



POLITECNICO
MILANO 1863

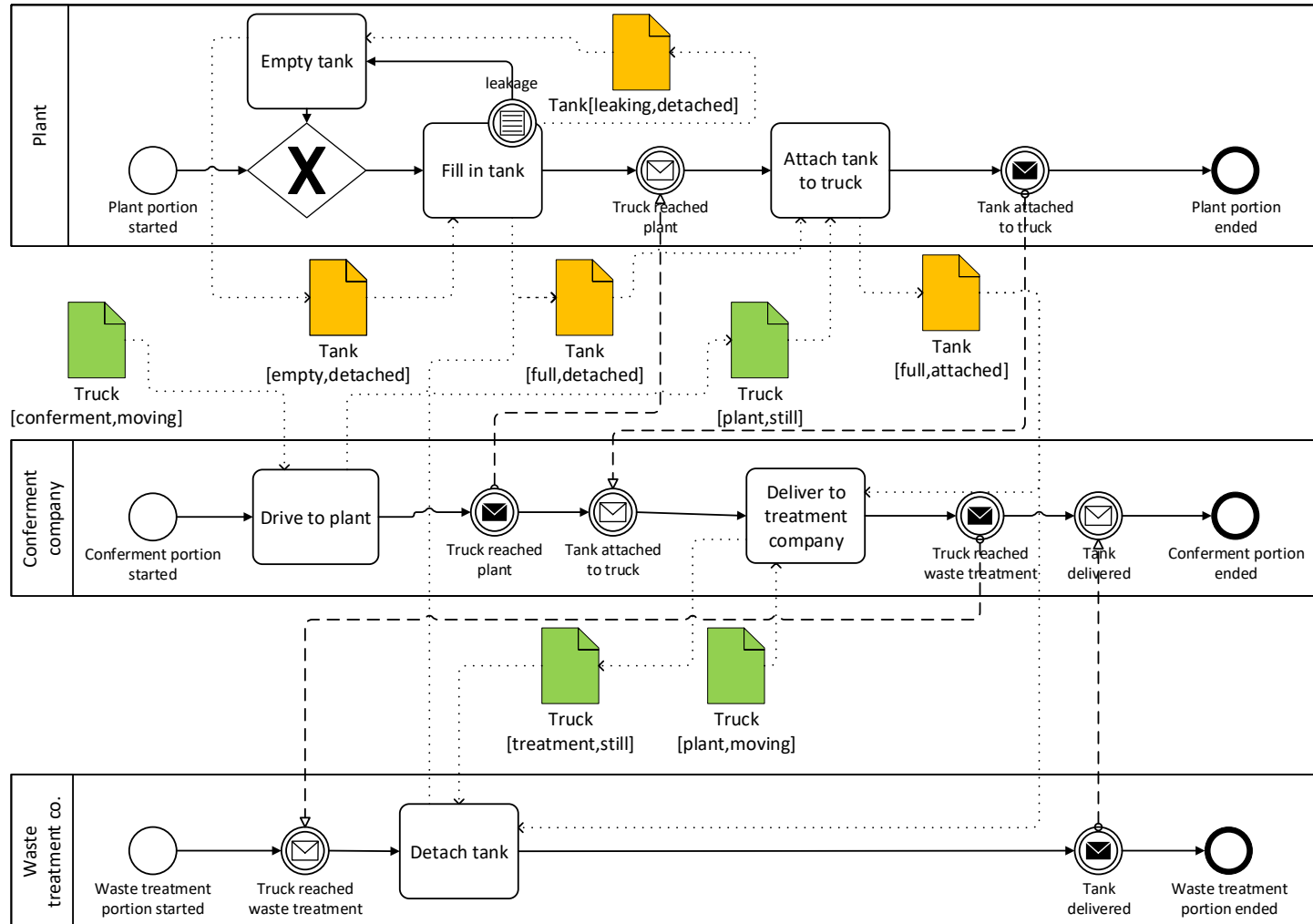
Trusted Artifact-driven Process Monitoring of Multi-Party Business Processes with Blockchain

