique** solution to the **fair allocation** of payoffs among members in a **coalition** participating in a game characterized by a specific set of axioms.



Lloyd Shapley
2012 Nobel Memorial Prize in Economic Sciences

[2002 Winter, Eyal.] "The shapley value." Handbook of game theory with economic applications

Shapley Values – Terminology

Coalition : Set of players

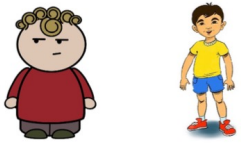
Null Coalition : Empty Set of players

$\varnothing$

Grand Coalition : Set of All players

S : players in coalition

S' : players out of coalition



V(S) : Value function for S

Score ( ) = 10

Shapley Values – Example

2 players playing table tennis (alone + team)

Q. Who is the better player?

Use Shapley Value to distribute their joint score
to them individually, fairly.

p1

p2

Game

Shapley Values – Example

p1 = 6

p2 = 8

p1 p2 = 12

$\emptyset$ = 0

Shapley Values – Formula

N=2
S=1

V- Value function
Su{i} - contribution in team with player
v(S)- without that player

$$\phi_i(v) = \underbrace{\sum_{S \subseteq N \setminus \{i\}}}_{\text{Iterating over all possible coalitions without feature i}} \underbrace{\frac{|S|!(|N| - |S| - 1)!}{|N|!}}_{\text{Weight of coalition}} \underbrace{(v(S \cup \{i\}) - v(S))}_{\text{Marginal contribution of feature i to this coalition S}}$$

Shapley Value for feature i

Shapley Values – Example

( ) p1 — (∅) => 6 - 0 = 6

(p1 p2) — (p2) => 12 - 8 = 4

Φ p1 => ½*(6) + ½*(4) = 5

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} (v(S \cup \{i\}) - v(S))$$

Shapley Values – Example

( ) — (∅) => 8 - 0 = 8

p2

( ) — ( ) => 12 - 6 = 6

p2 p1 p1

Φ => ½*(8) + ½*(6) = 7

p2

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} (v(S \cup \{i\}) - v(S))$$

Shapley Values – Example

$\Phi_{\text{ }}$ $\Phi_{\text{ }}$ ( )

$5 + 7 = 12$

Shapley Values – Feature Weights

Game

Players

Name	Age	Education	Marital	…	Hours
Rosa	34	Bachelors	Unmarried	…	40

Φ $+$ Φ

Payoff

$\Phi_{\text{Age}} + \Phi_{\text{Edu}} + \cdots \Phi_{\text{Marital}}$

Shapley Values – Feature Weights

There is problem in coalition
We need all features, we cannot take out some features

= 6

Name	Age	Education	Marital	...
?	34	Bachelors	?	...

=

= 8

Name	Age	Education	Marital	...
?	?	Bachelors	?	...

=

Finding prediction for partial instances.
This is now a missing value problem.

Shapley Values – Feature Weights – Marginalise

Marginalise on all members in S' to find the value function.

$$v(\mathbf{X}_S=\mathbf{x}_S) = E_D[f(\mathbf{x}_S,\mathbf{X}_{S'})]$$

Name	Education	Marital
Rosa	Bachelors	Unmarried
James	Bachelors	Unmarried
Imran	Masters	Married
Ilya	PhD	Married

Rosa's explanation.

S=(Marital),S'=(Education)

$v(\mathbf{X}_S$=(Marital=Unmarried))

Trying to find contribution of marital
add different values of education
send all of them to model
model will give some value
average out
take out effect of education

Marginalise over Education

Name	Education	Marital
Rosa	Bachelors	Unmarried
P1	Masters	Unmarried
P2	PhD	Unmarried

Issue : Out of distribution instances

These two are out of distribution because model might not have known about this combination
so result may not be correct

Used in Quantitative Input Influence

[2016 (SP). IEEE Datta, et al.] Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems

Shapley Values – Feature Weights – Condition

Condition on all members in S to find the value function.

$$v(\mathbf{X}_S=\mathbf{x}_S) = E[\, f(\, \mathbf{X} \mid \mathbf{X}_S= \mathbf{x}_S \,) \,]$$

Name	Education	Marital
Rosa	Bachelors	Unmarried
James	Bachelors	Unmarried
Imran	Masters	Married
Ilya	PhD	Married

Rosa's explanation.

S=(Marital),S'=(Education)

$v(\mathbf{X}_S=_{\text{(Marital=Unmarried)}})$

Name	Education	Marital
Rosa	Bachelors	Unmarried
James	Bachelors	Unmarried

Condition on Marital

Contribution of Marital without Education

Used in SHAP

[2017 NeurIPS Lundberg, et al.] A unified approach to interpreting model predictions

Shapley Values – Feature Weights – Issues

Some attributes are highly corelated
results are not as expected
here education and higher degree are related to one another
if we remove higher degree, we get completely different value which is not expected

Choosing the players is non-trivial

Two variables Education(A) and Marital(B).

Add third redundant variable Higher Degree(C), such that,

$P(X_c=X_B)=1$

Name	Education (A)	Marital (B)	Higher Degree (C)
Rosa	Bachelors	Unmarried	No
James	Bachelors	Unmarried	No
Imran	Masters	Married	Yes
Tian	Masters	Unmarried	Yes

G_1 : Players=(A,B,C) : $f(X_a,X_b,X_c)$

$\phi_{G1}(B) = \phi_{G1}(C)$

G_2 : Players=(A,B): $f(X_a,X_b,X_b)$

$\phi_{G2}(B) \mathrel{!=} \phi_{G1}(B)$ and

$\phi_{G2}(B) \mathrel{!=} \phi_{G1}(B) + \phi_{G1}(C)$

[2020 ICML Kumar, et al.] Problems with Shapley-value-based explanations as feature importance measures

Shapley Values – Feature Weights – Issues

Actionability

Manager

Φ Age Φ Edu $\cdots$ Φ Marital

Contrastiveness

Rosa

Emma

[2020 ICML Kumar, et al.] Problems with Shapley-value-based explanations as feature importance measures

Shapley Values – Feature Weights – Issues

Computing exact shapley values is intractable. unmanageable

SHAP values are approximations of shapley values.

Computing SHAP	Linear Regression, Decision Trees	Logistic Regression, NN
With Feature Independence assumption	Tractable	#P-hard
Without Feature Independence assumption	#P-hard	#P-hard

[2021 AAAI Van den Broeck et al.] On the tractability of SHAP explanations

Data-Based Explanations

Data-Based Explanations

Use **training data points** to explain ML model behavior

Rosa

Name	Age	Education	Marital	…	Race	Gender	Hours
Rosa	34	Bachelors	Unmarried	…	Black	female	40

