Dirk **Fahland**

based on joint work with
Wil v.d. Aalst, Boudewijn v. Dongen,
Marlon Dumas, Massimiliano de Leoni,
Luciano Garcia Banuelos, Viara Popova

Process Mining for Artifact-Centric Processes

TU/e Technische Universiteit Eindhoven University of Technology

Where innovation starts

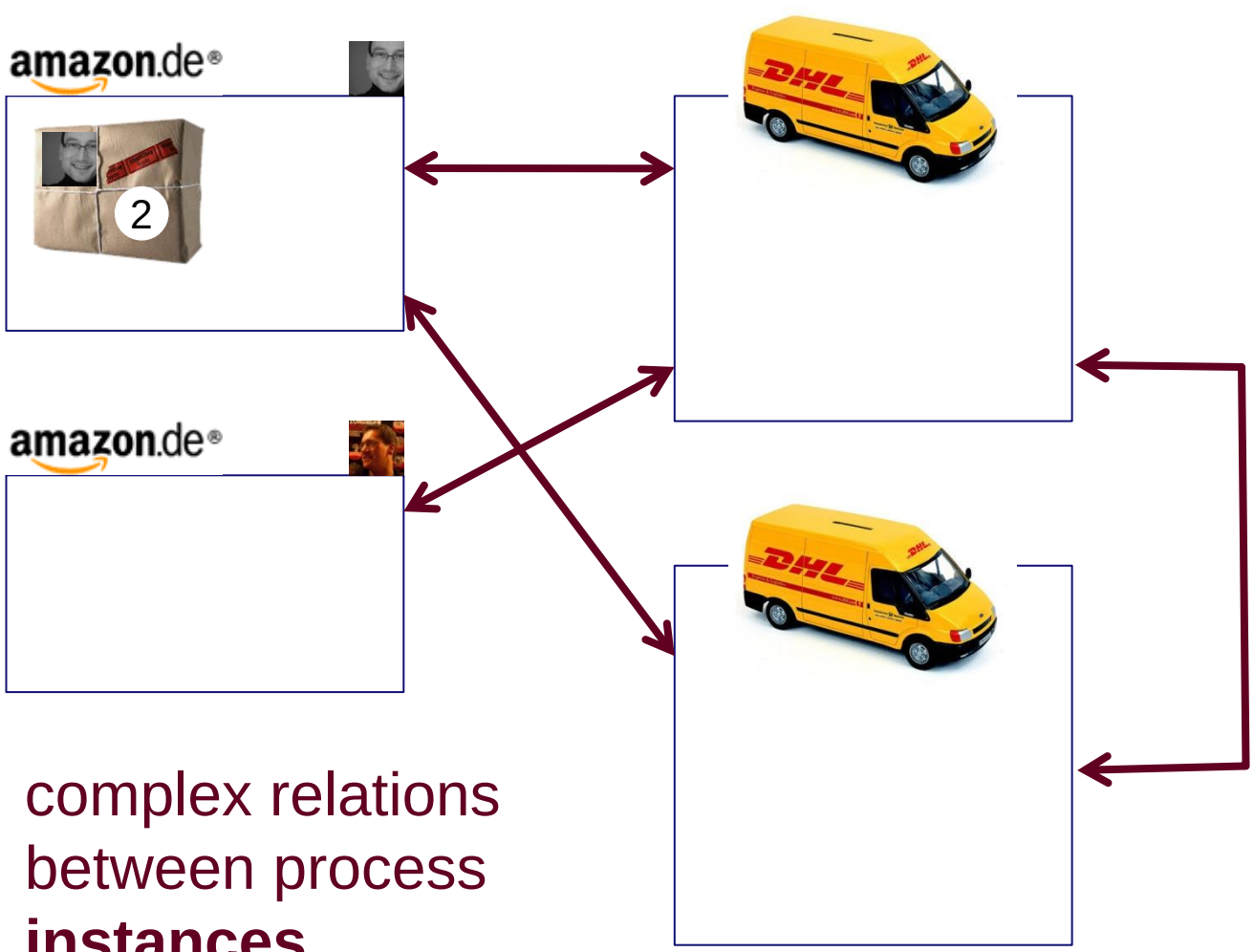
A process that needs multiple instances

A process that needs multiple instances

A process that needs multiple instances

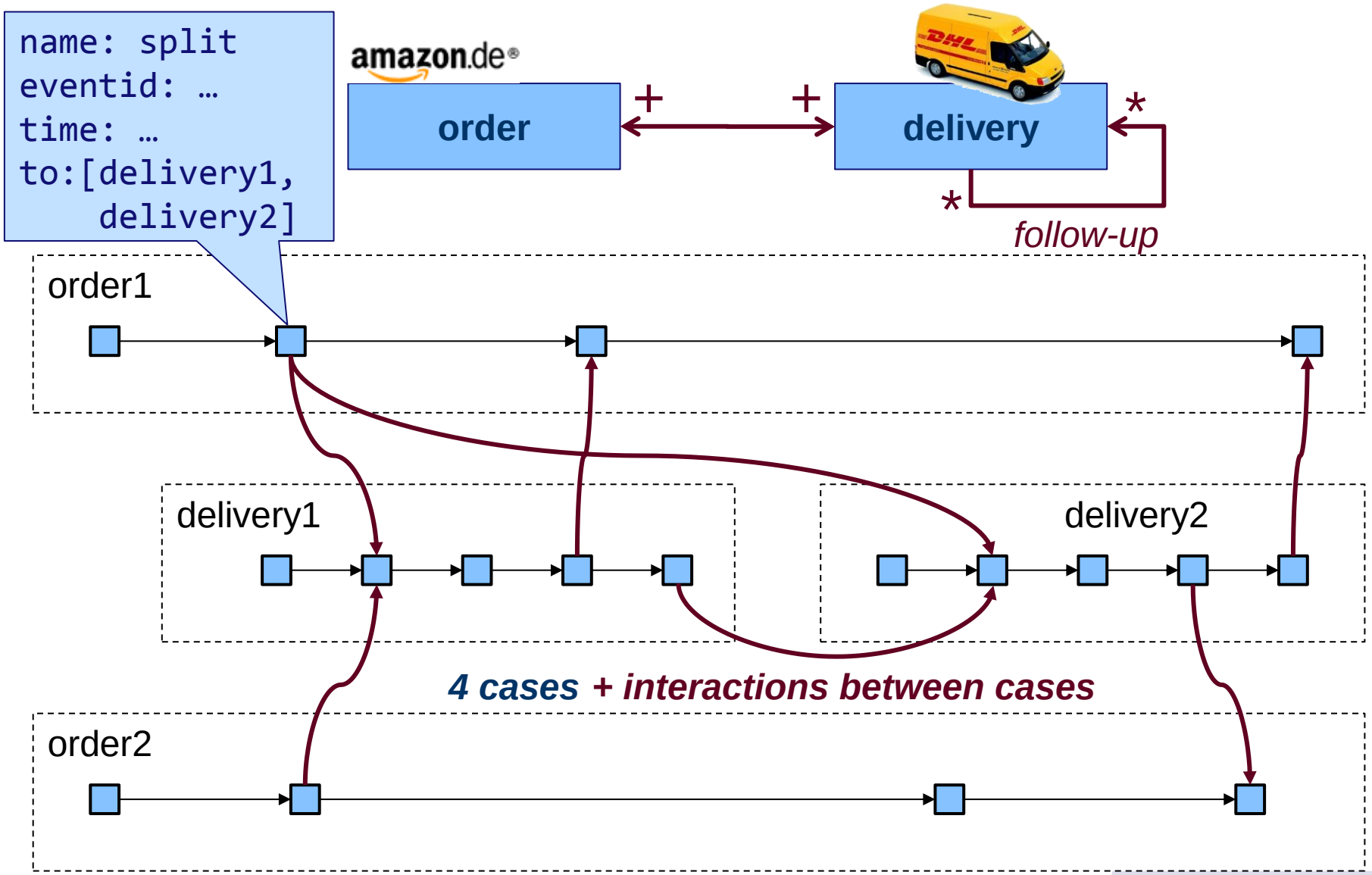
A process that needs multiple instances

A process that needs multiple instances



complex relations between process **instances**

Behavior observed in this process



Classical Process Discovery

c1: A B C D E
c2: A C B D E
c3: A F D E
…

assumptions

- case = sequence of events of this case
- cases are isolated:
  event A in c1 happens only in c1 (and not in c2)
- cases of the same process

- **one unique case id,**
- **each event associated to exactly one case id**

Classical Process Discovery

c1: A B C D E
c2: A C B D E
c3: A F D E
…

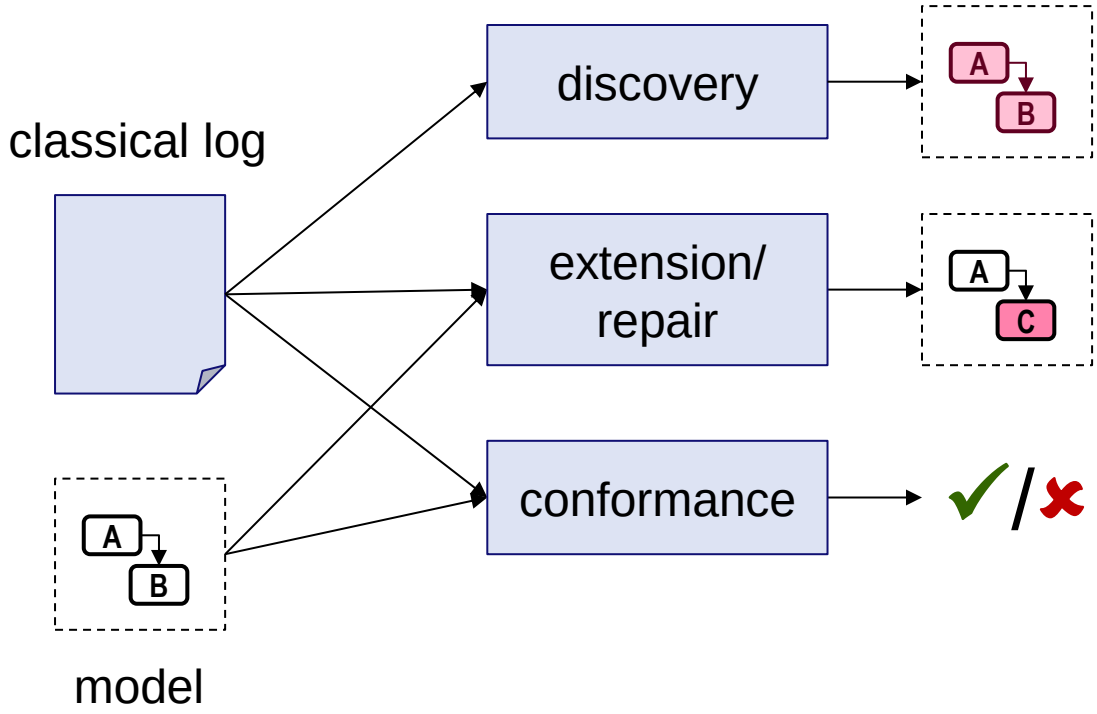
assumptions

- case = sequence of events of this case
- cases are isolated:
  event A in c1 happens [illegible] 1 (and not in c2)
- case