Giovanni Meroni, Pierluigi Plebani, Francesco Vona

BPM 2019 Blockchain Forum – Vienna, September 3, 2019

Motivating example

Hazardous waste disposal



Motivating example

Hazardous waste disposal

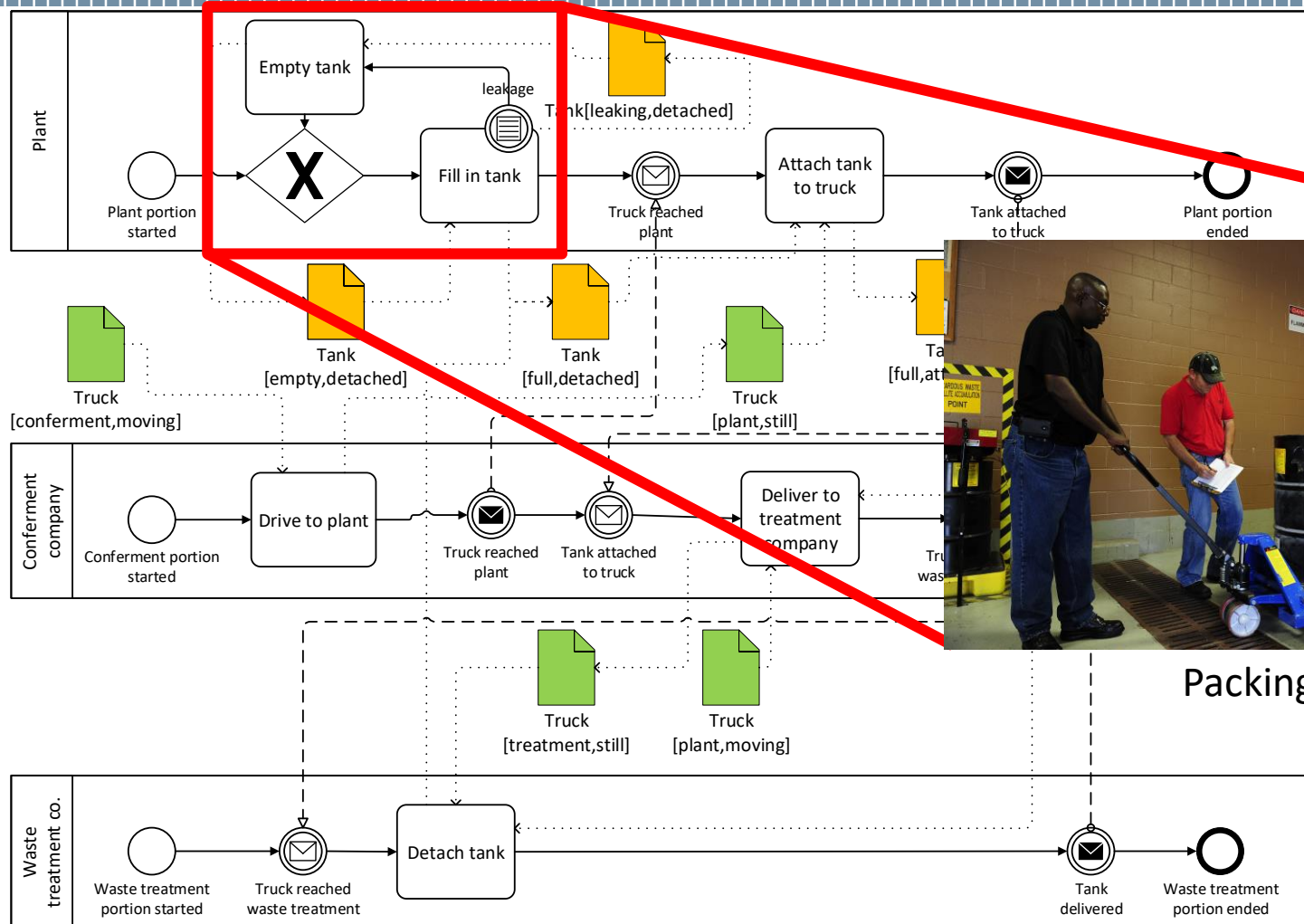
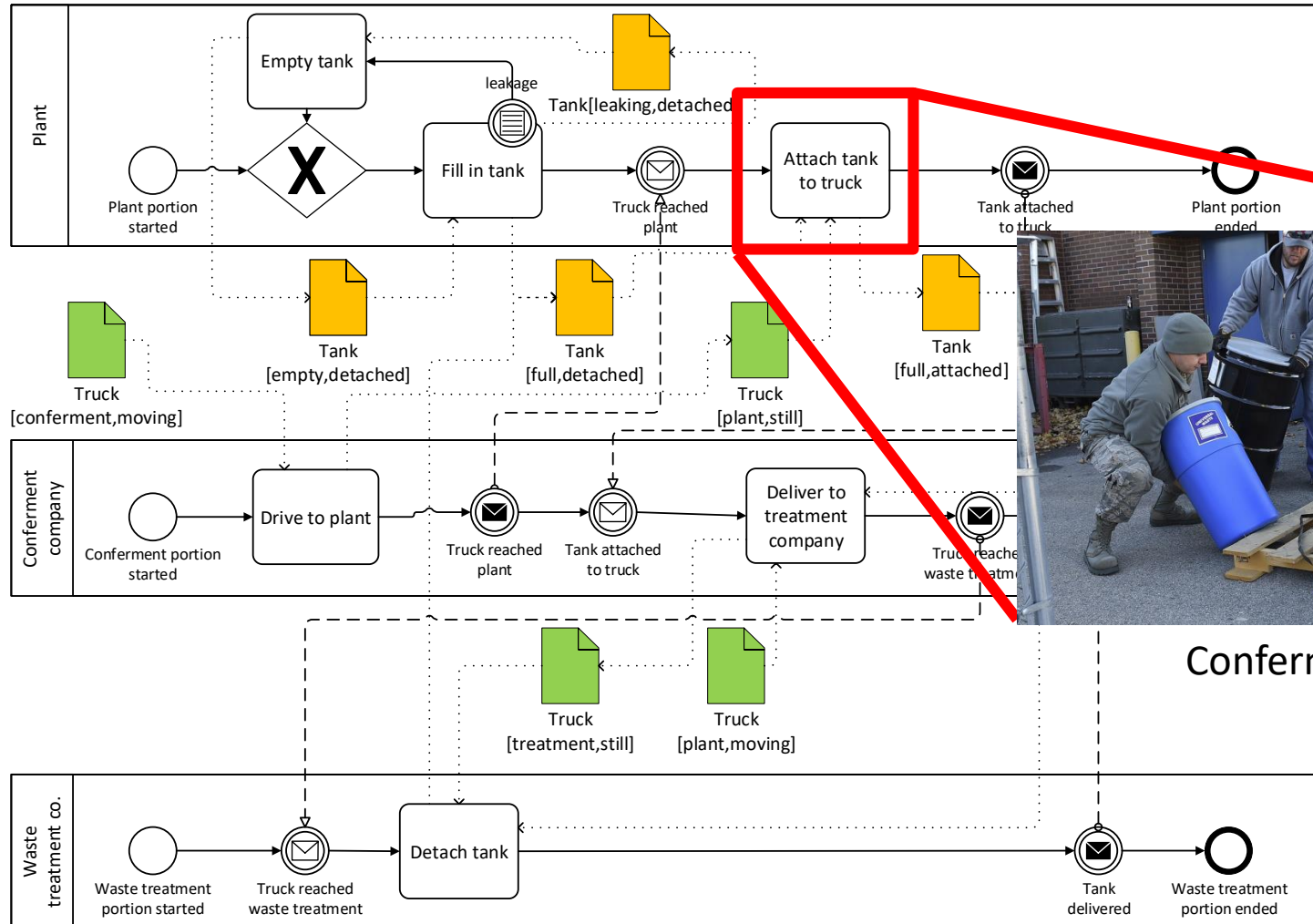


Image source: <https://www.shaw.af.mil/News/Photos/igphoto/2000108024/>

Motivating example

Hazardous waste disposal



Conferment

Image source: <https://www.dvidshub.net/image/2343067/environmental-office-ships-hazardous-waste-protects-environment>