Rosa's data

ML Model

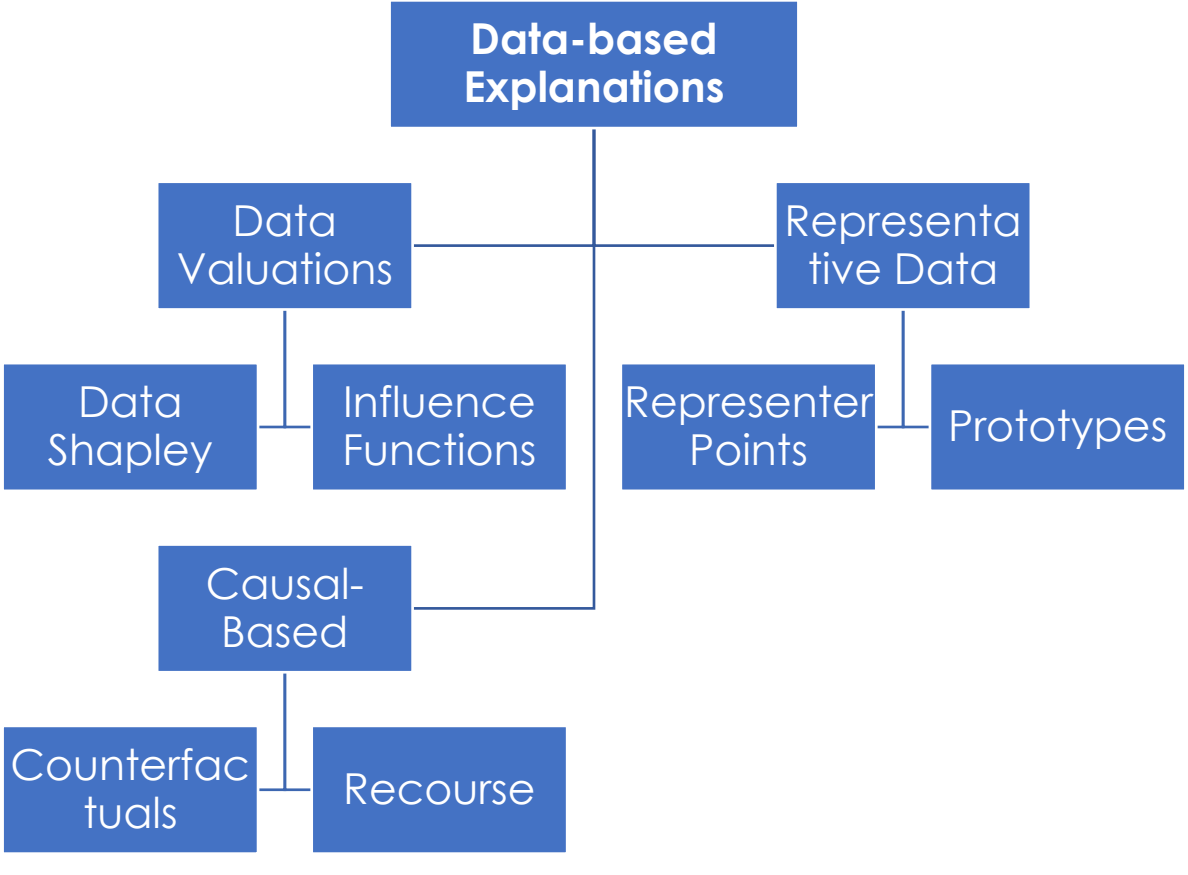
83%

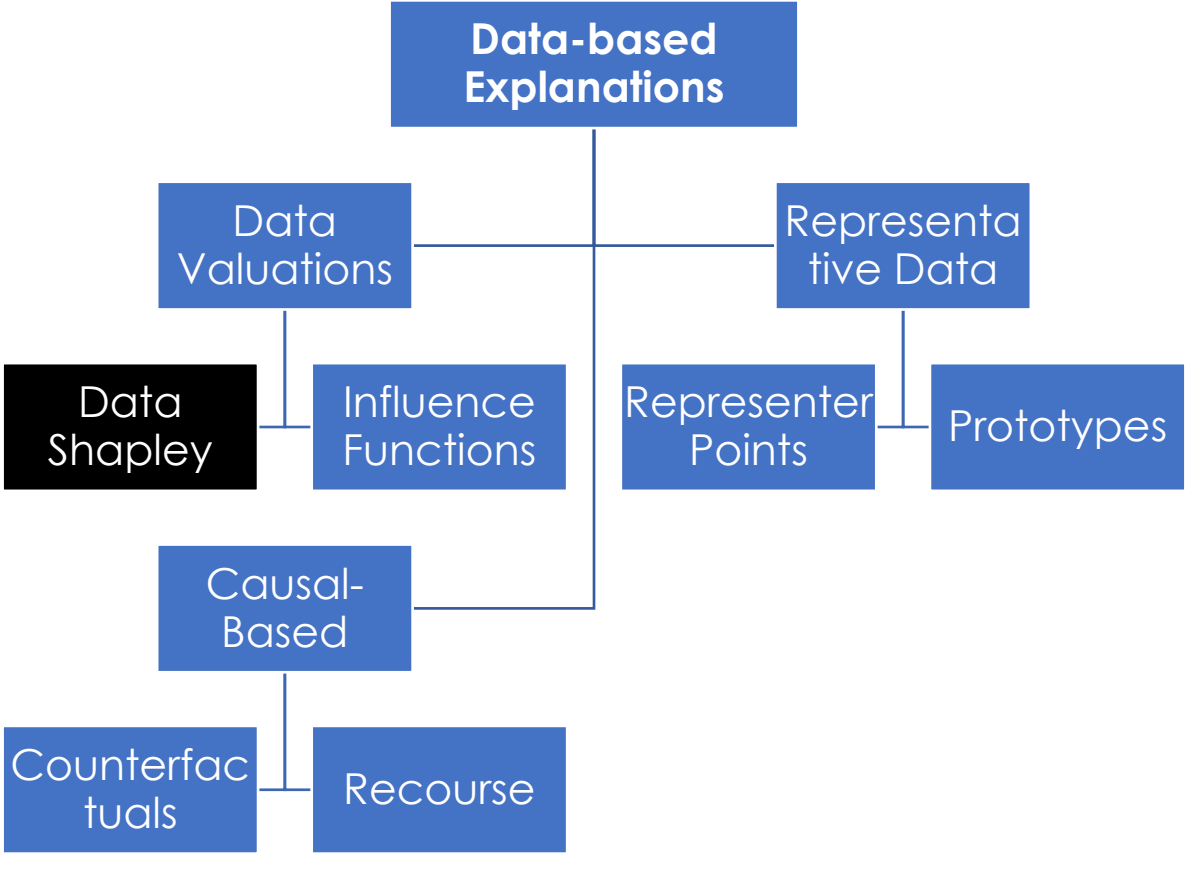
Probability that Rosa's income < 50K

find instances that are responsible for Rosa getting negative outcome

These instances are the most responsible for Rosa's negative outcome

Name	Age	Education	Marital	…	Race	Gender	Hours	Income
Nazia	36	Bachelors	Unmarried	…	Asian	female	40	<50K
Matt	26	Bachelors	Married	…	White	male	40	≥50K
Yeji	50	Masters	Married	…	Asian	male	16	≥50K
Neel	45	Masters	Unmarried	…	Black	male	28	<50K
…	…	…	…	…	…	…	…	…


Data-based Explanations
Data Valuations
Representative Data
Data Shapley
Influence Functions
Representer Points
Prototypes
Causal-Based
Counterfactuals
Recourse


Data-based Explanations
Data Valuations
Representative Data
Data Shapley
Influence Functions
Representer Points
Prototypes
Causal-Based
Counterfactuals
Recourse

Shapley Values for Data

- Given training data D and performance metric P, the value of a data point t_1 is given by Data Shapley Value

$$\text{Value}\ (t_1) = \underbrace{\sum_{\substack{\text{Subsets } S \subset D \\ \text{not containing } t_1}}}_{\text{All possible subsets}} \frac{\overbrace{P(S \cup t_1) - P(S)}^{\text{Marginal contribution of } t_1}}{\underbrace{\binom{|D|-1}{|S|}}_{\text{\# of subsets of size } |S| \text{ in } D}}$$

Expected contribution to all possible sizes of training data subsets

[2019 ICML Ghorbani et al.] Data Shapley: Equitable Valuation of Data for Machine Learning

Shapley Values for Data – Benefits

- The difference between the prediction and the average prediction is fairly distributed among the feature values of the instance
- Allows contrastive explanations; can compare value of a datum to a subset or even to a single data point
- Backed by solid theory (efficiency, symmetry, additivity axioms)

Shapley Values for Data – Challenges

- Is computationally expensive
  - For one data point: train and evaluate a model (expensive) for every possible subset of the training data (exponentially many)
- Can be misinterpreted
  - NOT the difference in the predicted values before and after removing the data point from model training
- Need access to the data to calculate the Shapley value for a new data point
- Does not account for distributional shift

if distribution change, prediction does not change
change in distribution does not affect it