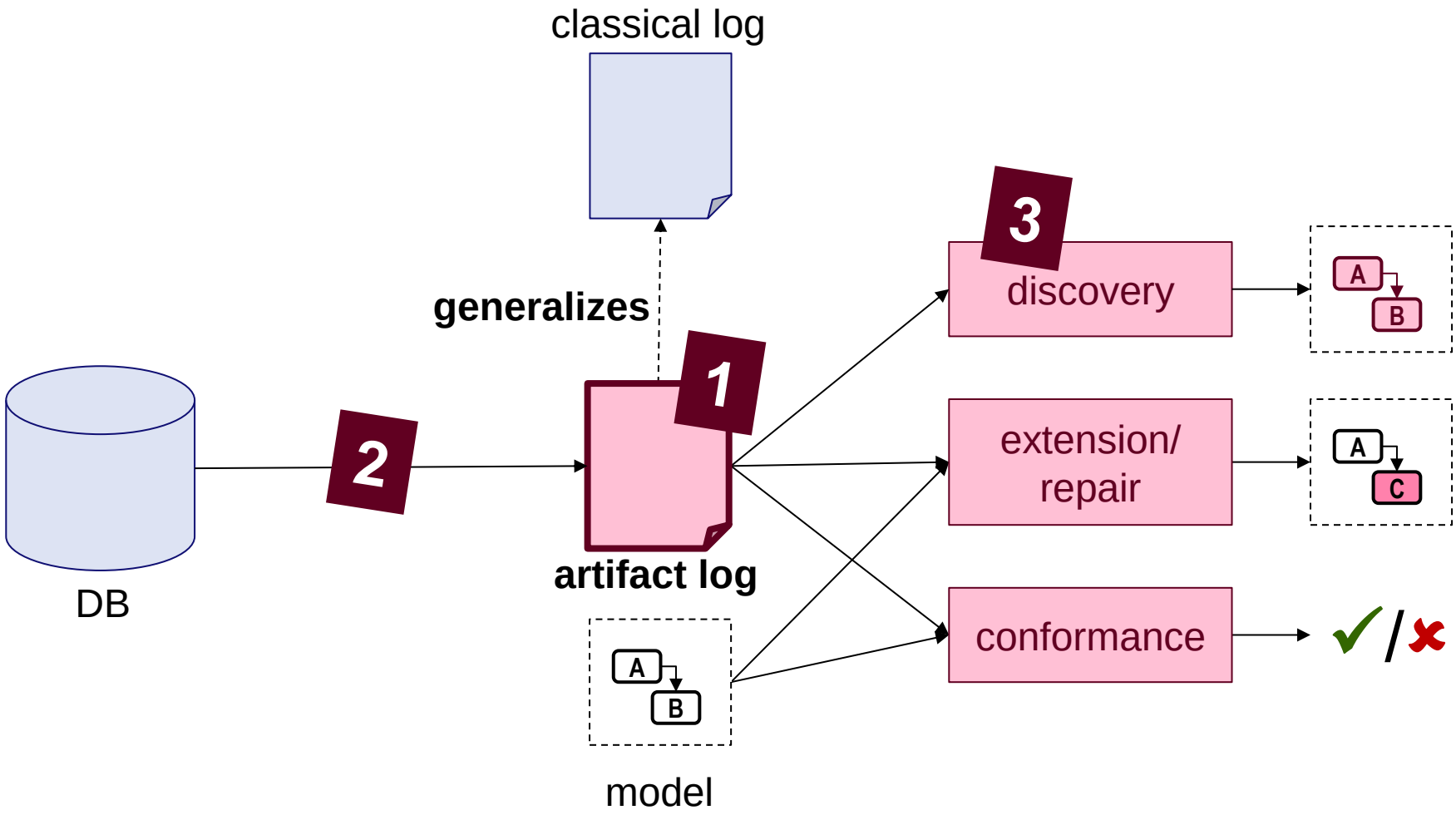
reality in many processes

- **one [illegible] case id,**
- **each event associated to exactly one case id**

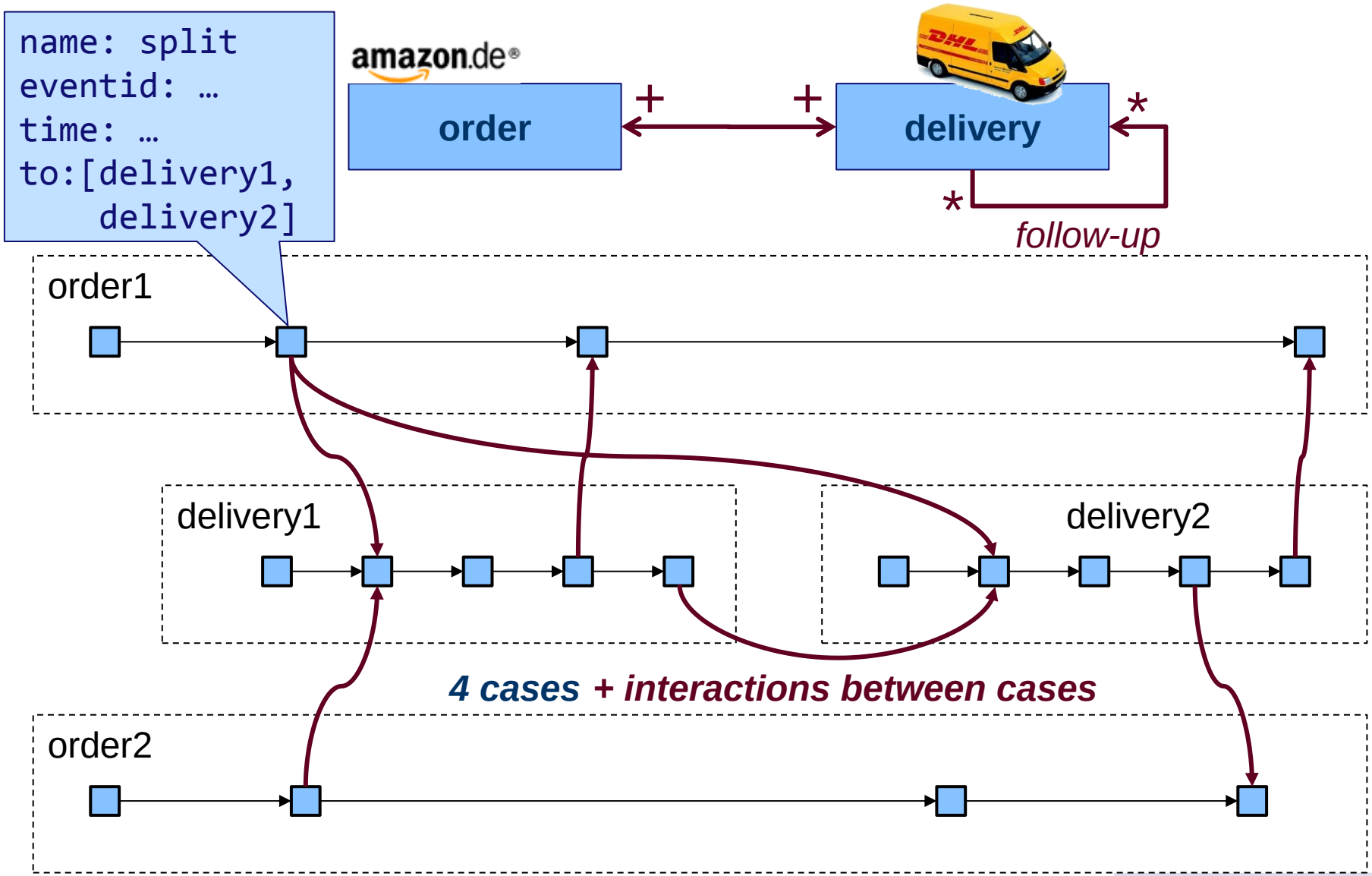
Process Mining (classical)