Motivating example

Hazardous waste disposal

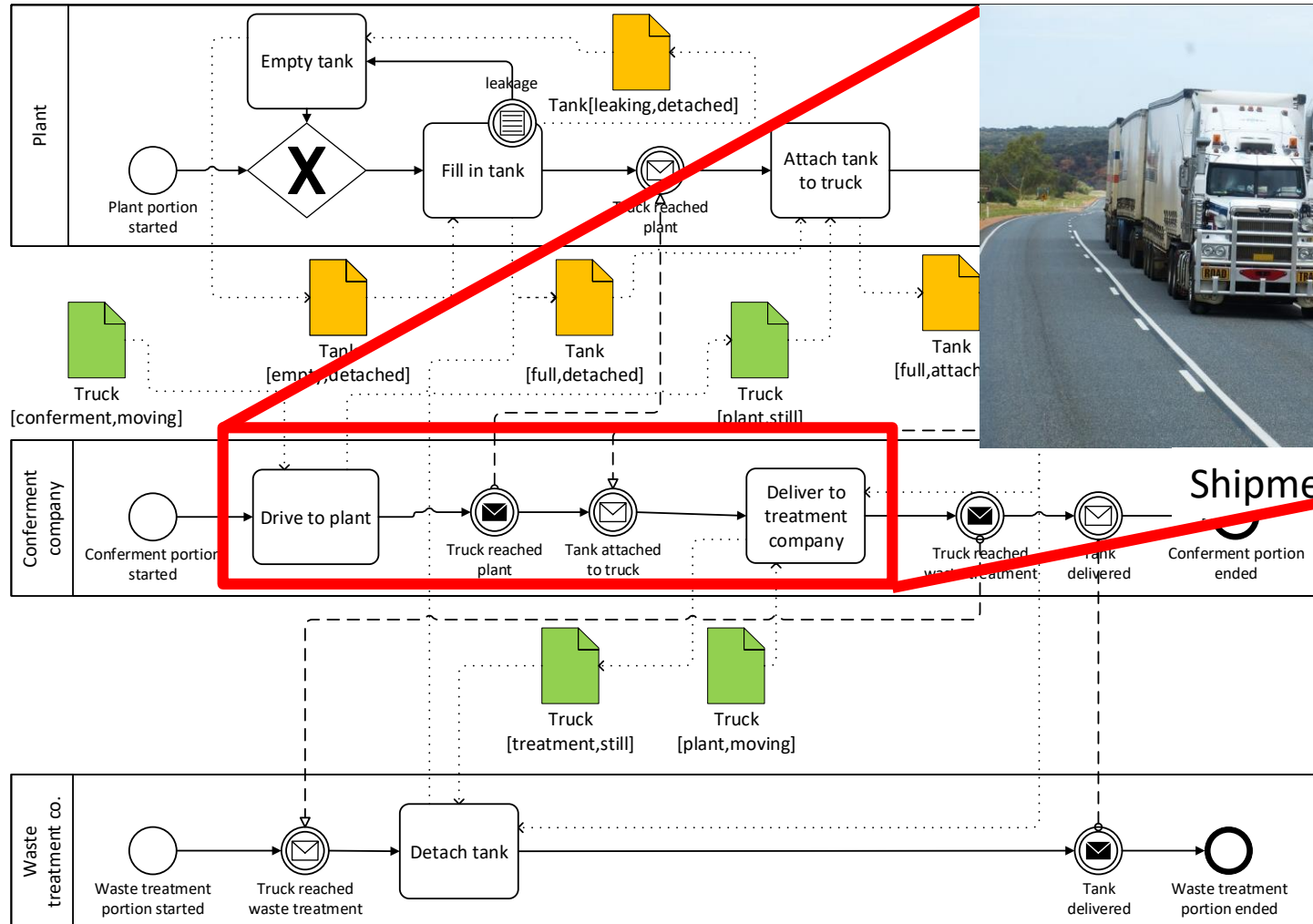


Image source: <https://picryl.com/media/semi-trailers-truck-road-transportation-traffic-17c822>