Shapley Values for Data – Efficiency

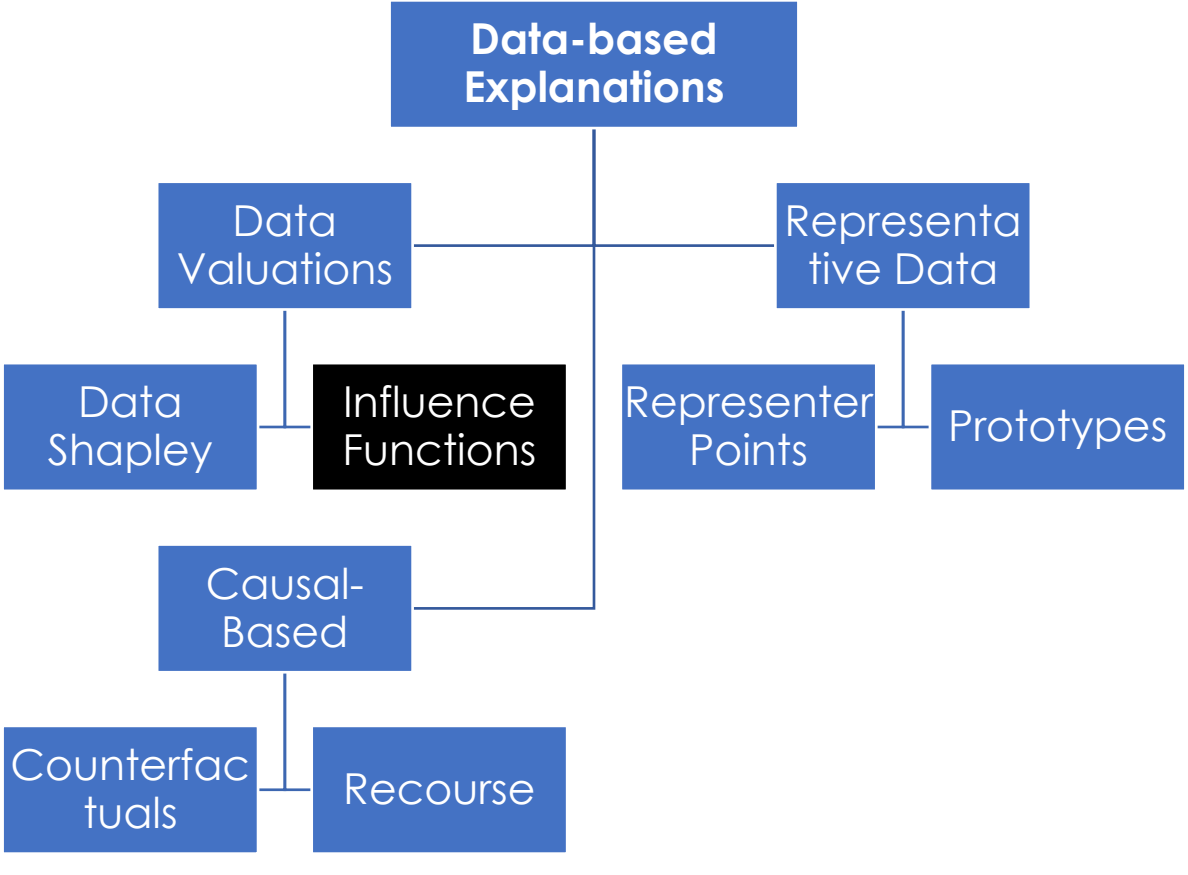
Sample data

- Monte-Carlo method (Ghorbani et al.)

$$\text{Value}\ (t_i) = \mathbb{E}_{\underbrace{\pi \sim \Pi}_{\text{Random permutation sampled from all permutations of data points}}}[V(S_\pi^i \cup i) - V(\underbrace{S_\pi^i}_{\text{Data appearing before i in a permutation}})]$$

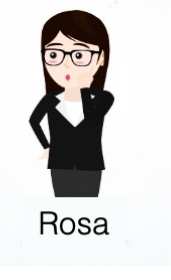
- Truncated Monte Carlo Shapley: Use early stopping when calculating data Shapley for a sampled permutation

[2019 ICML Ghorbani et al.] Data Shapley: Equitable Valuation of Data for Machine Learning


Data-based Explanations
Data Valuations
Representative Data
Data Shapley
Influence Functions
Representer Points
Prototypes
Causal-Based
Counterfactuals
Recourse

Influence of a Training Data Point

- How much does each training data point contribute to the learned model?



Name	Age	Education	Marital	…	Race	Gender	Hours
Rosa	34	Bachelors	Unmarried	…	Black	female	40

Rosa's data

Original Model

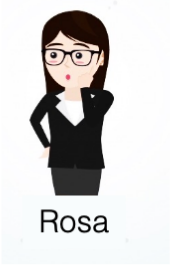
83%

Probability that Rosa's income < 50K

Name	Age	Education	Marital	…	Race	Gender	Hours	Income
Nazia	36	Bachelors	Unmarried	…	Asian	female	40	<50K
Matt	26	Bachelors	Married	…	White	male	40	≥50K
Yeji	50	Masters	Married	…	Asian	male	16	≥50K
Neel	45	Masters	Unmarried	…	Black	male	28	<50K
…	…	…	…	…	…	…	…	…

Influence of a Training Data Point

- Leave-one-out approach



Name	Age	Education	Marital	…	Race	Gender	Hours
Rosa	34	Bachelors	Unmarried	…	Black	female	40

Rosa's data

Updated Model

75%

Probability that Rosa's income < 50K

Name	Age	Education	Marital	…	Race	Gender	Hours	Income
~~Nazia~~	~~36~~	~~Bachelors~~	~~Unmarried~~	~~…~~	~~Asian~~	~~female~~	~~40~~	~~<50K~~
Matt	26	Bachelors	Married	…	White	male	40	≥50K
Yeji	50	Masters	Married	…	Asian	male	16	≥50K
Neel	45	Masters	Unmarried	…	Black	male	28	<50K
…	…	…	…	…	…	…	…	…

Influence of a Training Data Point

- Leave-one-out approach

Original model performance

Updated model performance

Value (

Name	Age	Education	Marital	…	Race	Gender	Hours	Income
Nazia	36	Bachelors	Unmarried	…	Asian	female	40	<50K
Matt	26	Bachelors	Married	…	White	male	40	≥50K
Yeji	50	Masters	Married	…	Asian	male	16	≥50K
Neel	45	Masters	Unmarried	…	Black	male	28	<50K
…	…	…	…	…	…	…	…	…

) = 0.83 – 0.75
= 0.08

- Might be expensive to compute for each data point
- Not reasonable when training data has duplicates

Influence Function

- Given data points $z \in Z$ and loss function L, the learning algorithm searches for a model with parameters $\theta \in \Theta$ such that

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta)$$

- Influence of training data point z_{train} (from robust statistics)
  - How do the model parameters change as we upweight z_{train} by an infinitesimal amount ϵ?

$$\theta_\epsilon = \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta) + \epsilon L(z_{\text{train}}, \theta)$$

[2017 ICML Koh et al.] Understanding Black-box Predictions via Influence Functions

Influence Function

- Original model parameters: θ, updated model parameters: θ_ϵ

- Influence of z_{train} on test loss is then measured as

$$\mathcal{I}_{\text{up,loss}}(z_{\text{train}}, z_{\text{test}}) = L(z_{test}, \theta_\epsilon) - L(z_{test}, \theta^*)$$

$$\stackrel{\text{def}}{=} \left.\frac{dL(z_{\text{test}}, \hat{\theta}_{\epsilon, z_{\text{train}}})}{d\epsilon}\right|_{\epsilon=0}$$

$$= -\underbrace{\nabla_\theta L(z_{\text{test}}, \hat{\theta})^\top}_{\text{Gradient of test loss}} H_{\hat{\theta}}^{-1} \underbrace{\nabla_\theta L(z_{\text{train}}, \hat{\theta})}_{\text{Gradient of training data loss}},$$

where $\underbrace{H_{\hat{\theta}} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta^2 L(z_i, \hat{\theta})}_{\text{Hessian matrix}}.$

Influence Function – Challenges

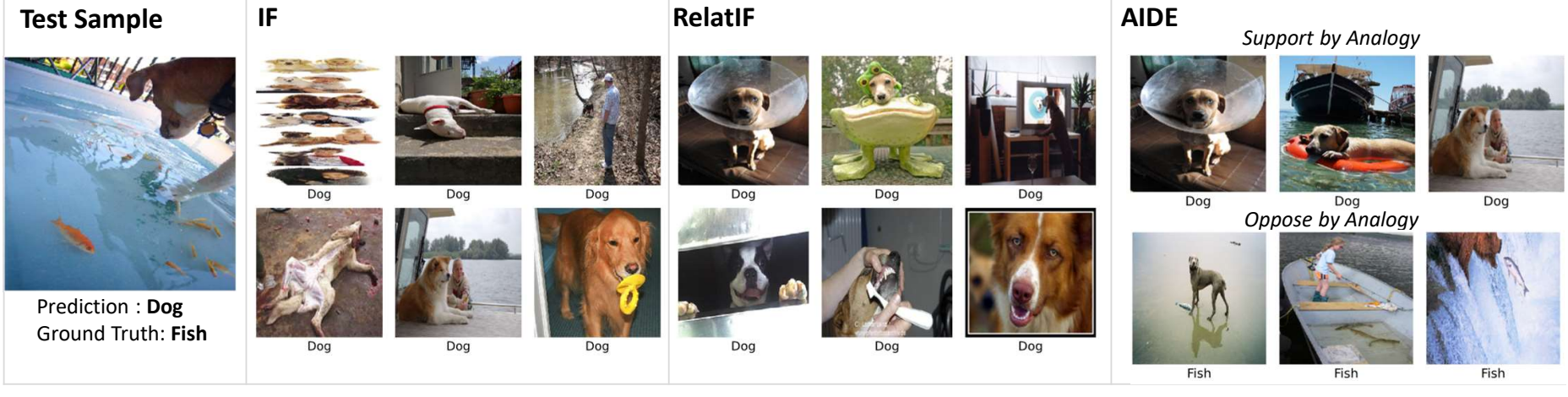
- Computationally expensive
  - Computing and inverting the Hessian matrix
    - ($O(np^2 + p^3)$), n: #data points, p: #model parameters
  - Computing the influence over test loss for all training points