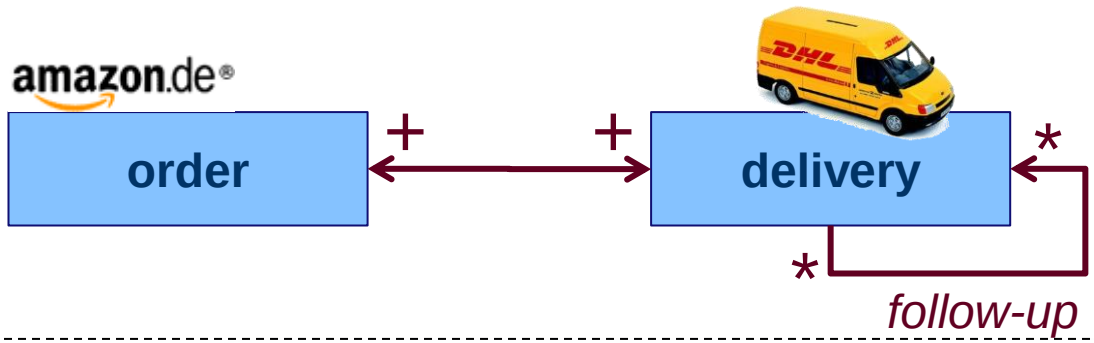
Process Mining for Artifacts

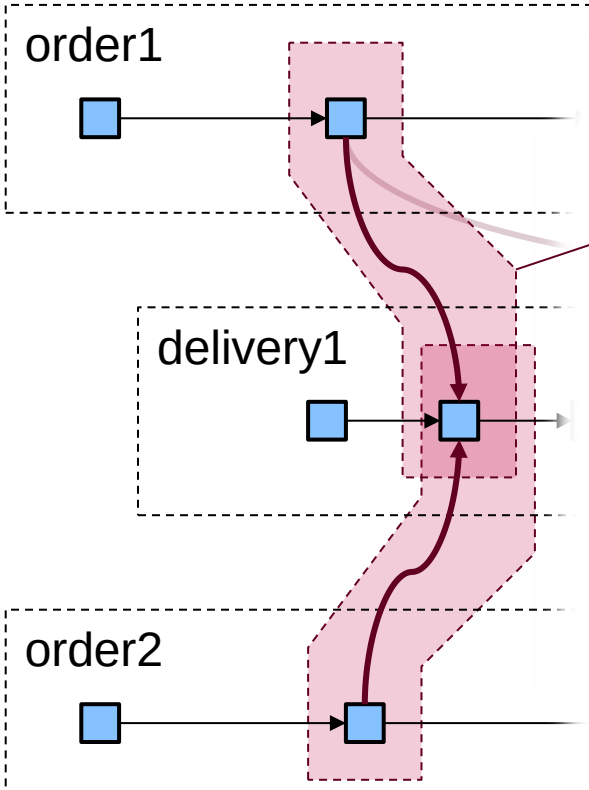


Artifact Log - Example



Artifact Log - Example



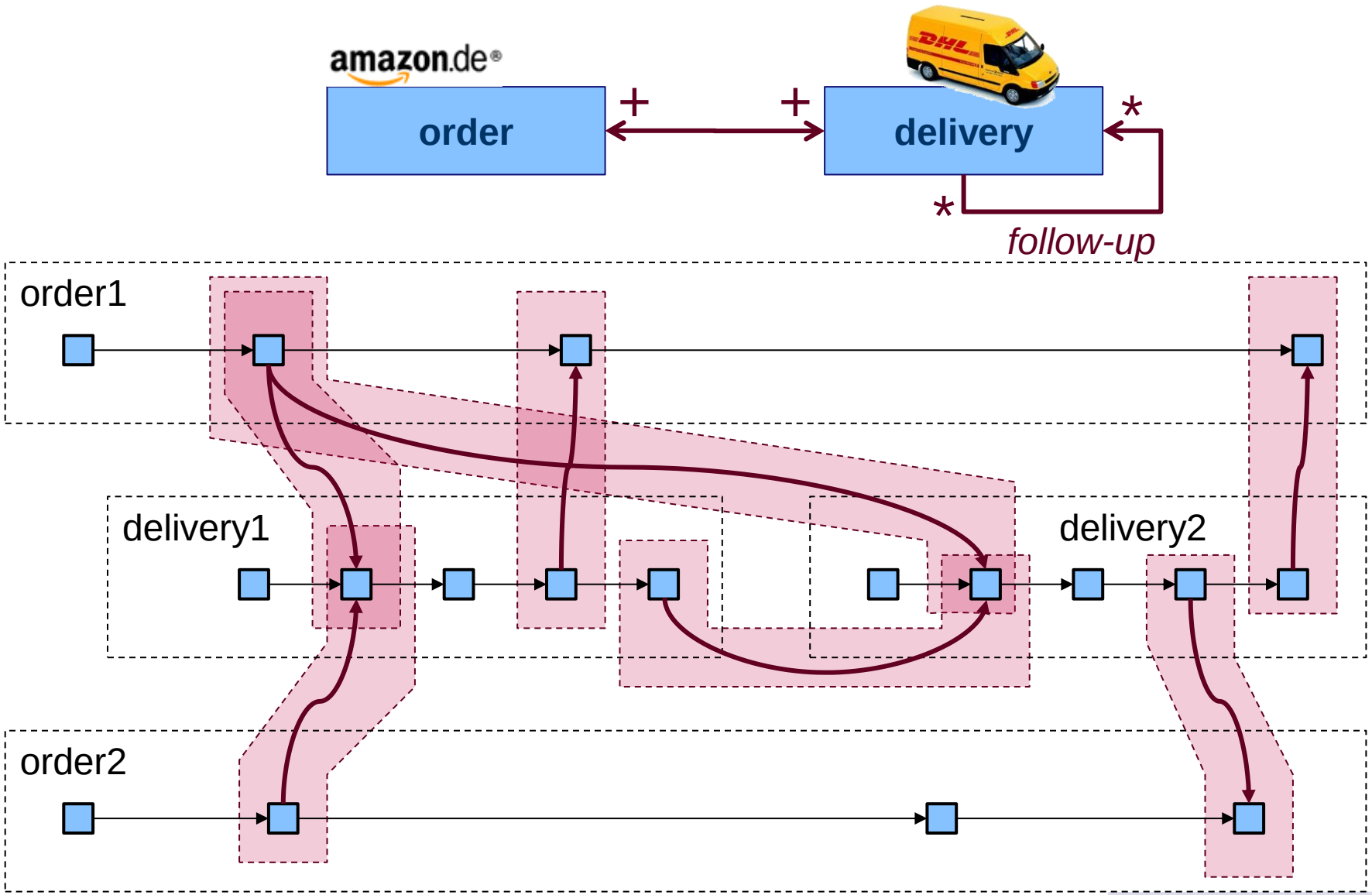


capture artifact interaction by a case: {send-event, receive-event}

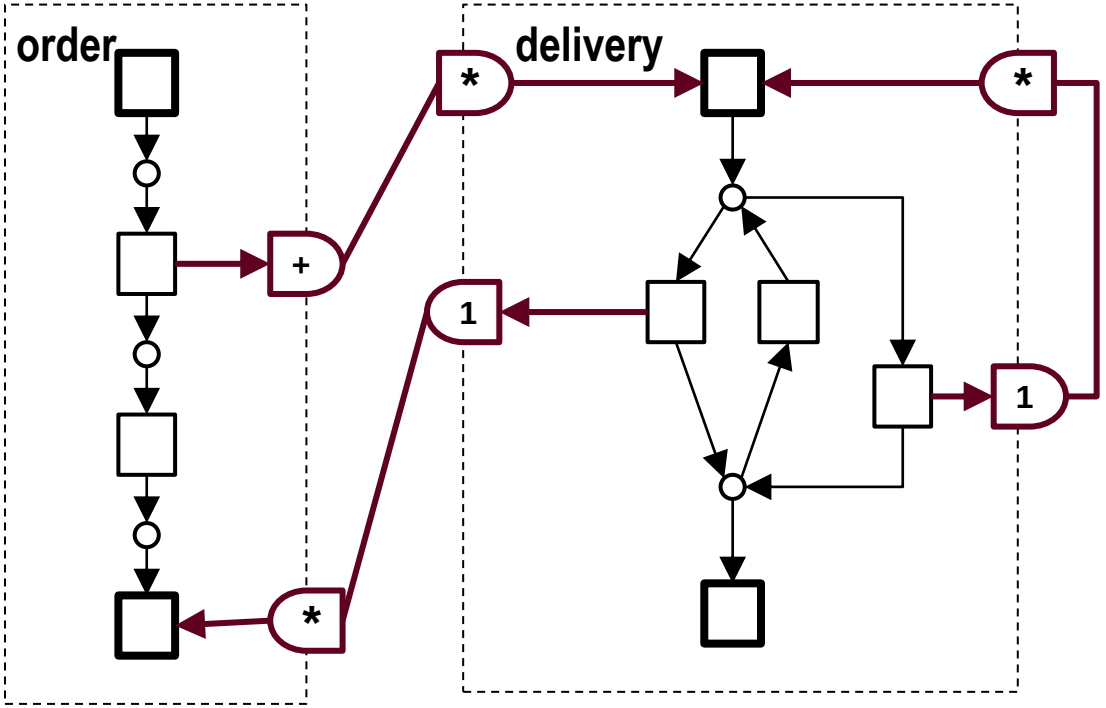
event can participate in **several cases (cases overlap on joint events)**

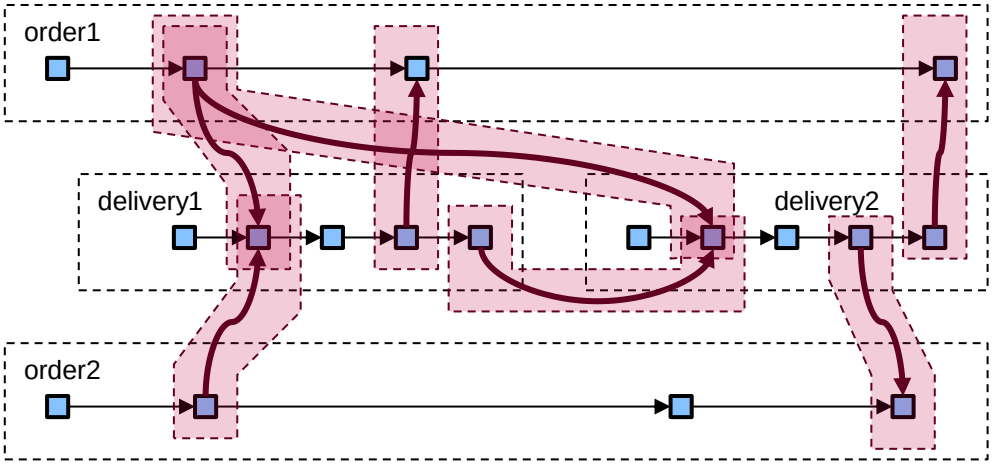
events are ordered within a case by their timestamps

Artifact Log - Example



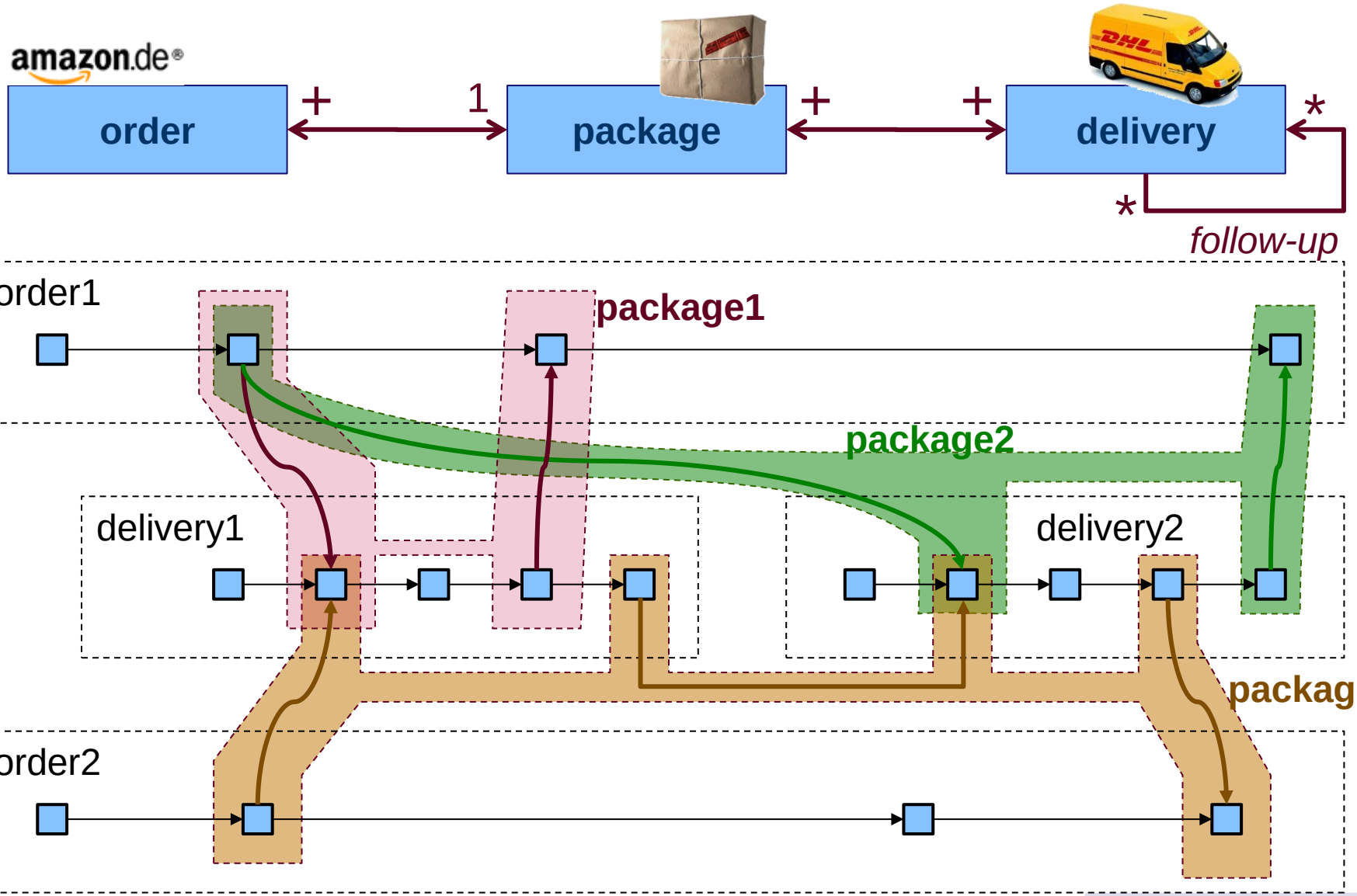
Artifact logs and models (Asynchronous)