Motivating example

Hazardous waste disposal

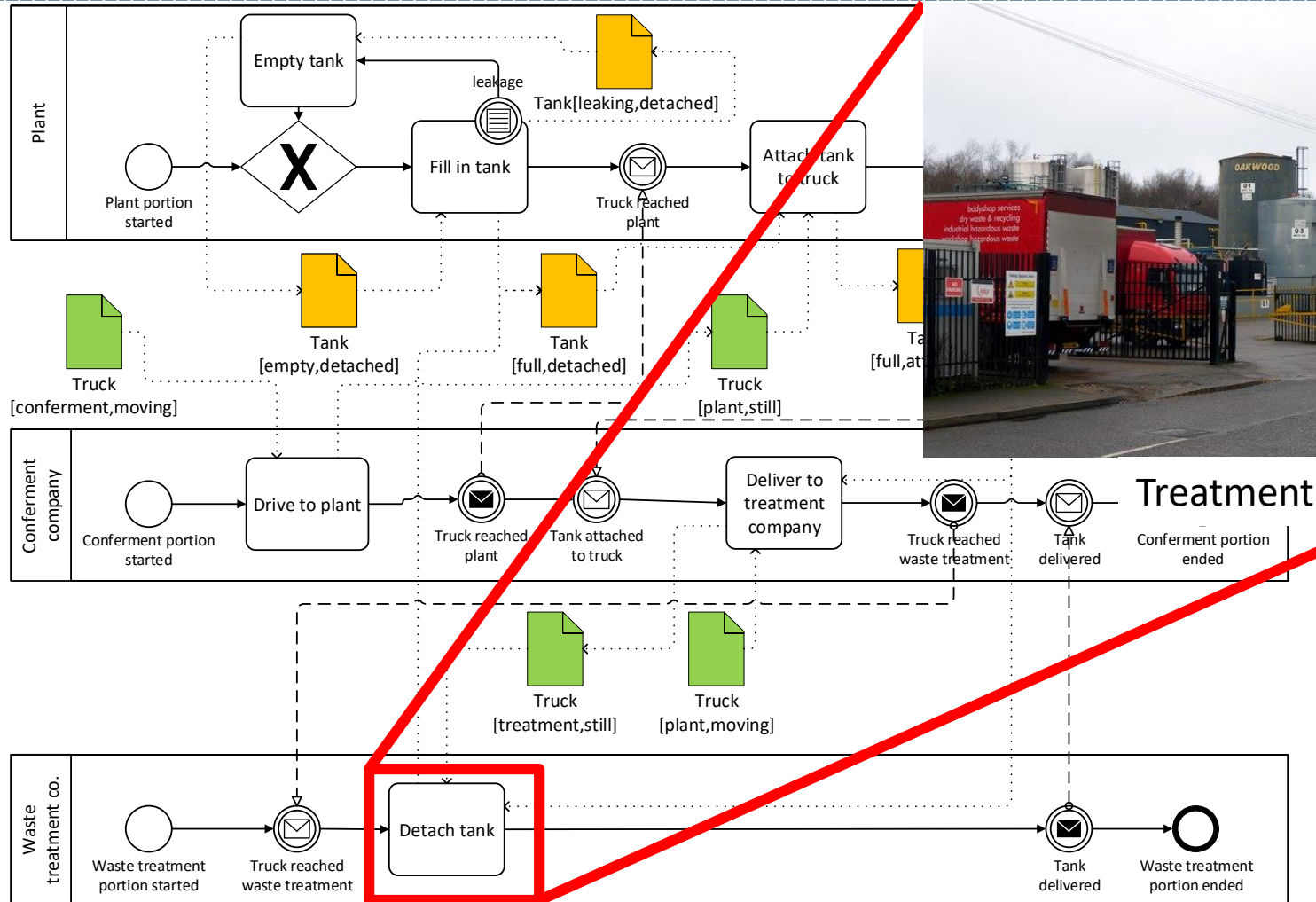


Image source: [cc-by-sa/2.0](https://creativecommons.org/licenses/by-sa/2.0/) - © [Graham Hogg](https://geograph.org.uk/p/4785641) - geograph.org.uk/p/4785641