- Applicable to models with twice-differentiable loss function
  - Alternatives for non-parametric gradient-boosted decision trees

- Influence functions are approximate: first-order approximations; the approach forms a quadratic expansion around the parameters

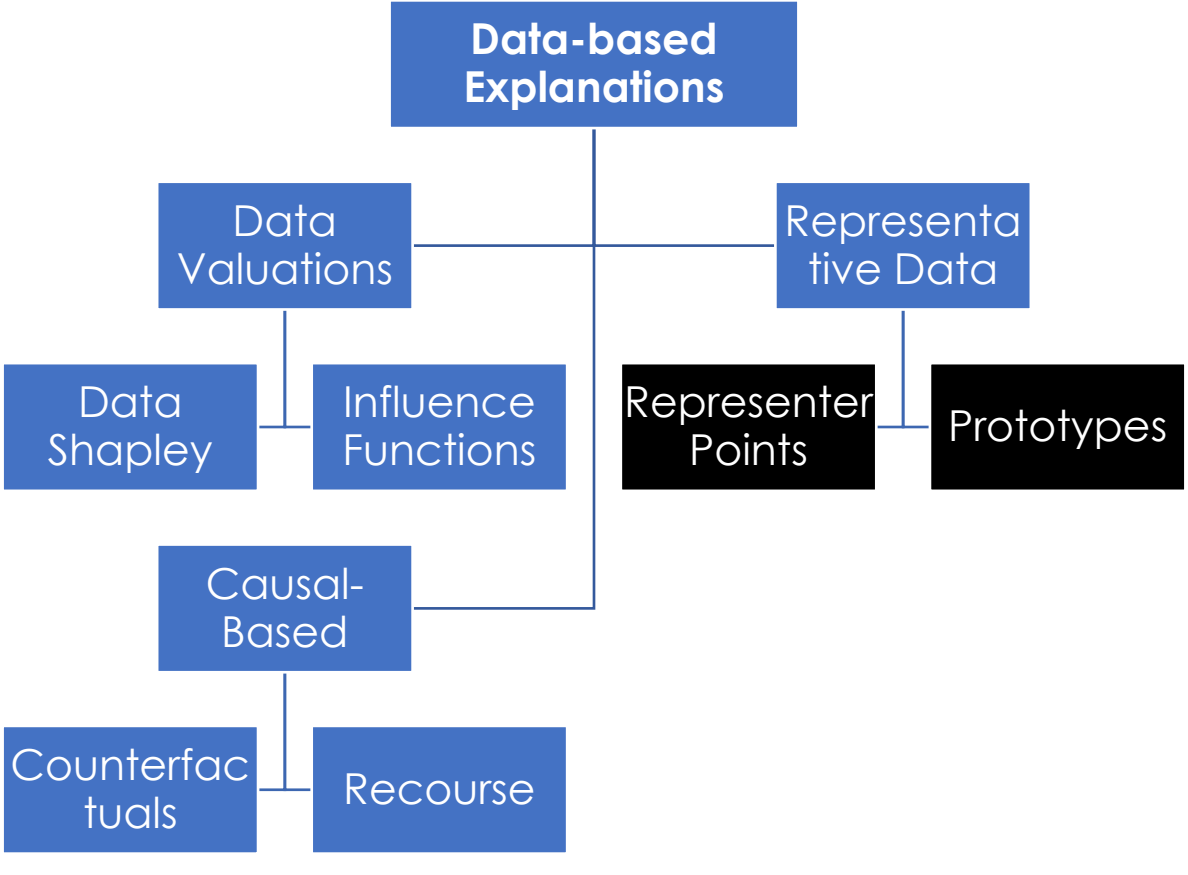
- No clear threshold to decide when a training data point is influential or non-influential

Influence Function – Example



Model says it has predicted Dog
But it is Fish
Why it was predicted as Dog?

[Nematov et al.] AIDE: Antithetical, Intent-based, and Diverse Explanations


Data-based Explanations
Data Valuations
Representative Data
Data Shapley
Influence Functions
Representer Points
Prototypes
Causal-Based
Counterfactuals
Recourse

Representer Points

- Representer points
  - Decompose a predictor into a linear combination of functions of training data points
  - Weighted similarities between test data point and each training data point

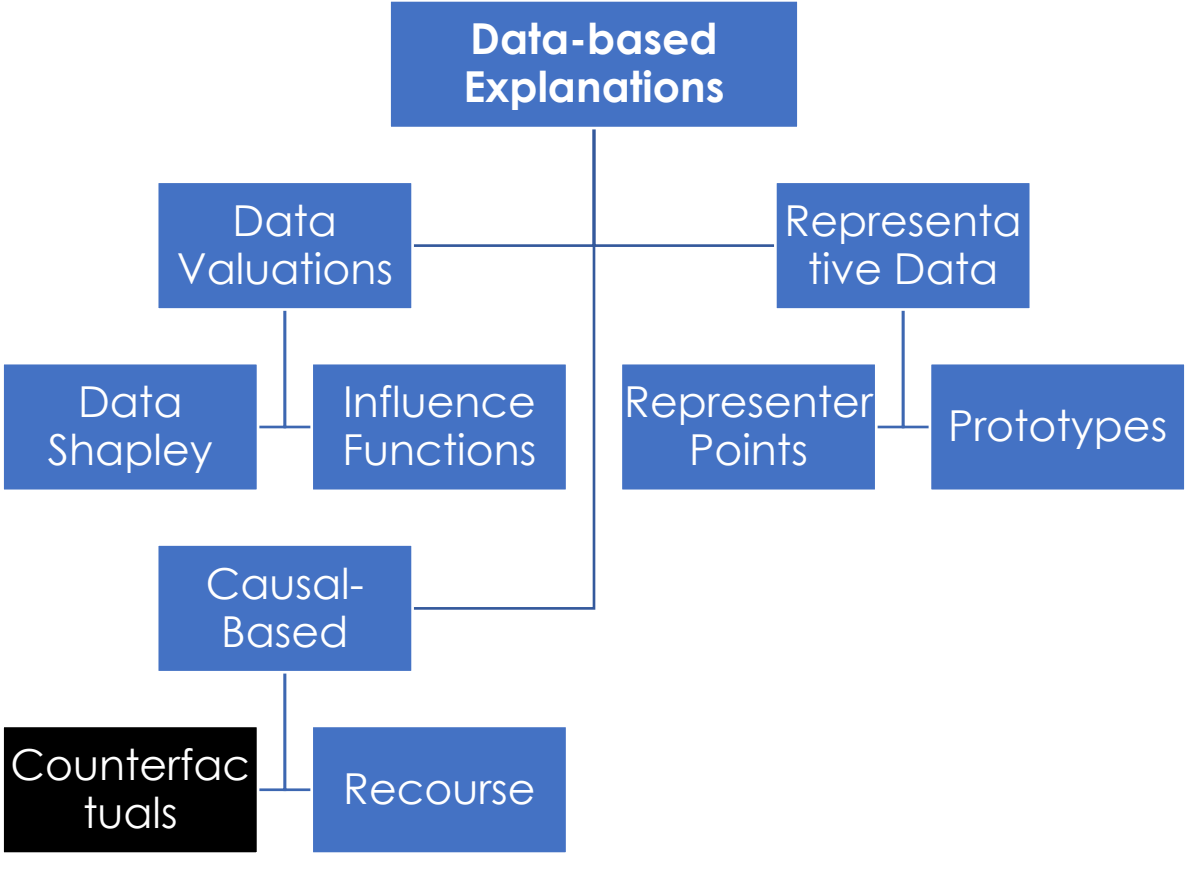
[2018 NeurIPS Yeh et al.] Representer Point Selection for Explaining Deep Neural Networks

Data Prototypes

- **Prototypes:** data points that are representative of all the data

- **Criticisms:** data points not represented by prototypes

[2016 NeurIPS Kim et al.] Examples are not enough, learn to criticize! Criticism for interpretability


Data-based Explanations
Data Valuations
Representative Data
Data Shapley
Influence Functions
Representer Points
Prototypes
Causal-Based
Counterfactuals
Recourse