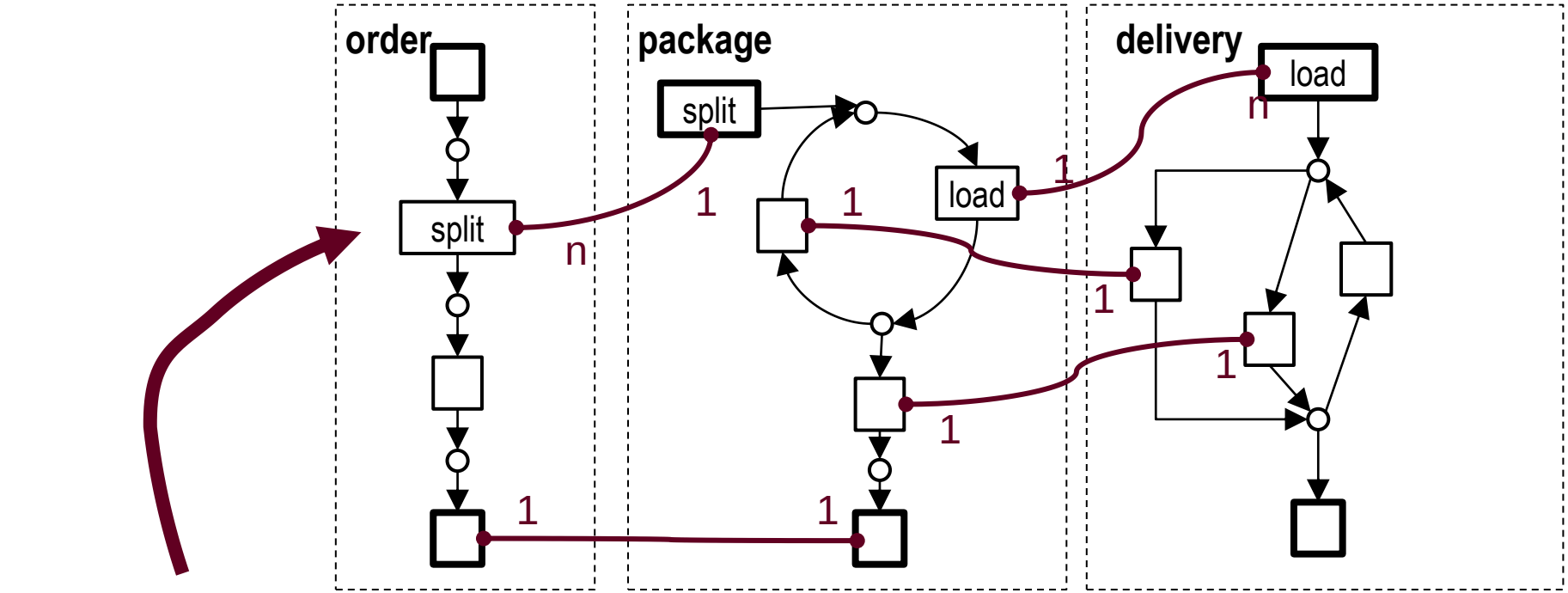


- cases ⇔ life-cycles
- interaction cases ⇔ channels

Different correlation of events

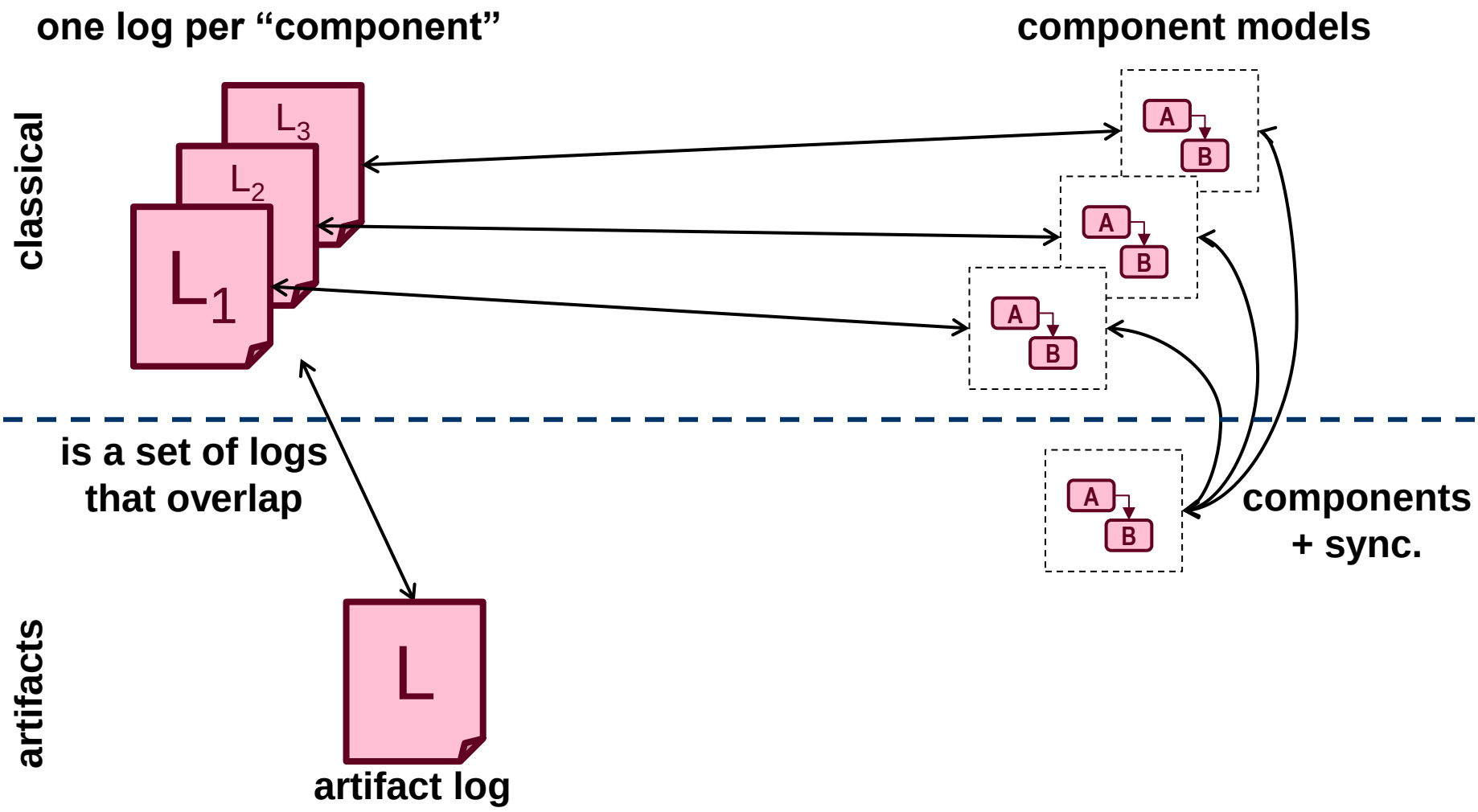


Artifact logs and models (Synchronous)