© 2019 G. Meroni, P. Plebani, F. Vona

POLITECNICO MILANO 1863

Motivating example

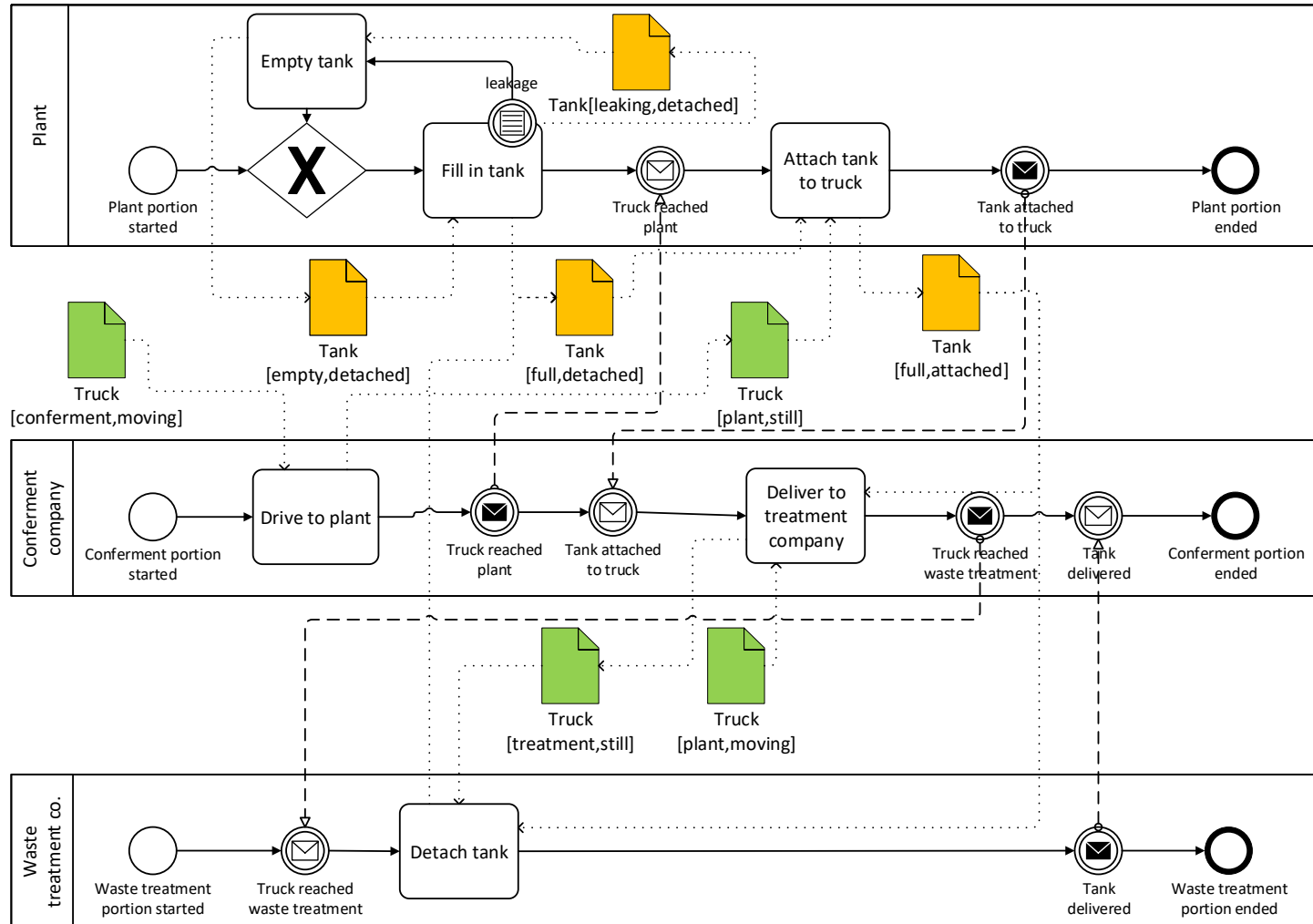
Hazardous waste disposal



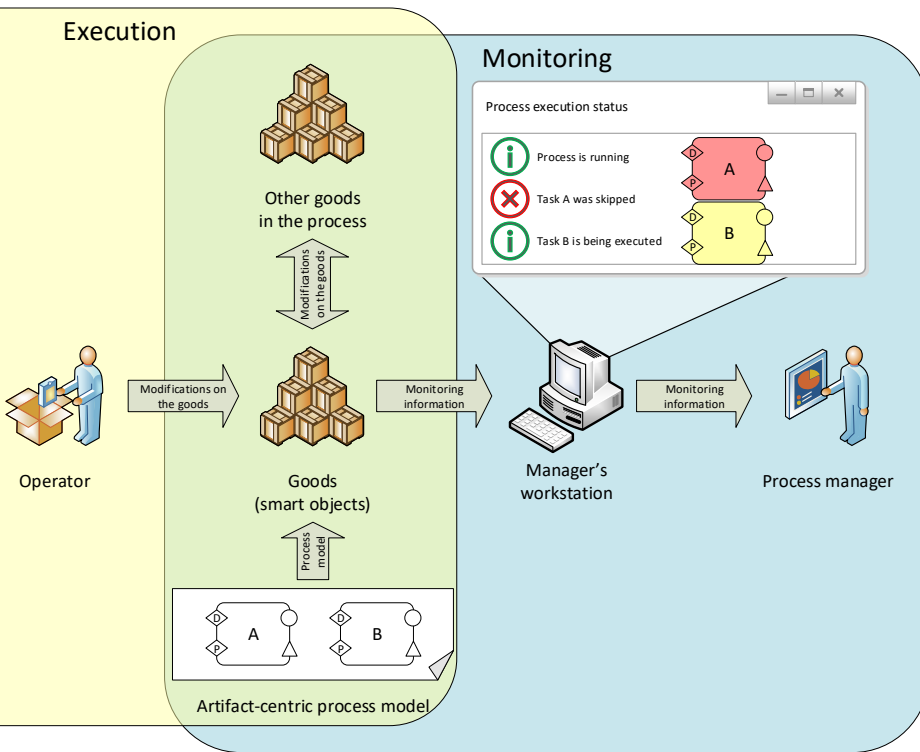
Image source: https://de.wikipedia.org/wiki/Datei:Umweltbelastung_durch_Industrie%C3%B6le_in_Industriebranche_2.jpg

Motivating example

Hazardous waste disposal



Trusted artifact-driven process monitoring



- Executing an activity changes one or more artifacts
 - If we monitor artifacts, we know when activities are run
- Key idea: Exploit the Internet of Things to make artifacts aware of:
 - Their current conditions
 - Process model
- Perform monitoring transparently and autonomously

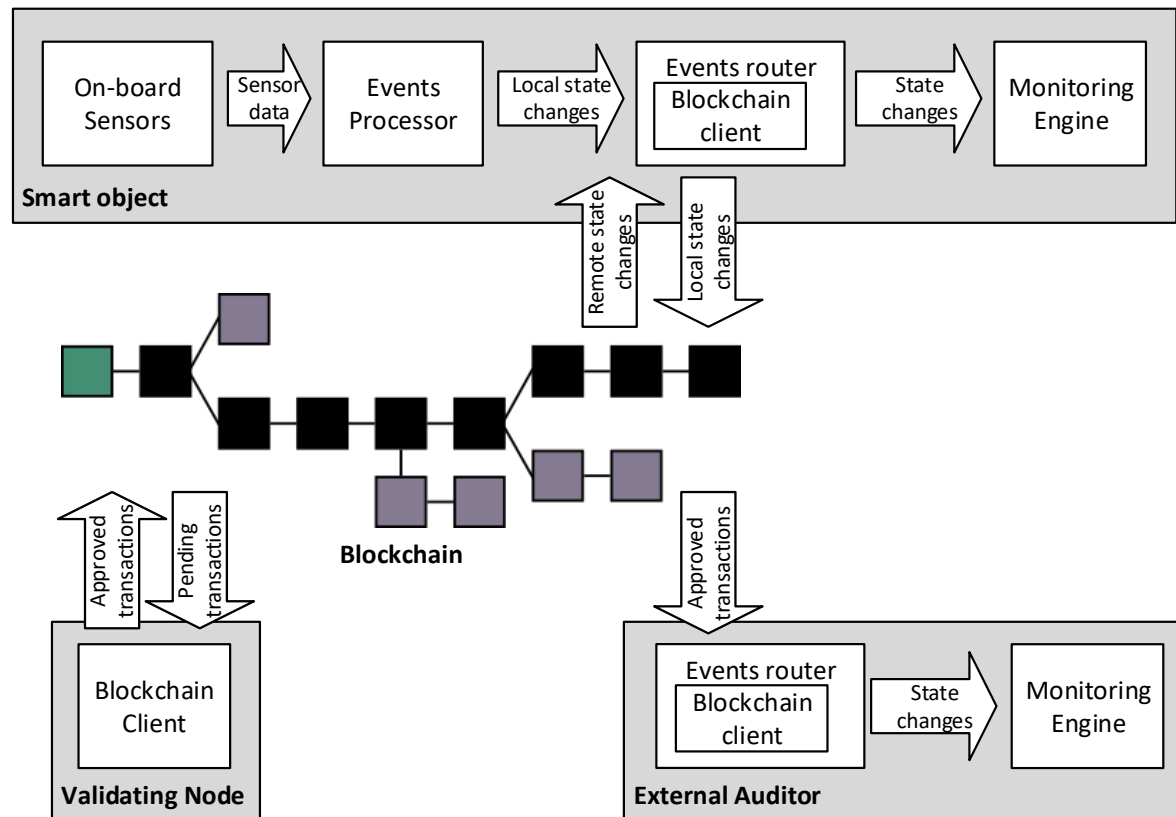
Trusted artifact-driven process monitoring

- Artifact-driven monitoring + Blockchain:
 - Use the blockchain to store changes in the artifacts
 - Includes sensor data
- Trusted notifications:
 - Sent by IoT devices, not human beings
 - Sensor data can be retrieved and audited
 - Known and certified originating artifact
 - Organizations cannot change notifications
- Two implemented platforms