Motivation – Causality

It explains
Why Y occured?
Because X occured

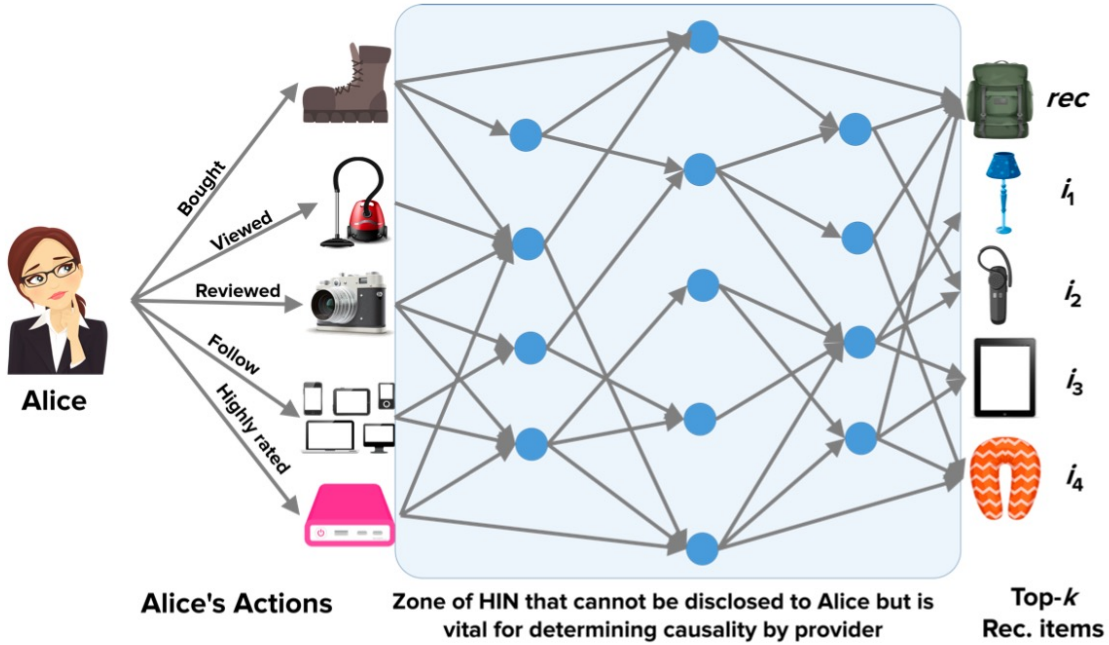
- consider a **causal relationship**: "*If X had not occurred, Y would not have occurred*"
- it explains why Y occurred: it's because X occurs

- a **counterfactual explanation** of a specific **outcome** describes the smallest **change** to **preferences** that results in **not seeing** that outcome
  - Example: to explain "Why was I recommended item Y?" look for smallest changes in preferences so that item Y no longer appears in the recommendations

- Original input = **factual**
- Fictitious, changed input = **counterfactual**

Motivation – Causality

Why I got this recommendation?
Because you clicked on these things



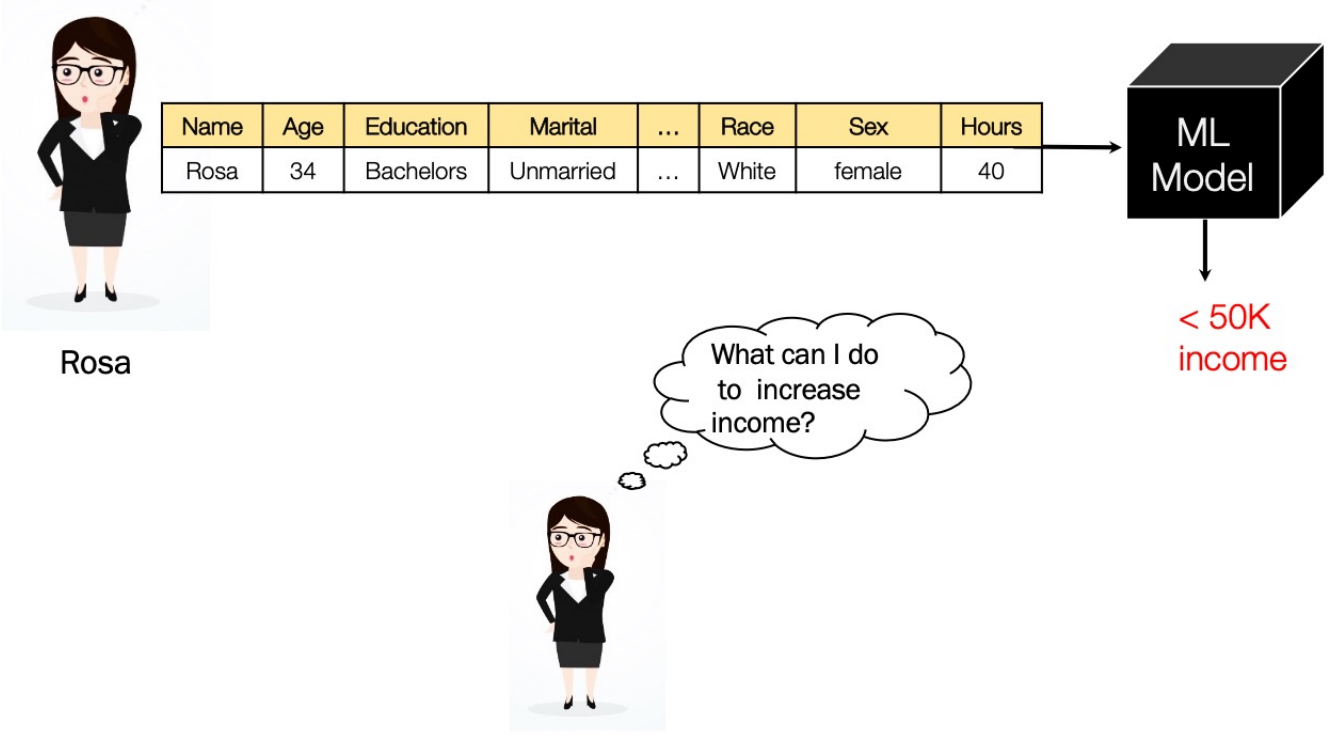
Alice: Why did I receive this recommendation "Jack Wolfskin backpack"?

PRINCE: You **bought** "Adidas Hiking Shoes";
You **reviewed** "Nikon Coolpix Camera" with "Sleek! Handy on hikes!";
You **rated** "Intenso Travel Power Bank" highly.

If you **had not** done these actions:
"iPad Air" **would have replaced** "Jack Wolfskin backpack".

[2020 WSDM A. Ghazimatin et al.] *PRINCE: Provider-side Interpretability with Counterfactual Explanations in Recommender Systems*

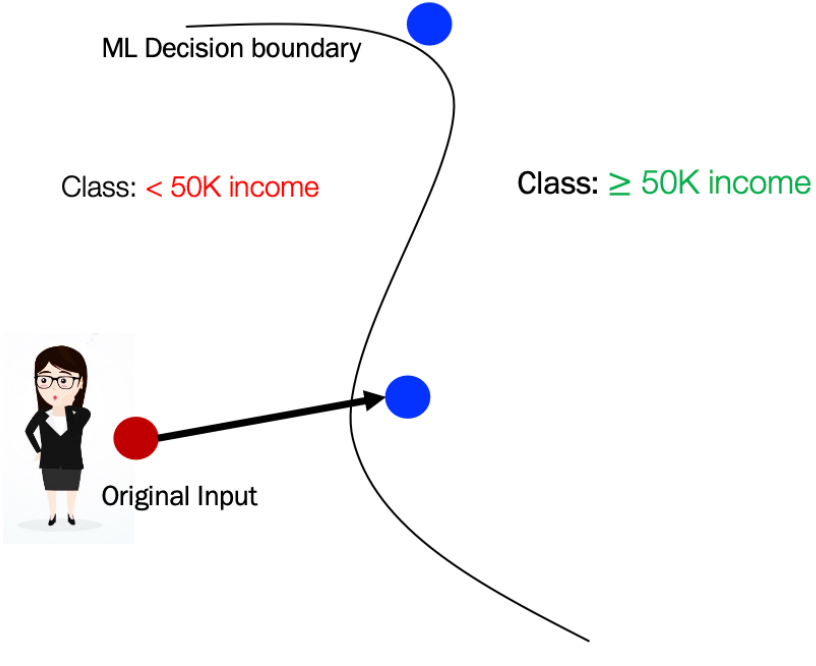
Counterfactual Explanations



Counterfactual Explanations

What is to be changed (MINIMUM) to get the prediction other way around?

- What features need to be changed and by how much to flip a model's prediction?