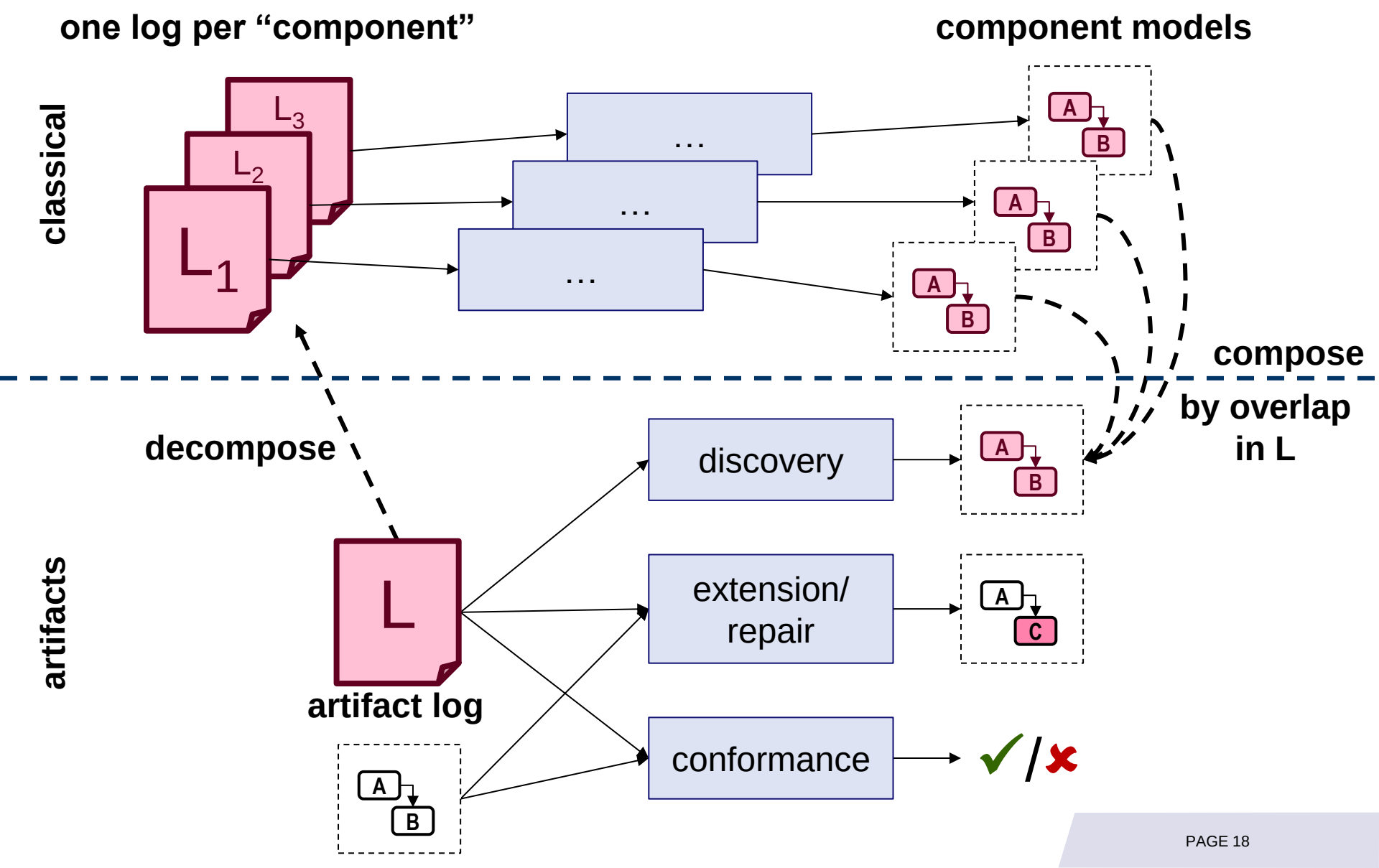


- cases ⇔ life-cycles
- artifacts synchronize on **shared actions**

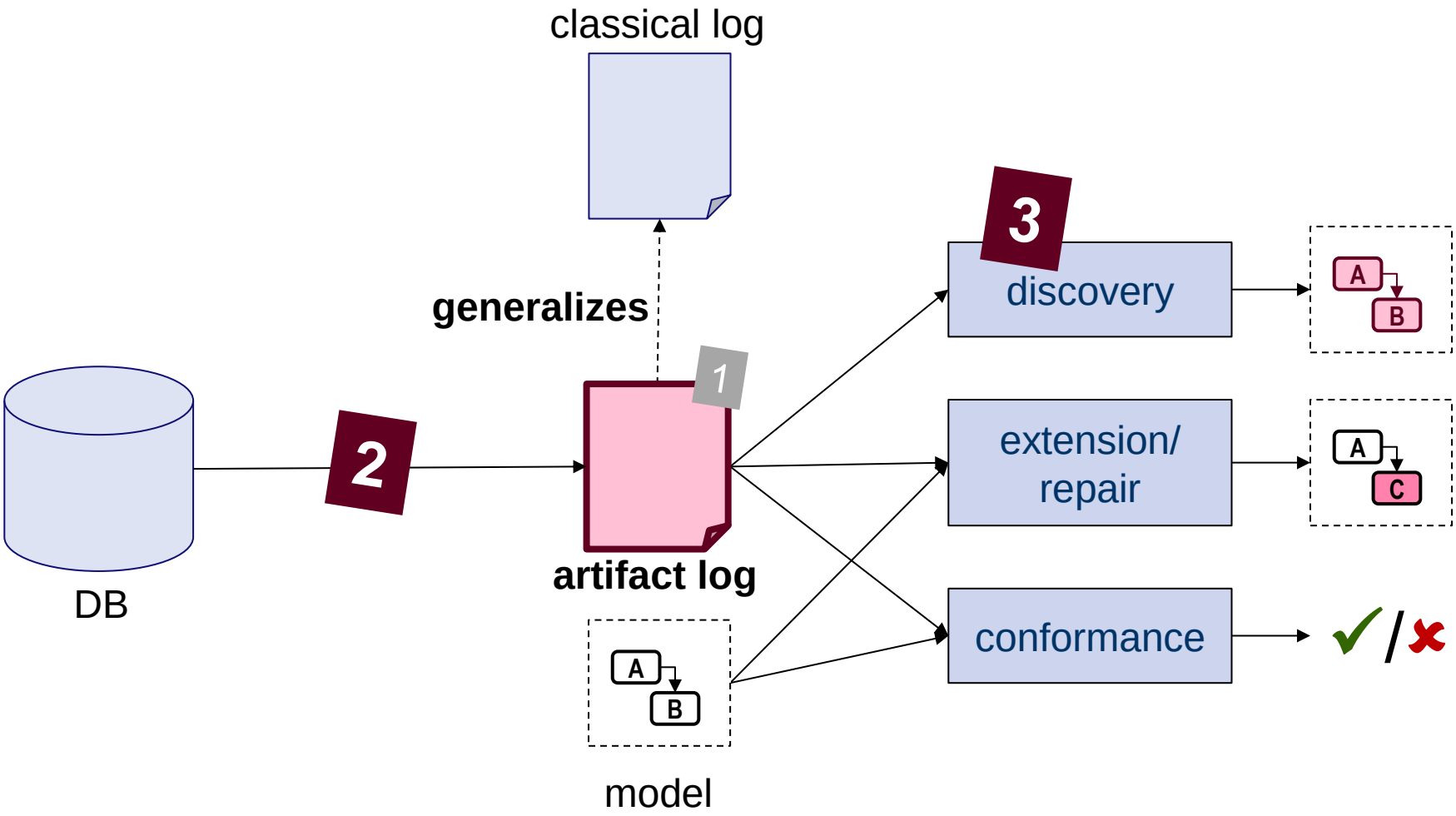
Compositionality



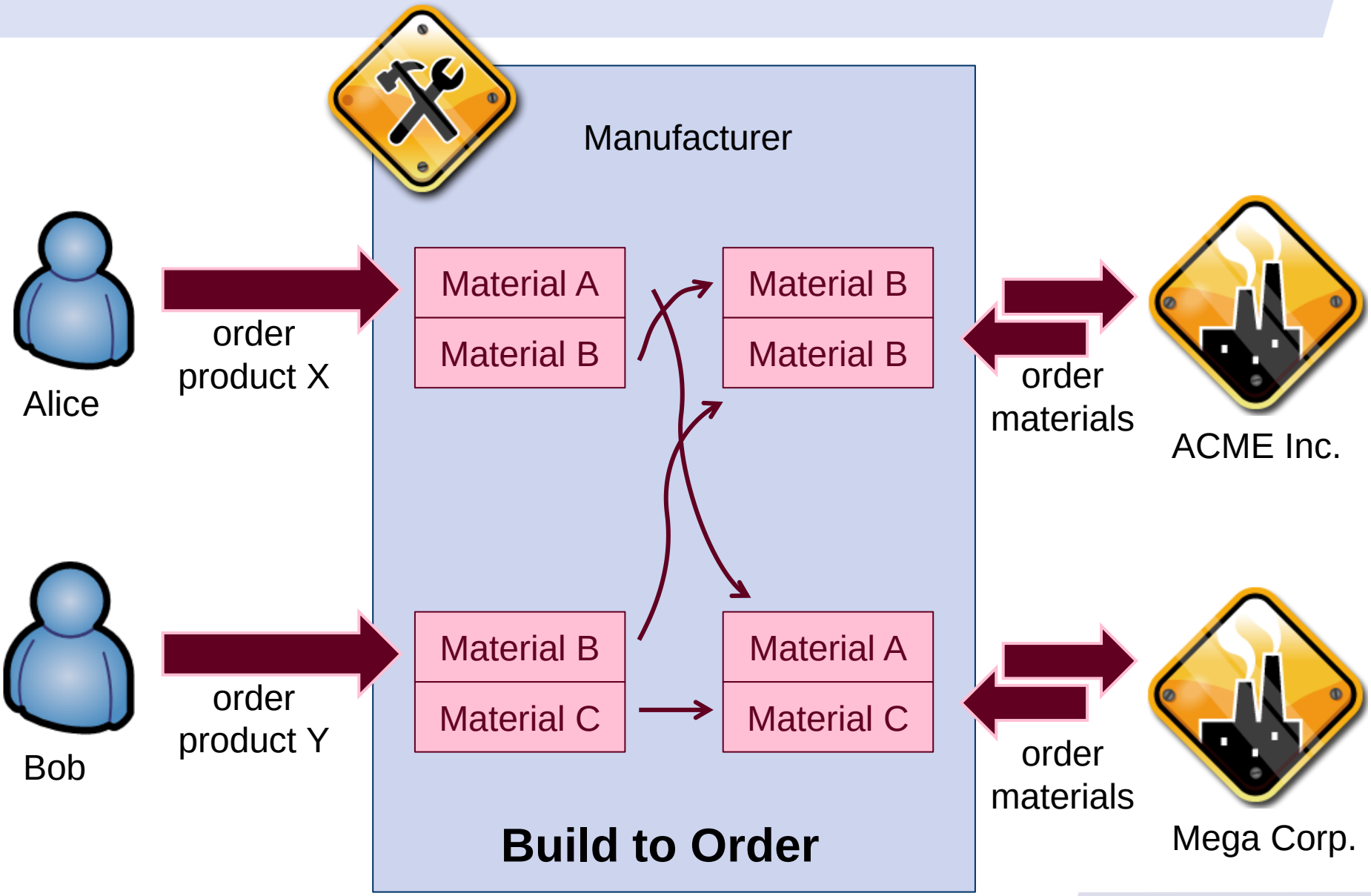
Mining for Artifacts: reduction to classical



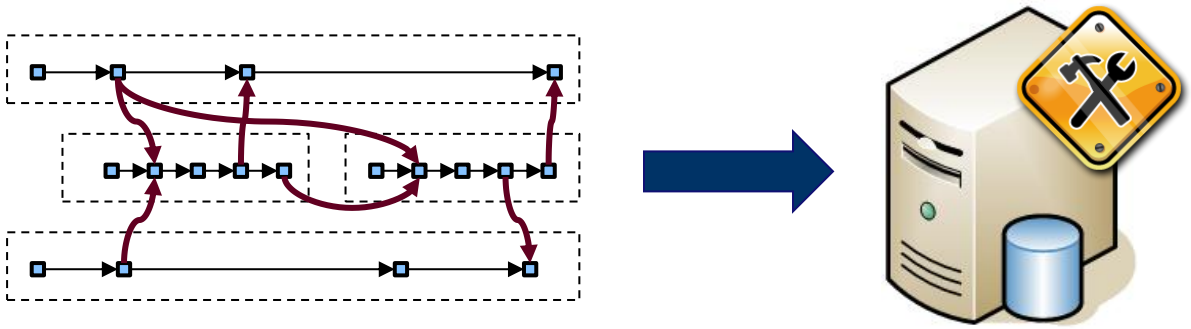
Process Mining for Artifacts



Typical Process in an ERP System



n-to-m relations → database



id attributes

time-stamp attributes

ProductOrder

poID	cust.	…	created	processed	built	shipped
po1	Alice		30-08 9:22	30-08 13:12	01-09 15:12	03-09 10:15
po2	Bob		30-08 10:15	30-08 13:14	01-09 16:13	03-09 17:18

Customer

cust.	address	…
Alice	…	…
Bob	…	…

data attributes

relations

OrderedMaterial

poID	moID	type	added
po1	mo3	B	30-08 13:13
po1	mo4	A	30-08 13:14
po2	mo3	B	30-08 13:15
po2	mo4	C	30-08 13:16

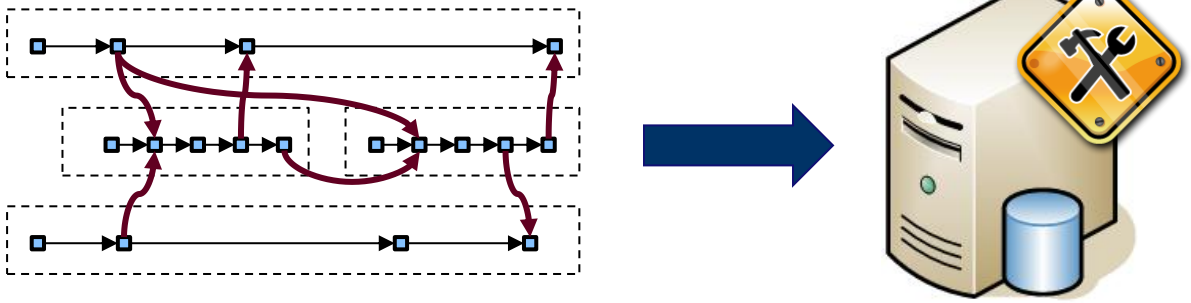
id attributes

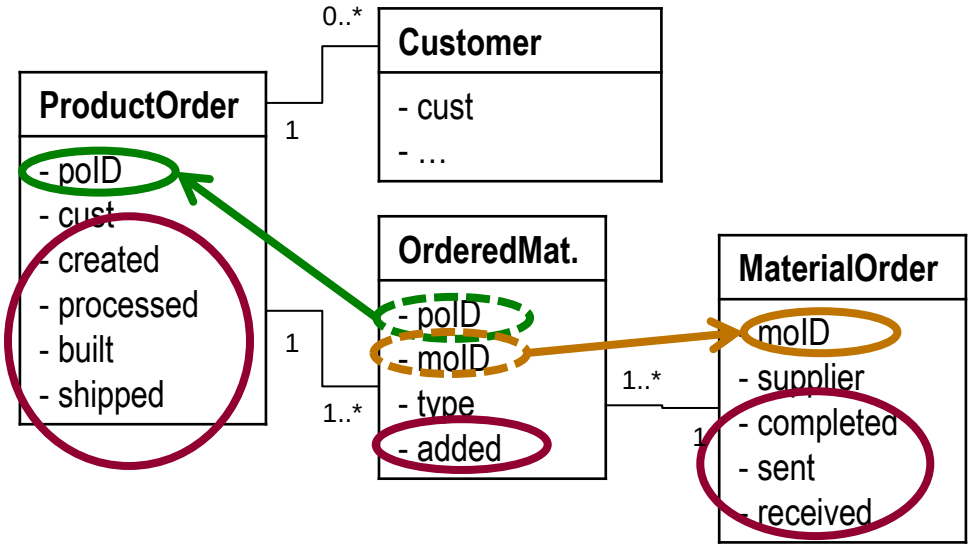
MaterialOrder

moID	suppl.	…	completed	sent	received
mo3	ACME		30-08 13:15	30-08 14:15	01-09 9:05
mo4	MEGA		30-08 13:17	30-08 16:12	01-09 10:13