Trusted artifact-driven process monitoring

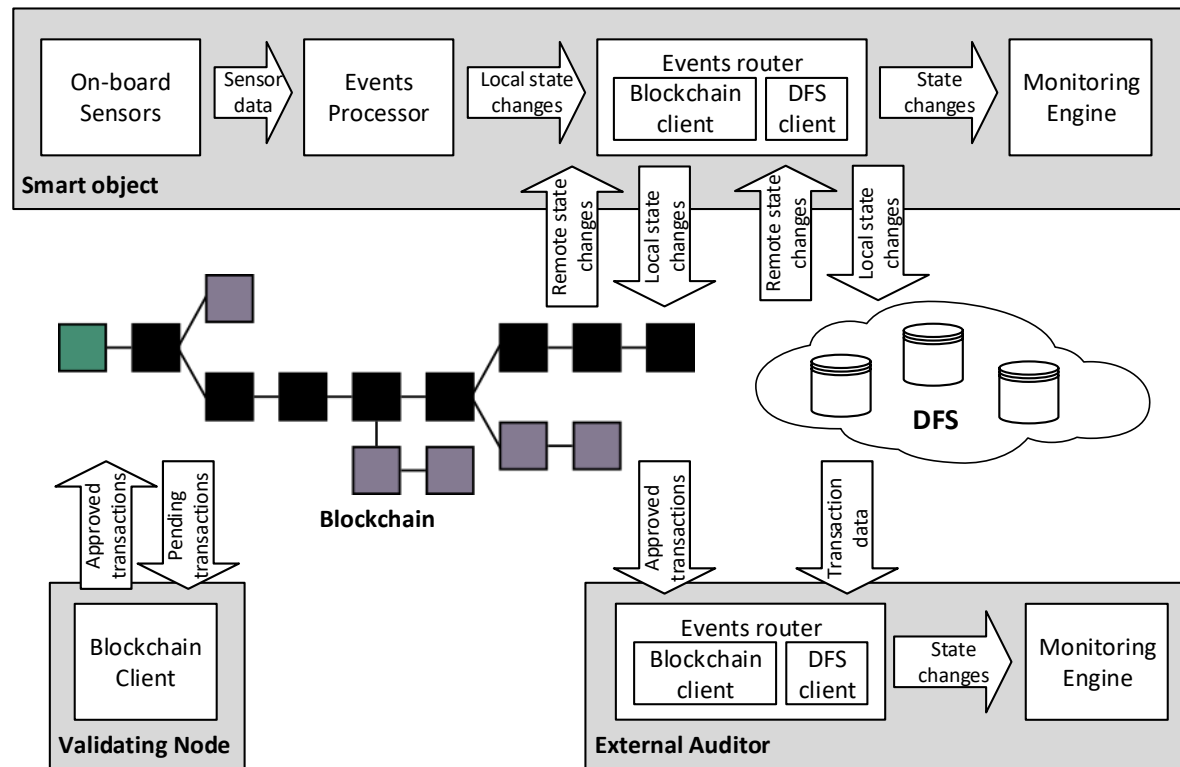
Fully blockchain-based monitoring platform

- Stores artifact conditions + sensor data on-chain
- All data are persistent
- Variable amount of data per transaction
- Not recommended for public blockchain



Trusted artifact-driven process monitoring DFS-blockchain hybrid monitoring platform

- Stores sensor data on distributed file system (e.g., IPFS)
- Stores artifact conditions + sensor data hash on-chain
- Fixed amount of data per transaction



Evaluation

Monitoring platforms comparison

<i>Process name</i>	<i>Executions per process</i>	<i>Median transactions per execution</i>	<i>Contract deployment cost (gas)</i>	<i>Median cost per transaction (gas)</i>	<i>Median cost per execution (gas)</i>
AMS-BRU	9	5.67	3276717	724547	4472261
AMS-CDG	8	8.88	3298198	724611	6846820
AMS-FRA	4	10.75	3277485	724529	8608058
AMS-LHR	12	10.58	3766963	724564	7979801
BRU-AMS	10	5.80	3298710	724609	4532603
CDG-AMS	10	11.00	3298710	724486	8299217

Results of the validation for the fully blockchain-based platform. **3.45 – 6.63 USD**

<i>Process name</i>	<i>Executions per process</i>	<i>Median transactions per execution</i>	<i>Contract deployment cost (gas)</i>	<i>Median cost per transaction (gas)</i>	<i>Median cost per execution (gas)</i>
AMS-BRU	9	5.67	1155343	116235	787424
AMS-CDG	8	8.88	1155343	116235	1176585
AMS-FRA	4	10.75	1155343	116235	1538362
AMS-LHR	12	10.58	1155343	116235	1326045
BRU-AMS	10	5.80	1155343	116235	789697
CDG-AMS	10	11.00	1155343	116235	1394119

Results of the validation for the DFS-blockchain hybrid platform. **0.60 – 1.19 USD**

Evaluation

Monitoring platforms comparison

Platform	Fully BC-based	DFS-BC hybrid
Decentralization	✓	✓
Auditability	✓	✓
Persistence	✓	✗
Gas per transaction	Variable	Fixed
Permissioned BC	✓	✓
Public BC	Not recommended	✓

Conclusion & future work

- Trusted artifact-driven monitoring solves the issues of:
 - Identifying when activities are executed
 - Avoiding fake or misleading information
- Future research directions:
 - Notifications and sensor data quality
 - Oracles
 - Sidechaining
 - Escrow mechanisms