Counterfactual Explanations

Goal: **Minimum change in input attributes to flip a model's prediction**

Popular Techniques

- Feature-based explanations: Feature score captures its likelihood to change outcome

yo feature eti ley change garyo bhane outcome change hune chance eti huncha

- Instance-level explanations: Identify counterfactual scenarios to change outcome
  - Inverse Classification
  - Nearest-counterfactual explanation

sab same cha
euta feature differenct huda result different cha bhane

Inverse Classification

Find another instance with similar feature with Rosa but opposite prediction

- Identify a close neighbor of the point which has the opposite outcome

Name	Age	Education	Marital	…	Race	Sex	Hours
Rosa	34	Bachelors	Unmarried	…	White	female	40

ML Model → < 50K income

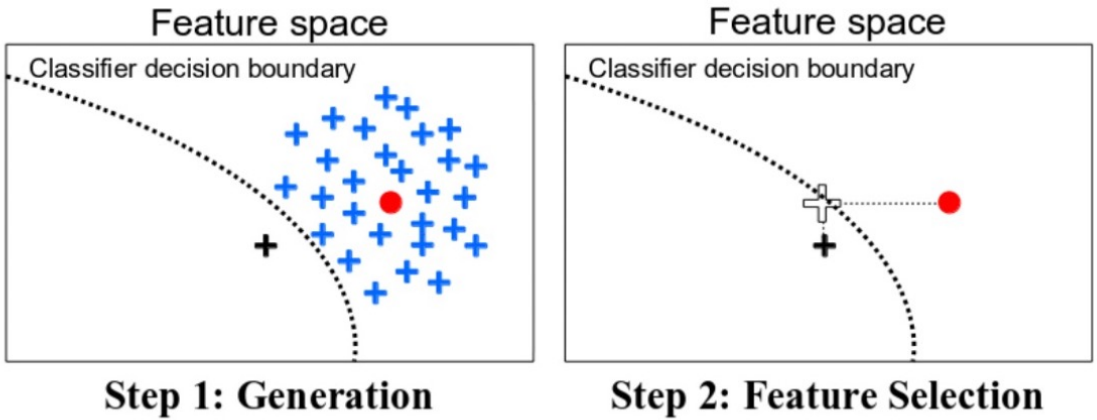
Name	Age	Education	Marital	…	Race	Sex	Hours
Counterfactual Rosa	34	Bachelors	Unmarried	…	White	female	65

ML Model → ≥ 50K income

[IPMU 2018 Laugel et al.] Inverse Classification for Comparison-based Interpretability in Machine Learning

Inverse Classification

- Identify a close neighbor of the point which has the opposite outcome
- Growing sphere algorithm



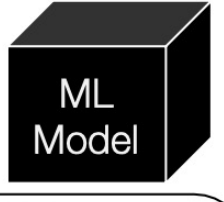
[IPMU 2018 Laugel et al.] Inverse Classification for Comparison-based Interpretability in Machine Learning

DICE: Diverse Counterfactual Explanations

- Post-hoc, "what-if" style of explanations
- Generate multiple diverse counterfactuals that receive a favorable outcome

multiple

Name	Age	Education	Marital	…	Race	Sex	Hours
Rosa	34	Bachelors	Unmarried	…	White	female	40



Counterfactual Examples by DiCE

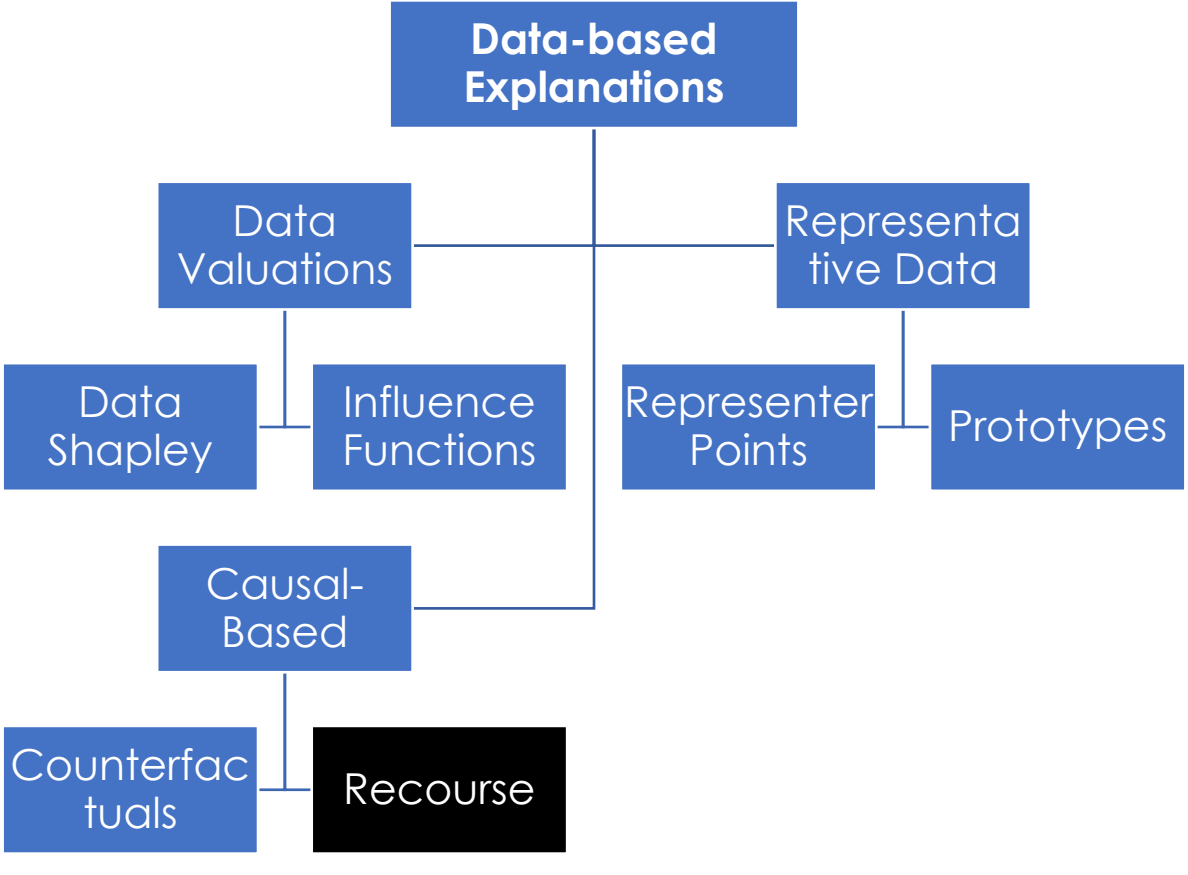
1. Increase hours to at least 65
2. Change education to Masters and increase work experience by 1 year

[2020 ACM FAT* Mothilal et al.] Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations

DICE – Algorithm

- Model the search space of counterfactuals as an optimization problem

- Three different objectives
  - Proximity to original input original bata najik ko herne
  - Sparsity of change kam change bhako
  - Diversity of explanation examples aru instance pani herne

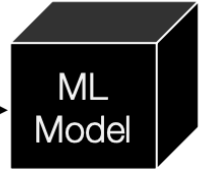

Data-based Explanations
Data Valuations
Representative Data
Data Shapley
Influence Functions
Representer Points
Prototypes
Causal-Based
Counterfactuals
Recourse

Algorithmic Recourse

Only look for things that can be changed

- **Input**: Individual with negative outcome
- **Goal**: Identify smallest cost intervention that can flip the outcome in the future

Name	Age	Education	Marital	…	Race	Sex	Hours
Rosa	34	Bachelors	Unmarried	…	White	female	40



What should I do to flip the outcome?

Algorithmic Recourse

- Model recourse as an optimization problem
- Linear Program:
  - Objective:

    Minimize cost(

Age	Education	Marital	…	Hours
a1	a2	a3	…	ak

)
  - Constraint

    f (

Name	Age	Education	Marital	…	Race	Sex	Hours
Rosa	34	Bachelors	Unmarried	…	White	female	40

    +

Age	Education	Marital	…	Hours
a1	a2	a3	…	ak

) =1
- Assumes f is a linear function

[2019 FAT Ustun et al.] Actionable Recourse in Linear Classification

ARES – Global Recourse Summary

- explain not a specific instance but the whole model (global explainability)

If Foreign_Worker = True: ← Subgroup descriptor

Recourse rule 1:
If Years_At_Current_Home <= 1 *and* Num_Loans_Other_Bank > 0,
then Years_At_Current_Home >= 3 *and* Num_Loans_Other_Bank = 0

Recourse rule 2:
If Job_Class_Skilled = False *and* Years_At_Current_Job < 1,
then Job_Class_Skilled = False *and* Years_At_Current_Job >= 4

If Foreign_Worker = False: ← Subgroup descriptor

Recourse rule 3:
If Missed_Payments = True *and* Critical_Loans_Elsewhere = True,
then Missed_Payments = True *and* Critical_Loans_Elsewhere = False

[2020 NeurIPS Rawal et al.] Beyond Individualized Recourse: Interpretable and Interactive Summaries of Actionable Recourses