relations

Extracting event logs from artifact systems

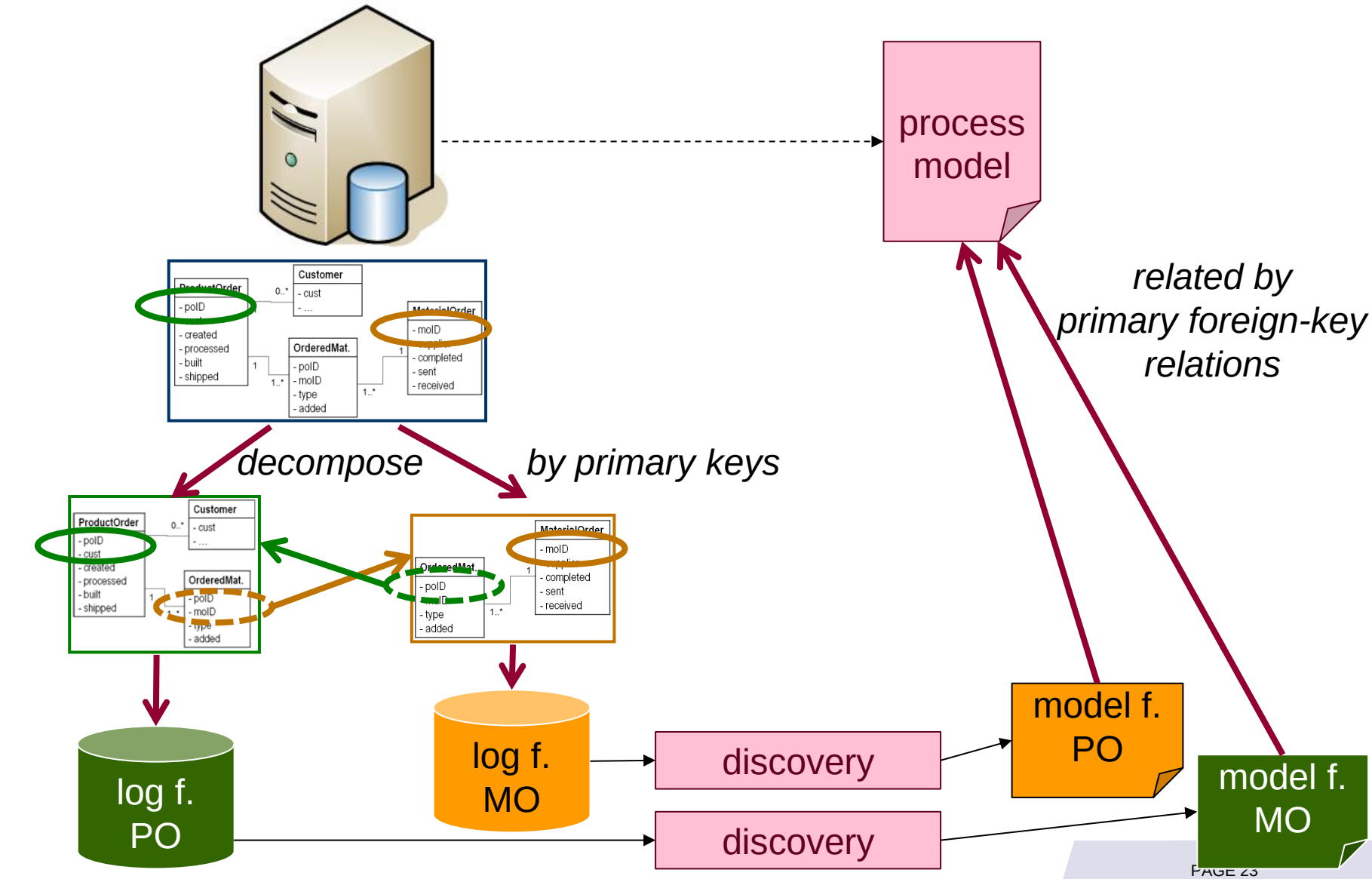




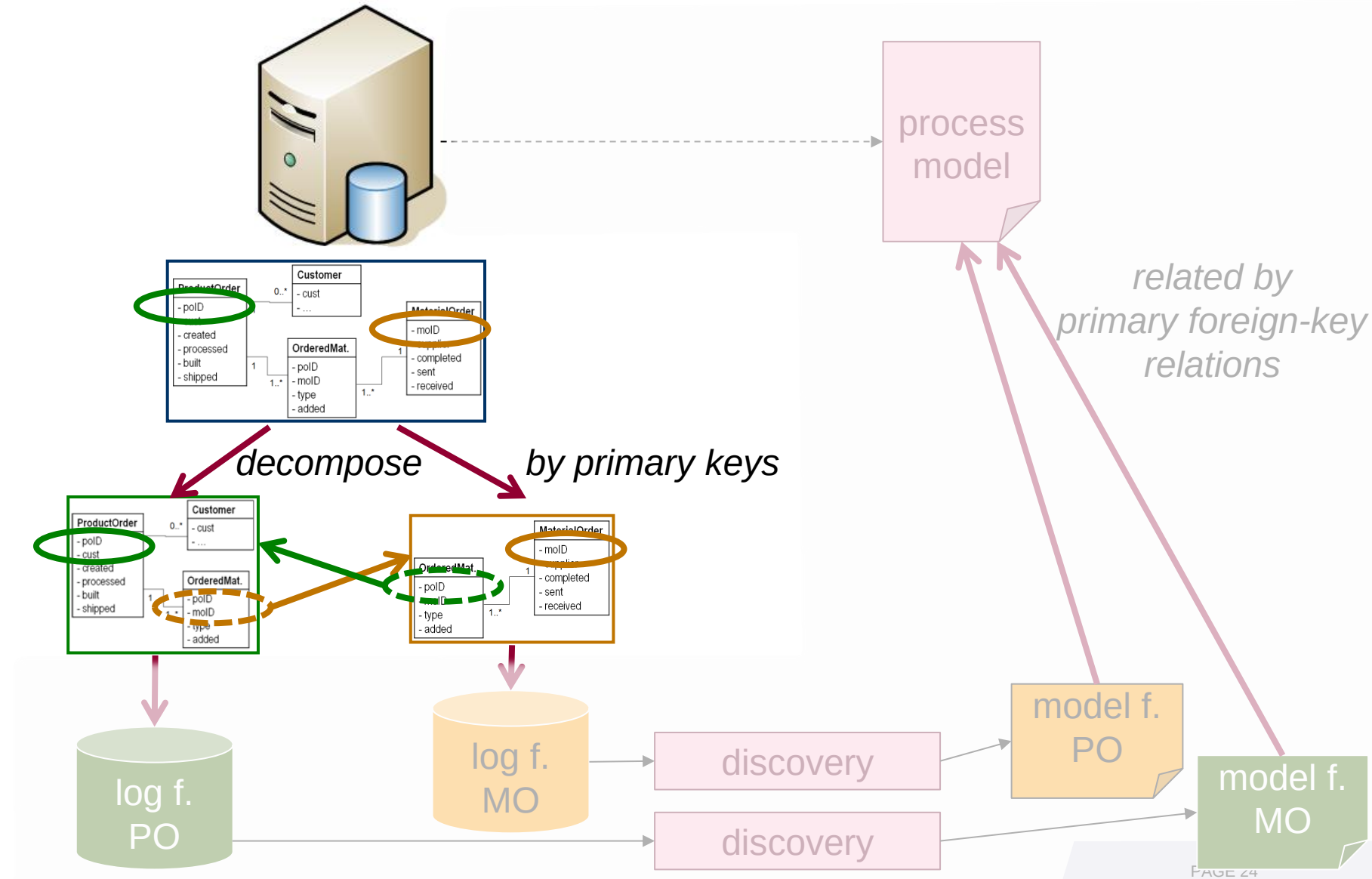
reality: data in a relational DB

- events stored as time-stamped attributes in tables
- multiple primary keys
  → **multiple notions of case**
- tables are related
  → **one event related to multiple cases**

Approach to log extraction & discovery

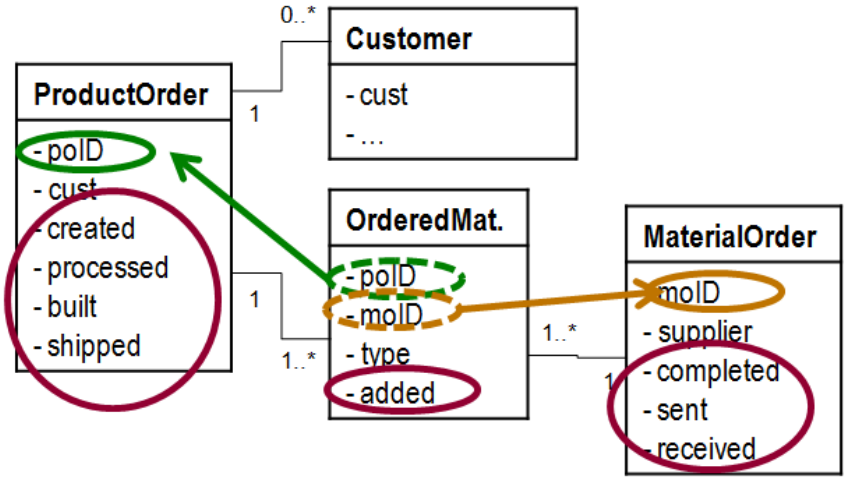


Find Artifact Schemas



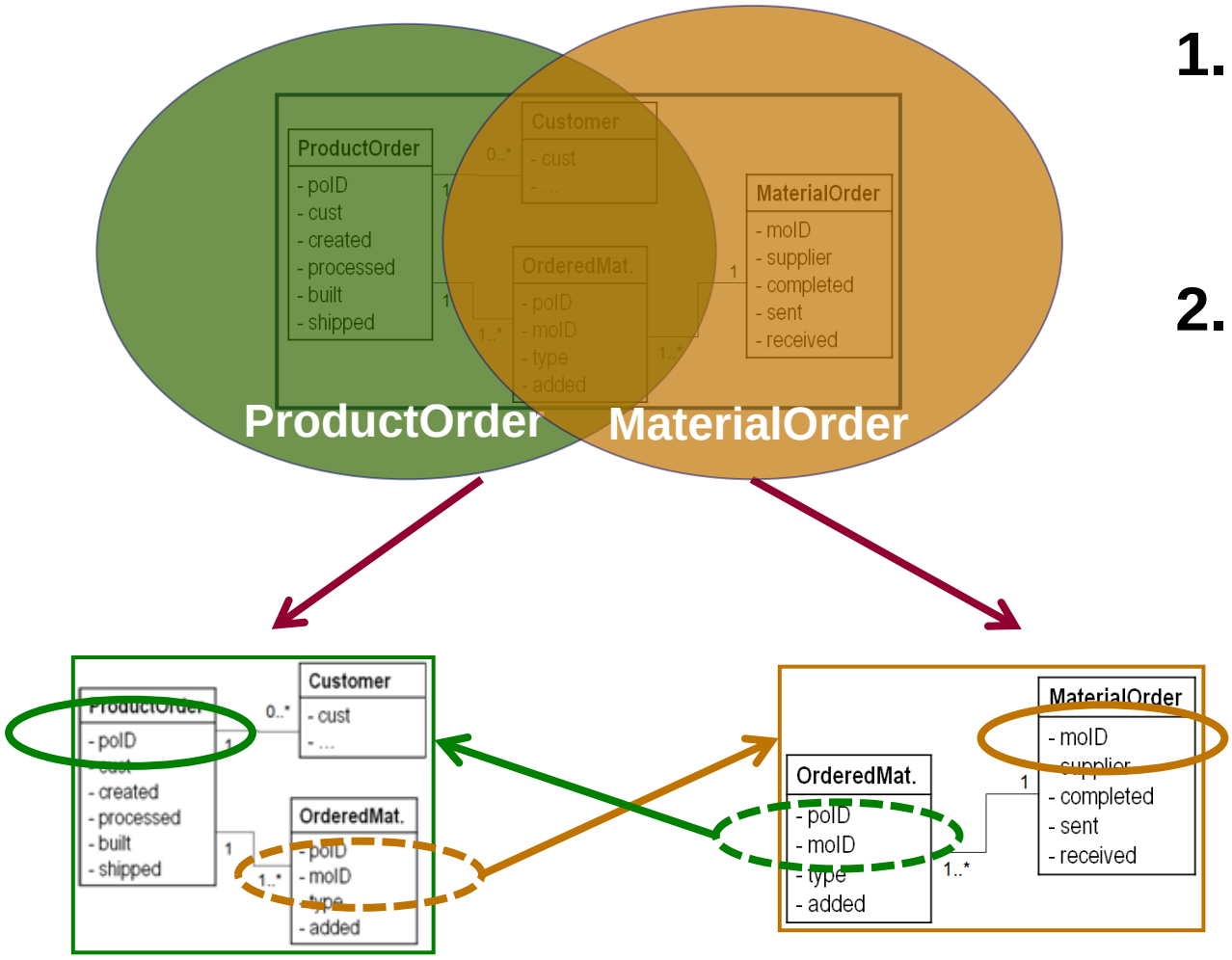
Step 0: discover database schema

- document schema vs. **actual** schema → identify
  - column types (esp. time-stamped columns)
  - primary keys
  - foreign keys
- various (non-trivial) techniques available
- key discovery is NP-complete in the size of the table(s)
- result:



Step 1: decompose schema into processes

= schema summarization



find:

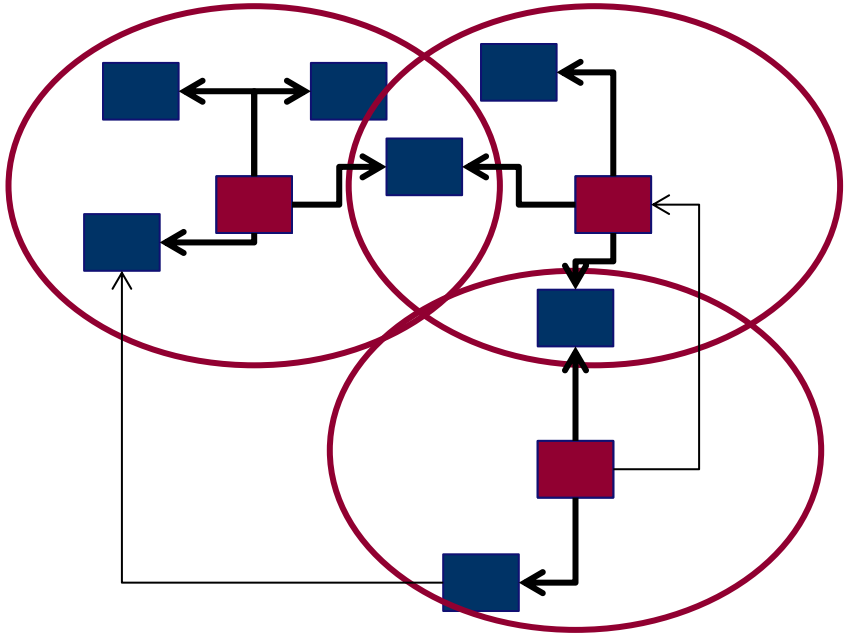
1. sets of corresponding tables

2. links between those

Automatic Schema Summarization

= group similar tables through clustering

- define a distance between any 2 tables
  - by relations
  - by information content



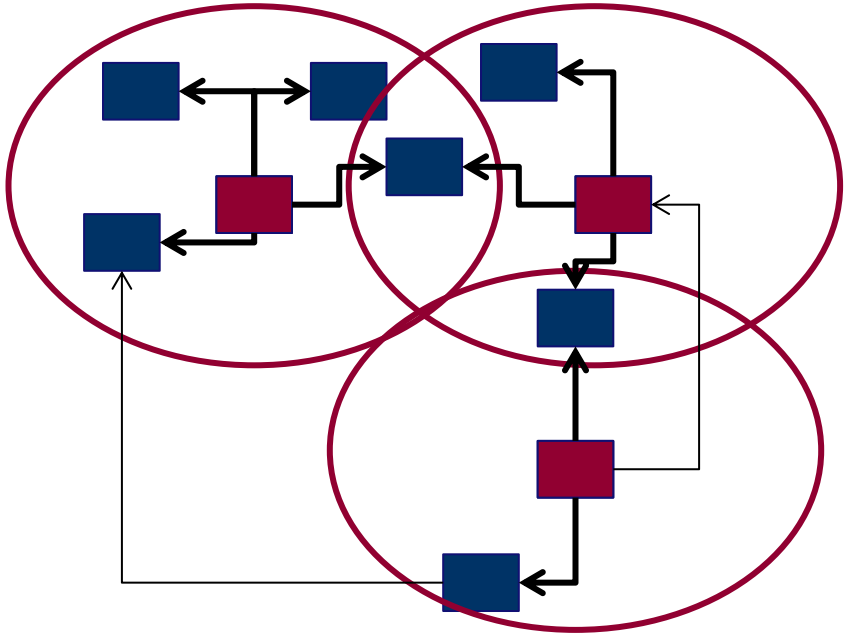
- tables that are close to each other
  → same cluster
- # of clusters: user input