POLITECNICO
MILANO 1863



Scan me to know more

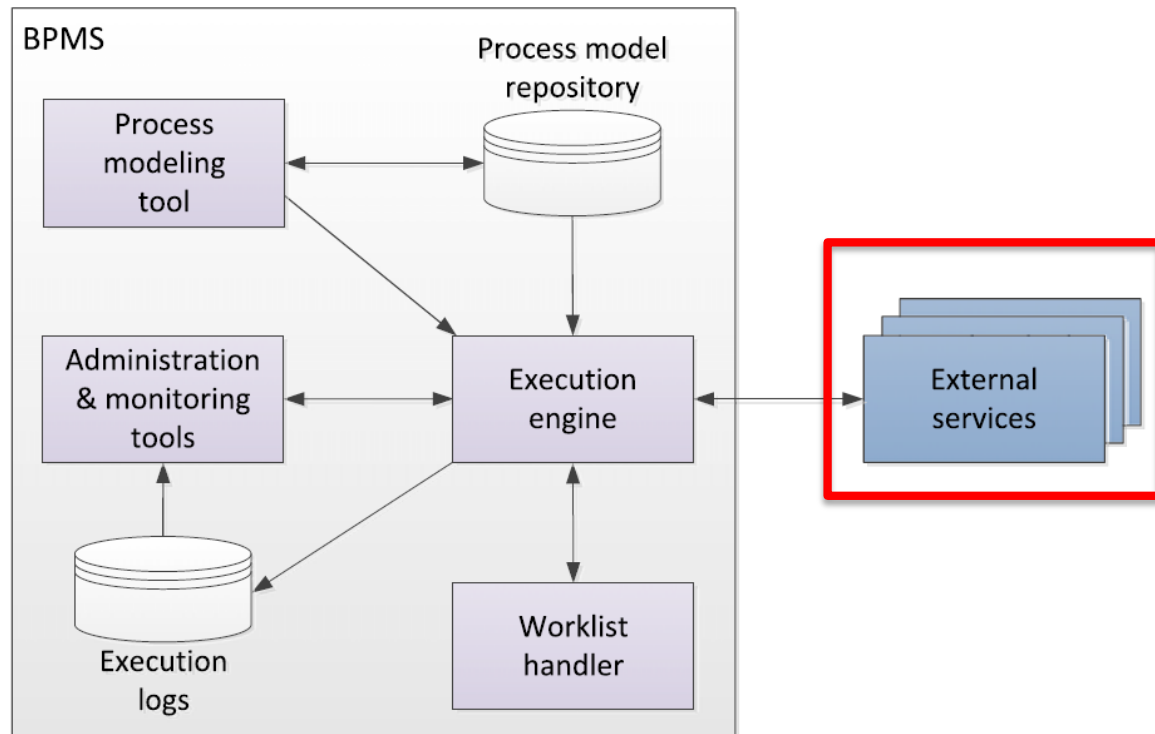
Question time

Trusted Artifact-driven Process Monitoring
of Multi-Party Business Processes with Blockchain

Traditional solutions

Interfacing external organizations

- One organization grants access to its BPMS
- Federation
- Third parties
- Organizations must trust each other



Traditional solutions

Worklist handler

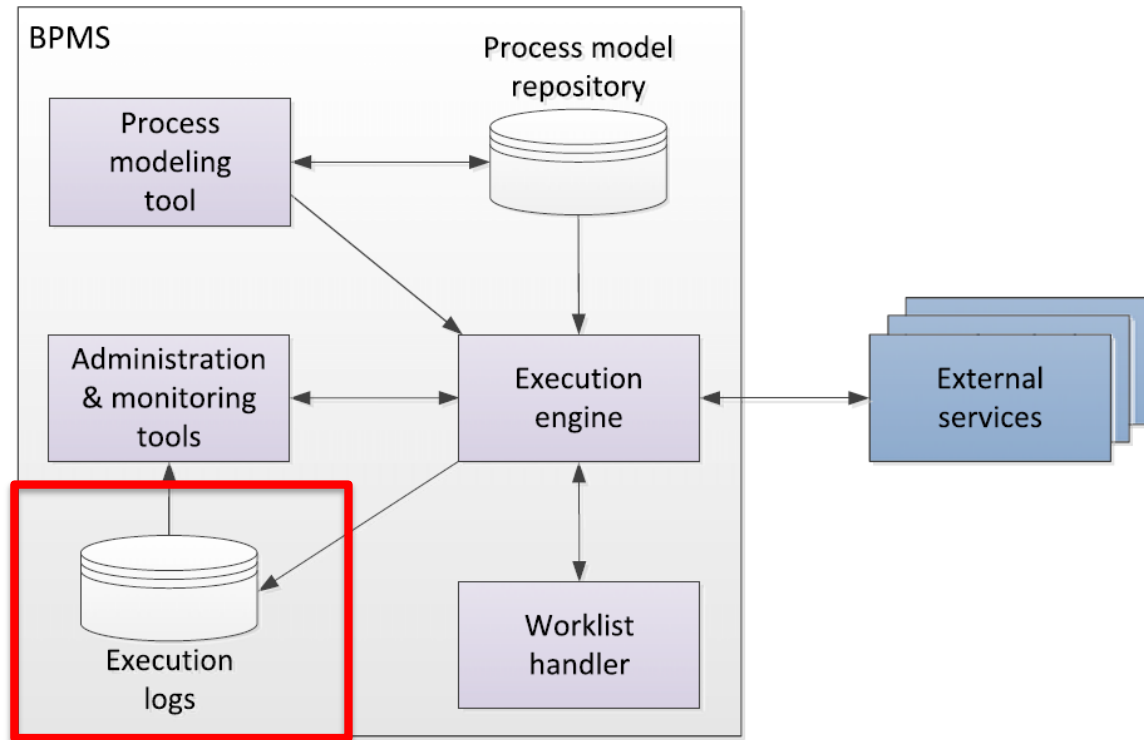
- Notifies operators of their tasks
- Operators have to claim and complete tasks:
 - Claim: started
 - Complete: ended
- They rely on manual notifications:
 - Mistakes occur
 - Operators may forget
 - False information can be introduced
- Organizations must trust their employees

The screenshot displays the Camunda Tasklist web application. On the left, a sidebar shows a task list with columns for task name, due date, creation time, and priority. The main area shows a task form for 'Prepare Bank Transfer' with fields for 'Invoice Receipt', 'Invoice A...', 'Invoice Nu...', 'Creditor', 'Amount', 'Invoice Number', and 'Approved by'. A 'Claim' button is highlighted with a red box in the top right corner, and a 'Complete' button is highlighted with a red box in the bottom right corner.

Traditional solutions

Properly storing logs

- Notifications stored into execution logs
- Logs are analyzed for monitoring
- Logs can be tampered with
- Logs can be destroyed



Evaluation

Tank[empty, detached]