FACTS – Recourse and Fairness

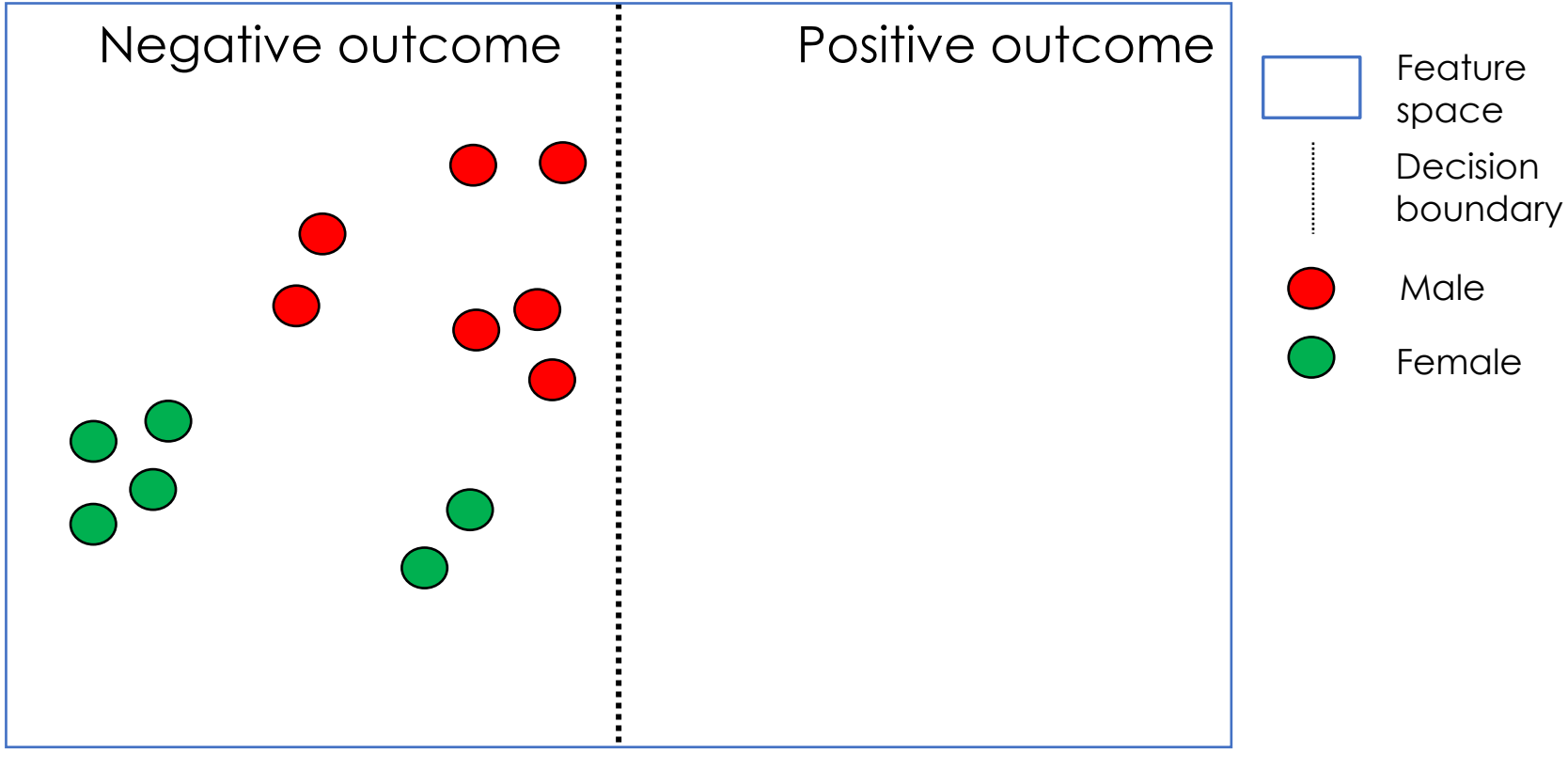
- recourse = the way to reverse an unfavorable outcome

- fairness of recourse = does the recourse cost the same?

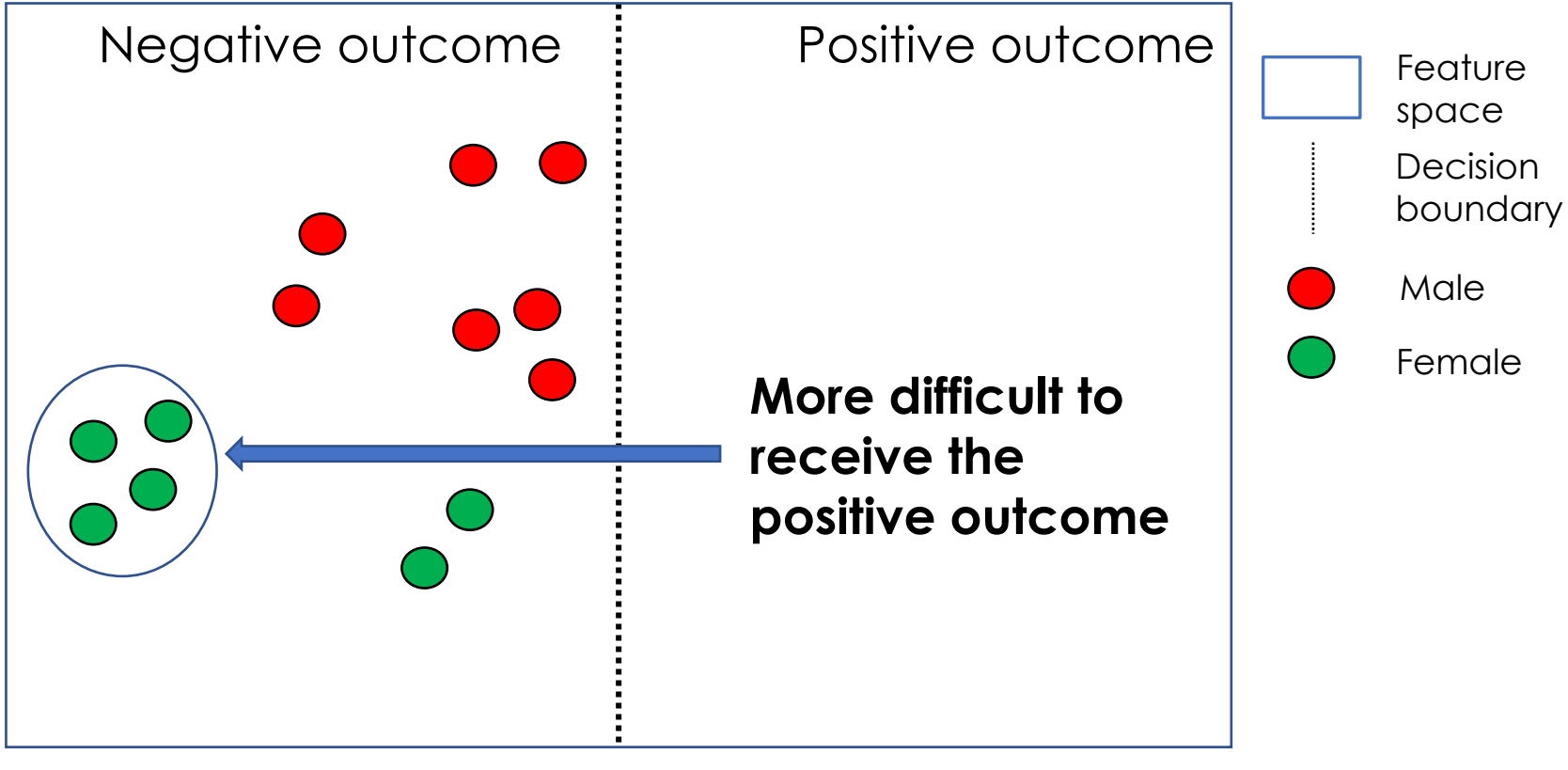
[2023 NeurIPS Kavouras et al.] Fairness-Aware Counterfactuals for Subgroups