Grouping by Clustering

1. **structural distance**
   many references = close
2. **information distance**

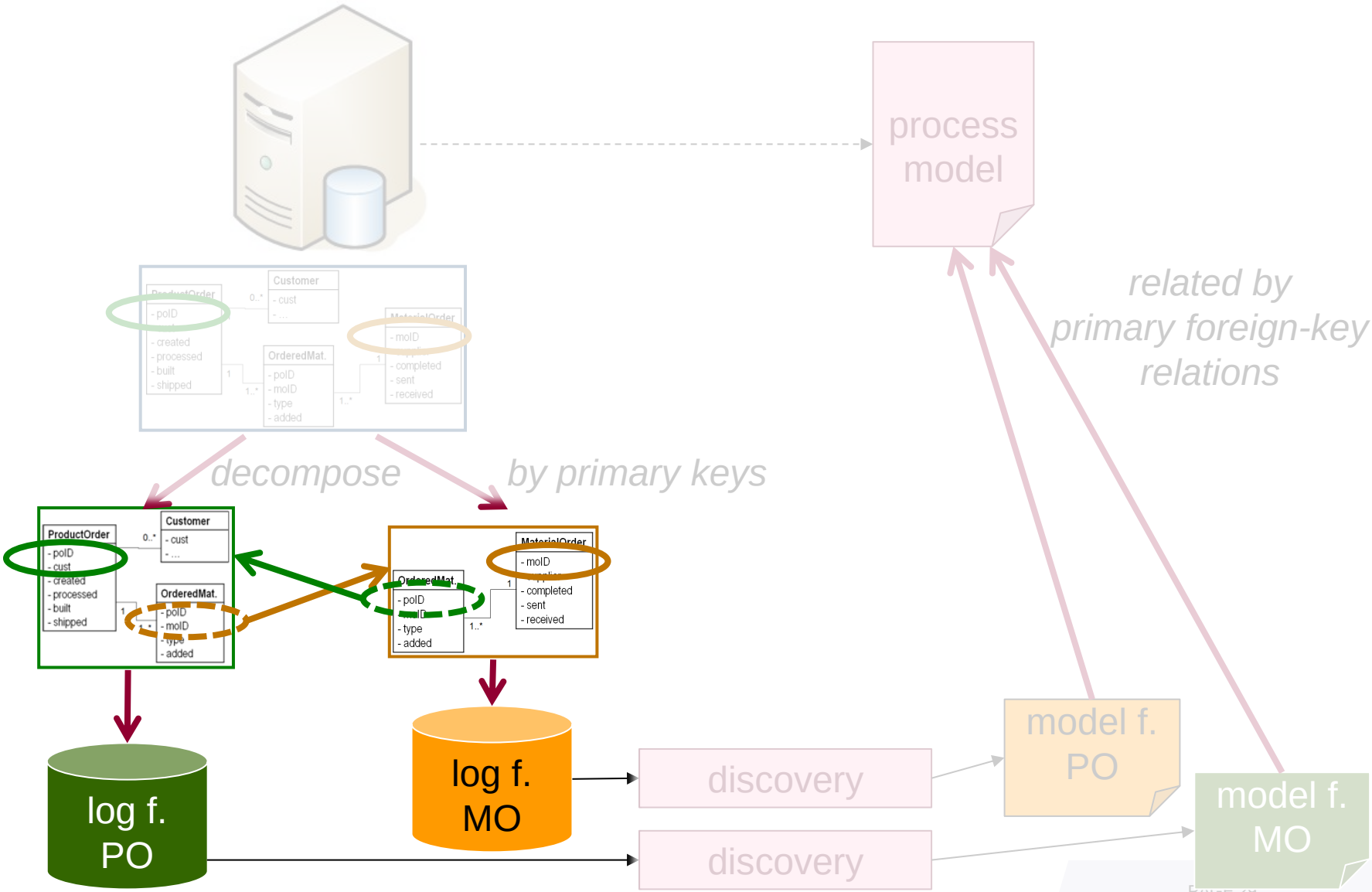
   importance of each table
   = **entropy** (is maximal if all records are different)

   distance: 2 tables with high entropies → large distance
3. **weighted distance**
   by structure + information
4. **k-means clustering**:
   k clusters based on weighted distance

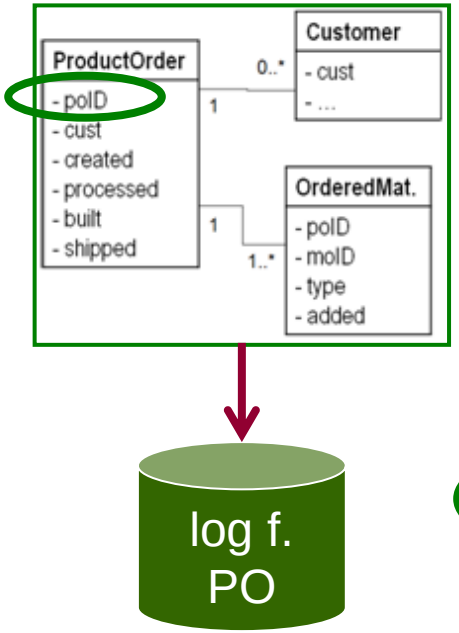


most important table of cluster
= table with least distance to all
→ key attribute of the cluster

Artifact Schema → Artifact Log



Log Extraction



cluster = set of related tables
+ primary key of most important table

case id

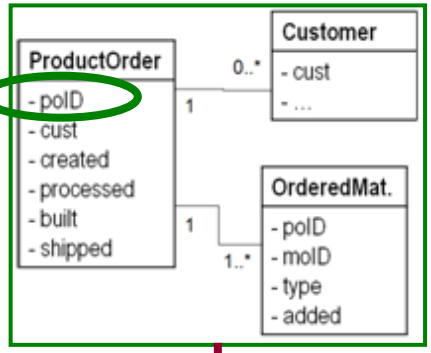
poID	cust.	…	created	processed	built	shipped
po1	Alice		30-08 9:22	30-08 13:12	01-09 15:12	03-09 10:15
po2	Bob		30-08 10:15	30-08 13:14	01-09 16:13	03-09 17:18

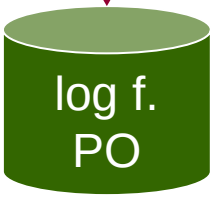
poID	moID	type	added
po1	mo3	B	30-08 13:13
po1	mo4	A	30-08 13:14
po2	mo3	B	30-08 13:15
po2	mo4	C	30-08 13:16

po1:

po2:

Log Extraction





cluster = set of related tables
+ primary key of most important table

case id

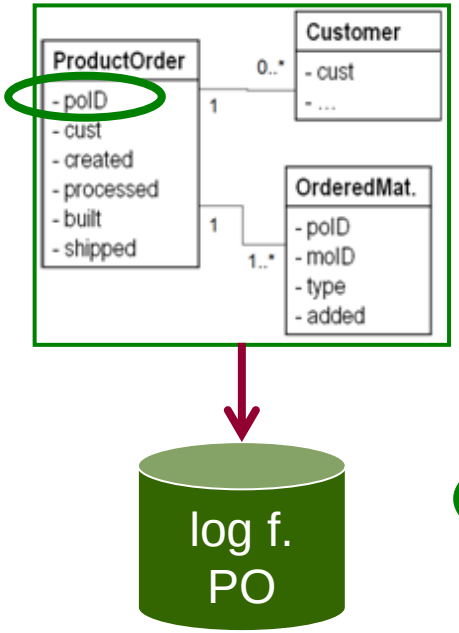
time-stamped attribute → event

poID	cust.	…	created	processed	built	shipped
po1	Alice		30-08 9:22	30-08 13:12	01-09 15:12	03-09 10:15
po2	Bob		30-08 10:15	30-08 13:14	01-09 16:13	03-09 17:18

poID	moID	type	added
po1	mo3	B	30-08 13:13
po1	mo4	A	30-08 13:14
po2	mo3	B	30-08 13:15
po2	mo4	C	30-08 13:16

po1:(**created**, poID=po1, time=30-08 9:22, …)

Log Extraction



cluster = set of related tables
+ primary key of most important table

case id

time-stamped attribute → event

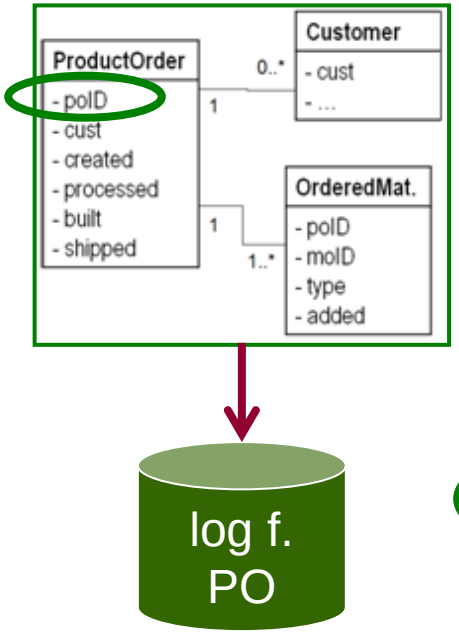
related attributes → event attributes

poID	cust.	…	created	processed	built	shipped
po1	Alice		30-08 9:22	30-08 13:12	01-09 15:12	03-09 10:15
po2	Bob		30-08 10:15	30-08 13:14	01-09 16:13	03-09 17:18

poID	moID	type	added
po1	mo3	B	30-08 13:13
po1	mo4	A	30-08 13:14
po2	mo3	B	30-08 13:15
po2	mo4	C	30-08 13:16

po1:(**created**, poID=po1, time=30-08 9:22, cust.=Alice, …

Log Extraction



cluster = set of related tables
+ primary key of most important table

case id

time-stamped attribute → event

related attributes → event attributes

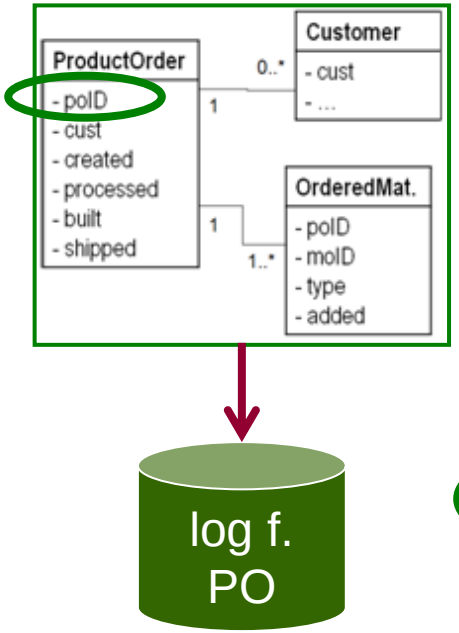
poID	cust.	…	created	processed	built	shipped
po1	Alice		30-08 9:22	30-08 13:12	01-09 15:12	03-09 10:15
po2	Bob		30-08 10:15	30-08 13:14	01-09 16:13	03-09 17:18

poID	moID	type	added
po1	mo3	B	30-08 13:13
po1	mo4	A	30-08 13:14
po2	mo3	B	30-08 13:15
po2	mo4	C	30-08 13:16

po1:(**created**, poID=po1, time=30-08 9:22, cust.=Alice, …

(**processed**, poID=po1, time=30-08 13:12, …)

Log Extraction



cluster = set of related tables
+ primary key of most important table

case id

time-stamped attribute → event

related attributes → event attributes

poID	cust.	…	created	processed	built	shipped
po1	Alice		30-08 9:22	30-08 13:12	01-09 15:12	03-09 10:15
po2	Bob		30-08 10:15	30-08 13:14	01-09 16:13	03-09 17:18

poID	moID	type	added
po1	mo3	B	30-08 13:13
po1	mo4	A	30-08 13:14
po2	mo3	B	30-08 13:15
po2	mo4	C	30-08 13:16

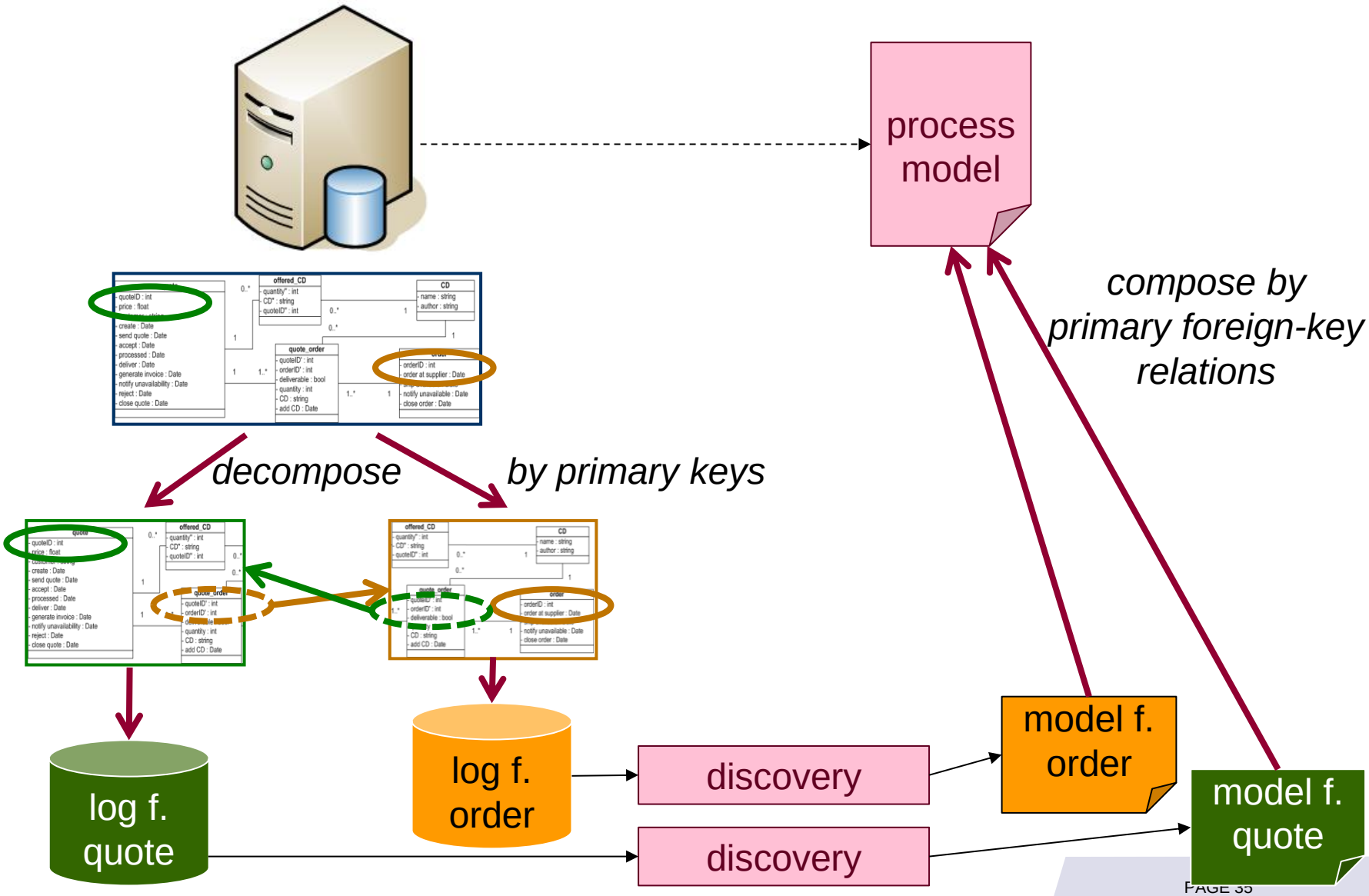
po1:(**created**, poID=po1, time=30-08 9:22, cust.=Alice, …)

(**processed**, poID=po1, time=30-08 13:12, …)

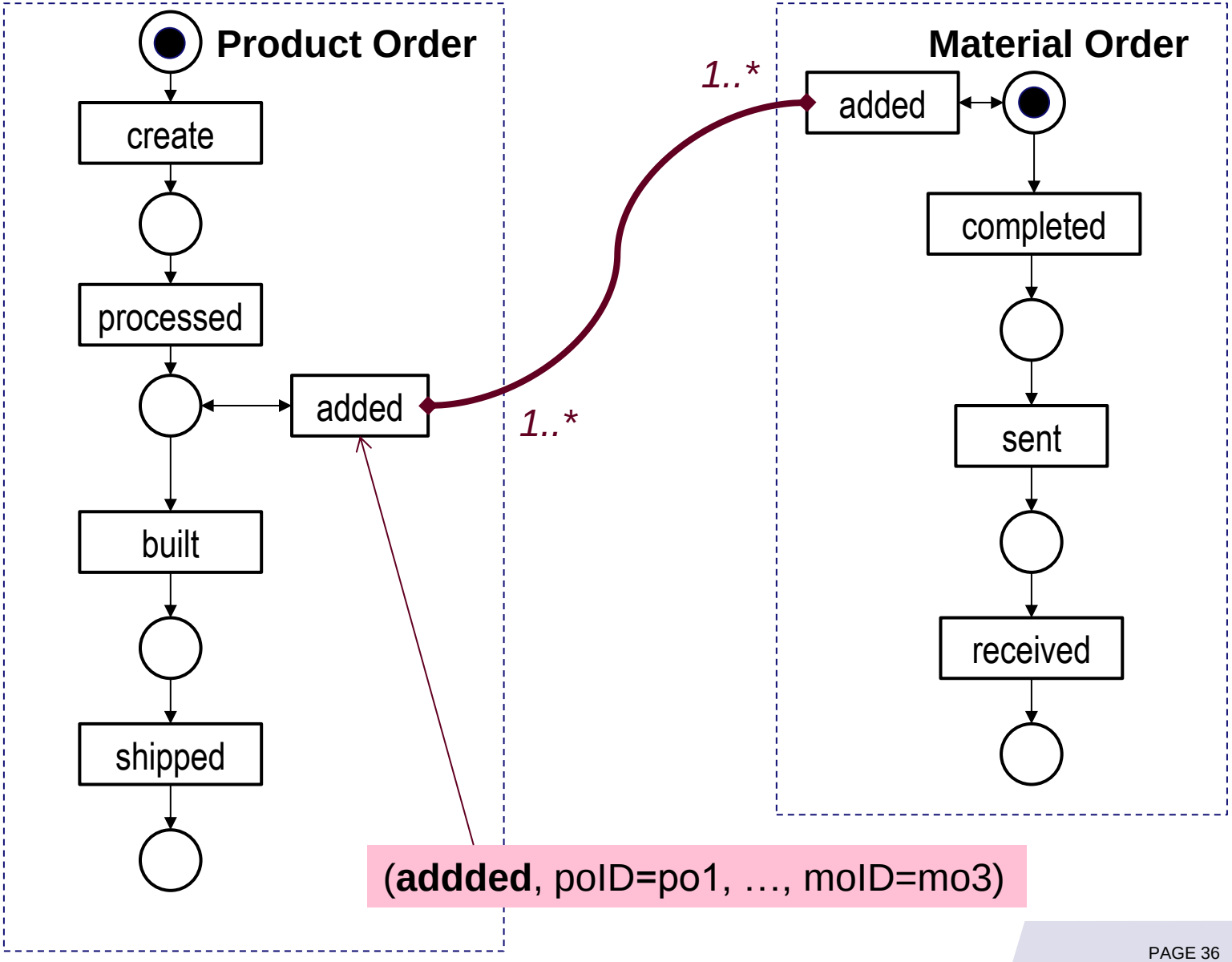
(**added**, poID=po1, time=30-08 13:13, moID=mo3, …)

refers to artifact "MaterialOrder"

Outline


process model
compose by primary foreign-key relations
decompose
by primary keys
log f. quote
log f. order
discovery
discovery
model f. order
model f. quote

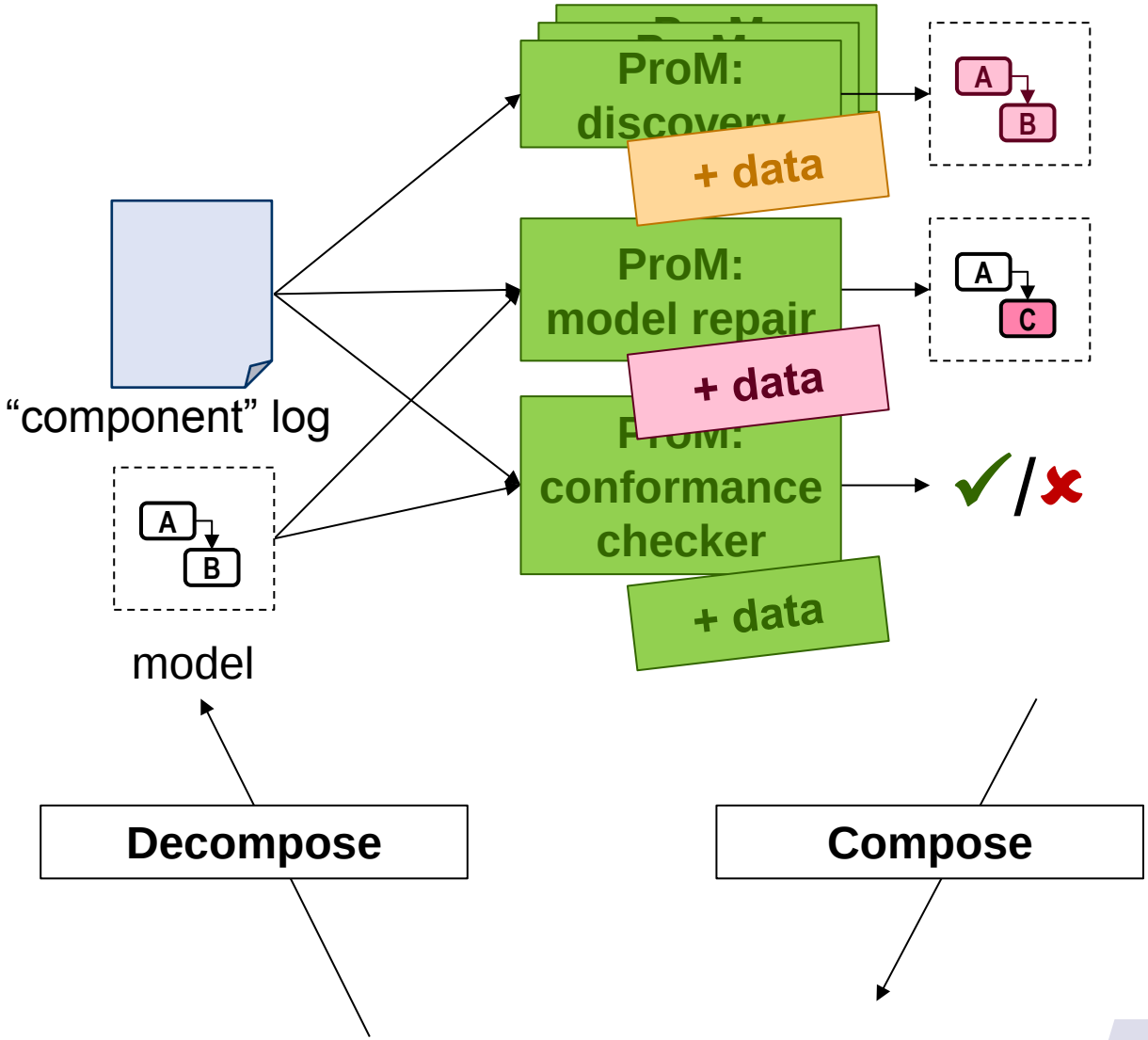
Resulting Model(s)



Open issues

- performance
  - key discovery: NP-complete in R (# of records)
  - → sampling of data, domain knowledge, semi-automatic
- requires good database structure
  - proper relations, proper keys
  - otherwise wrong clusters are formed
  - events don’t get right attributes
  - → semi-automatic approach
- events shared by multiple cases
  - just a reference between two tables does not tell an event participated was shared by two cases → working on it

Summary: Mining for Artifact Lifecycles



Summary: Mining for Artifacts