FACTS – Recourse

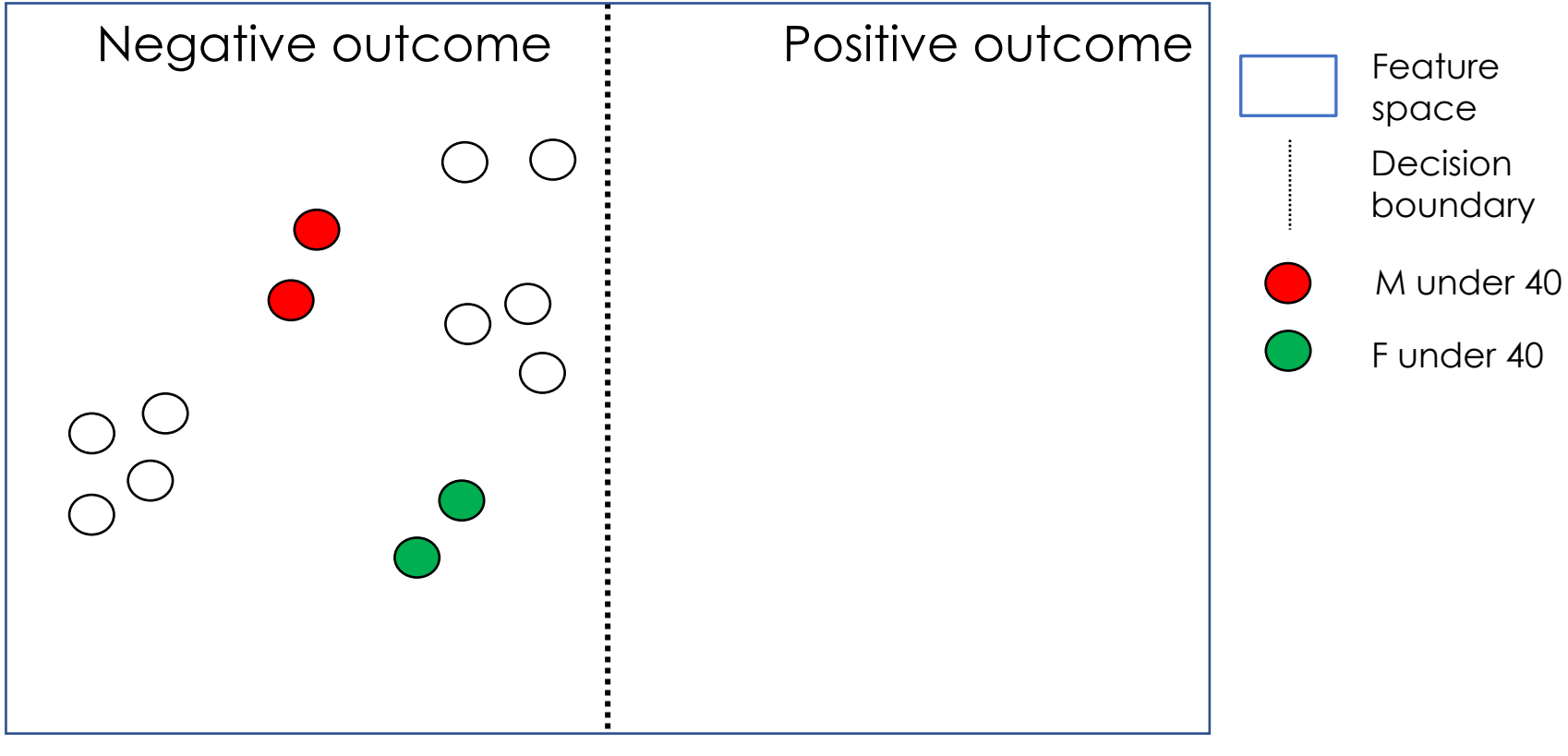
Cost is more for crossing boundary for female



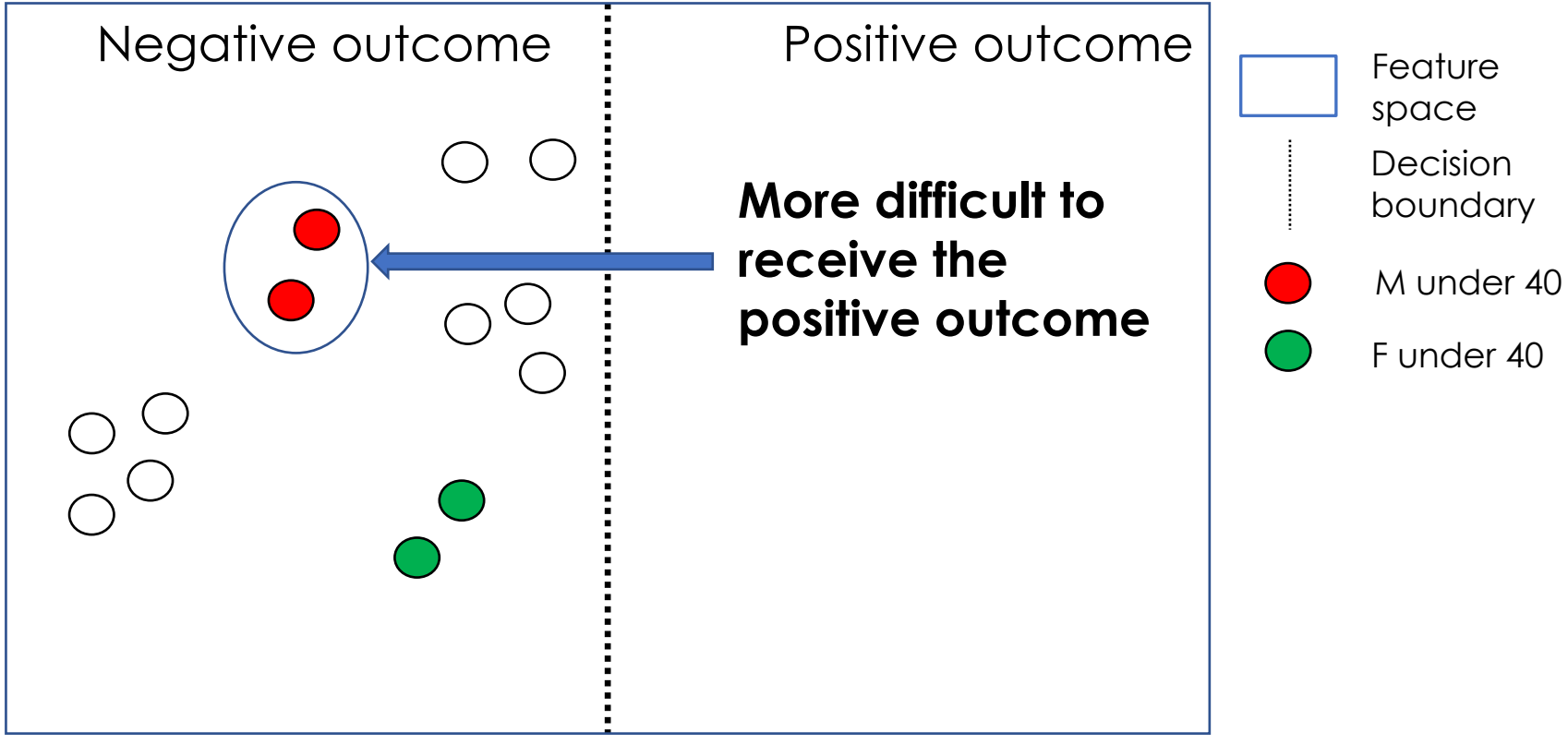
FACTS – Fairness of recourse



FACTS – Fairness of recourse in subgroups



FACTS – Fairness of recourse in subgroups



FACTS – Definitions for Fair Recourse

- Equal Effectiveness
- Equal Choice for Recourse
- Equal Effectiveness within Budget
- Equal Cost of Effectiveness
- Fair Effectiveness-Cost Trade-off
- Equal Mean Recourse

FACTS – Example

```
If capital-gain = 0, education-num = 10, hours-per-week = FullTime, race =  White:
        Protected Subgroup ' Male', 8.40% covered
                Make hours-per-week = OverTime with effectiveness 8.92% and counterfactual cost = 2.0.
                Make education-num = 11 with effectiveness 8.92% and counterfactual cost = 3.0.
                Make hours-per-week = BrainDrain with effectiveness 16.98% and counterfactual cost = 4.0.
                Make education-num = 11, hours-per-week = OverTime with effectiveness 23.16% and counterfactual cost = 5.0.
                Make education-num = 12 with effectiveness 23.16% and counterfactual cost = 6.0.
                Make education-num = 12, hours-per-week = OverTime with effectiveness 31.56% and counterfactual cost = 8.0.
                Aggregate cost of the above recourses = 8.00
        Protected Subgroup ' Female', 8.32% covered
                Make hours-per-week = OverTime with effectiveness 1.79% and counterfactual cost = 2.0.
                Make education-num = 11 with effectiveness 1.79% and counterfactual cost = 3.0.
                Make hours-per-week = BrainDrain with effectiveness 3.27% and counterfactual cost = 4.0.
                Make education-num = 11, hours-per-week = OverTime with effectiveness 4.46% and counterfactual cost = 5.0.
                Make education-num = 12 with effectiveness 4.46% and counterfactual cost = 6.0.
                Make education-num = 12, hours-per-week = OverTime with effectiveness 5.95% and counterfactual cost = 8.0.
                Aggregate cost of the above recourses = inf
        Bias against  Female. Unfairness score = inf.
```


[2023 NeurIPS Kavouras et al.] Fairness-Aware Counterfactuals for Subgroups

Acknowledgements

- Some material from [2022 ACM SIGMOD Pradhan et al.] Explainable AI: Foundations, Applications, and Opportunities for Data Management Research https://explainable-ai-tutorial.github.io