Packing

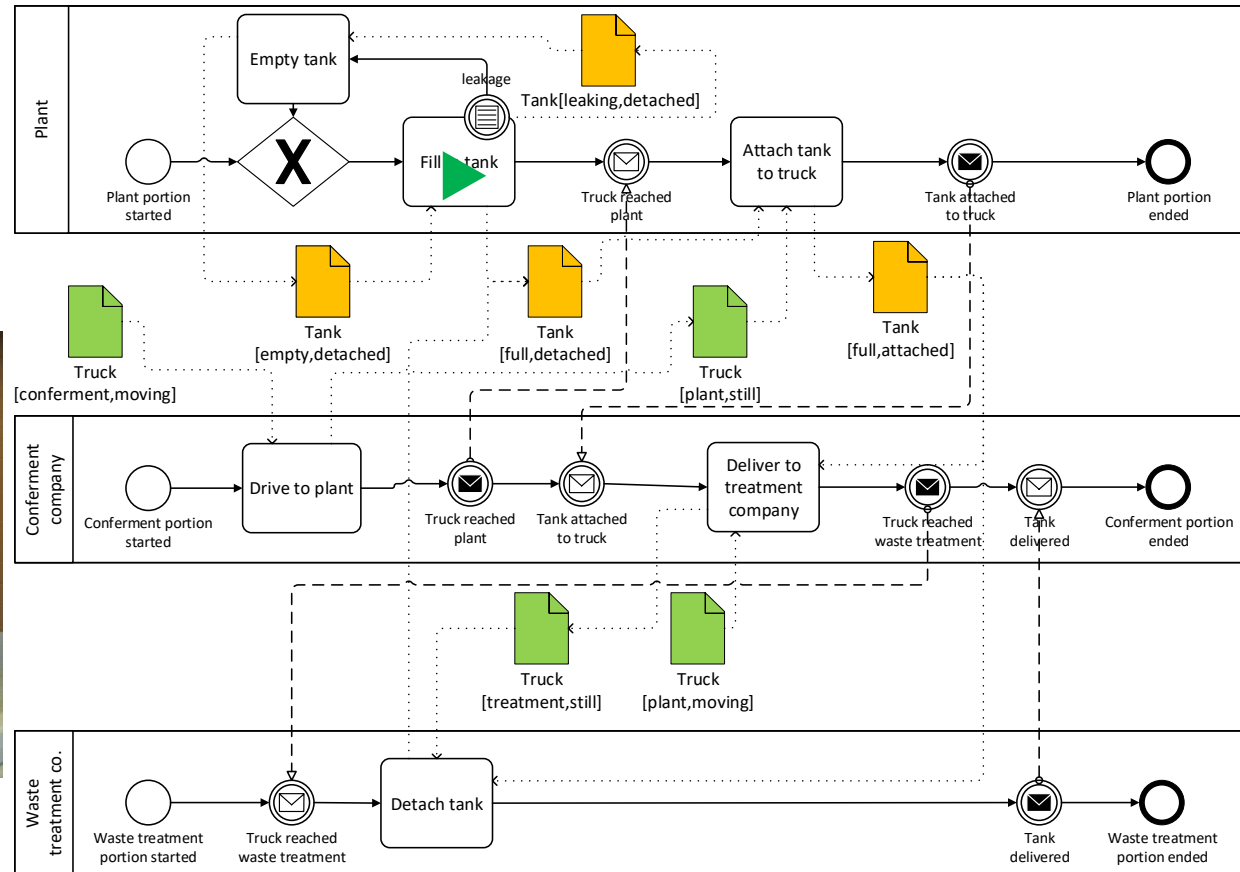


Image source: <https://www.shaw.af.mil/News/Photos/igphoto/2000108024/>

Evaluation

Tank[empty, detached]

Tank[full, detached]



Packing ✓

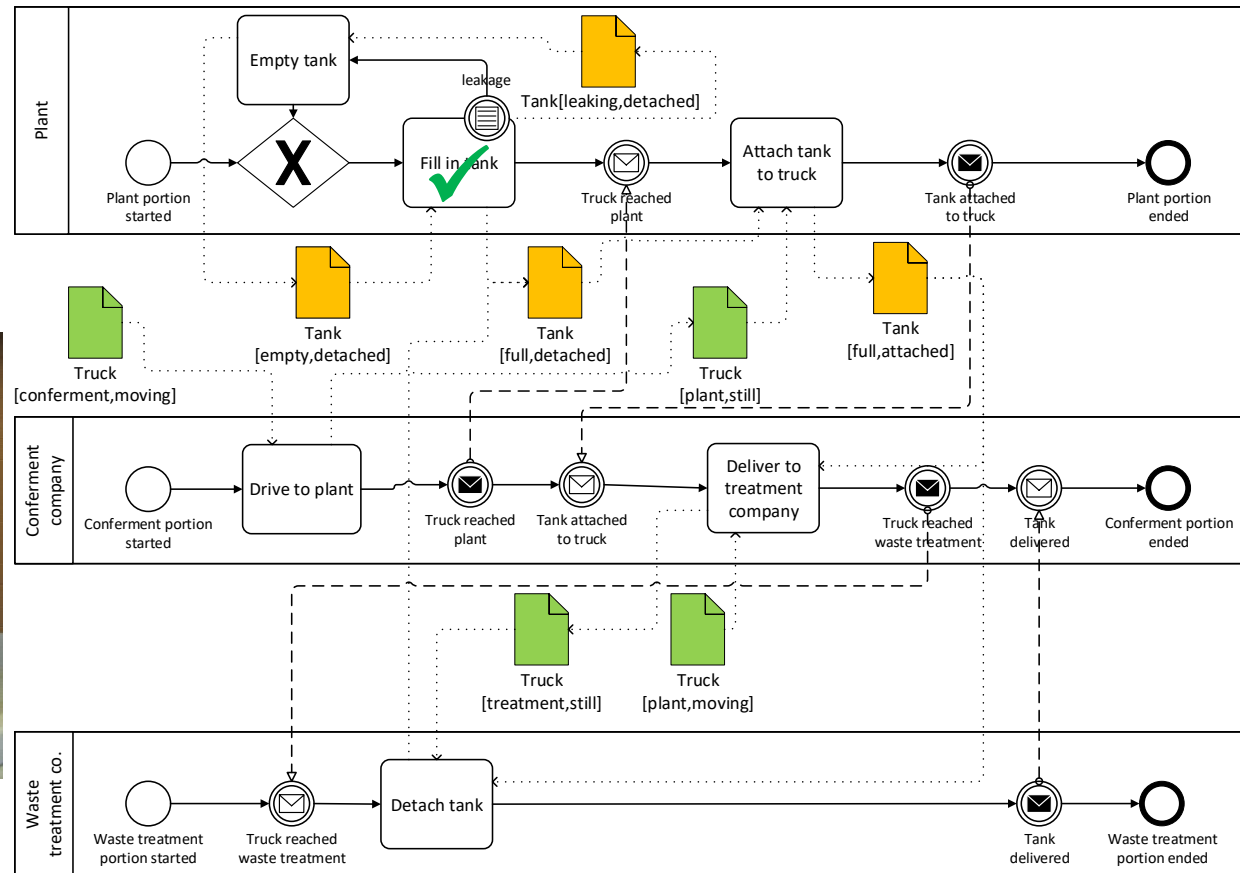


Image source: <https://www.shaw.af.mil/News/Photos/igphoto/2000108024/>

© 2019 G. Meroni, P. Plebani, F. Vona

POLITECNICO MILANO 1863

Evaluation

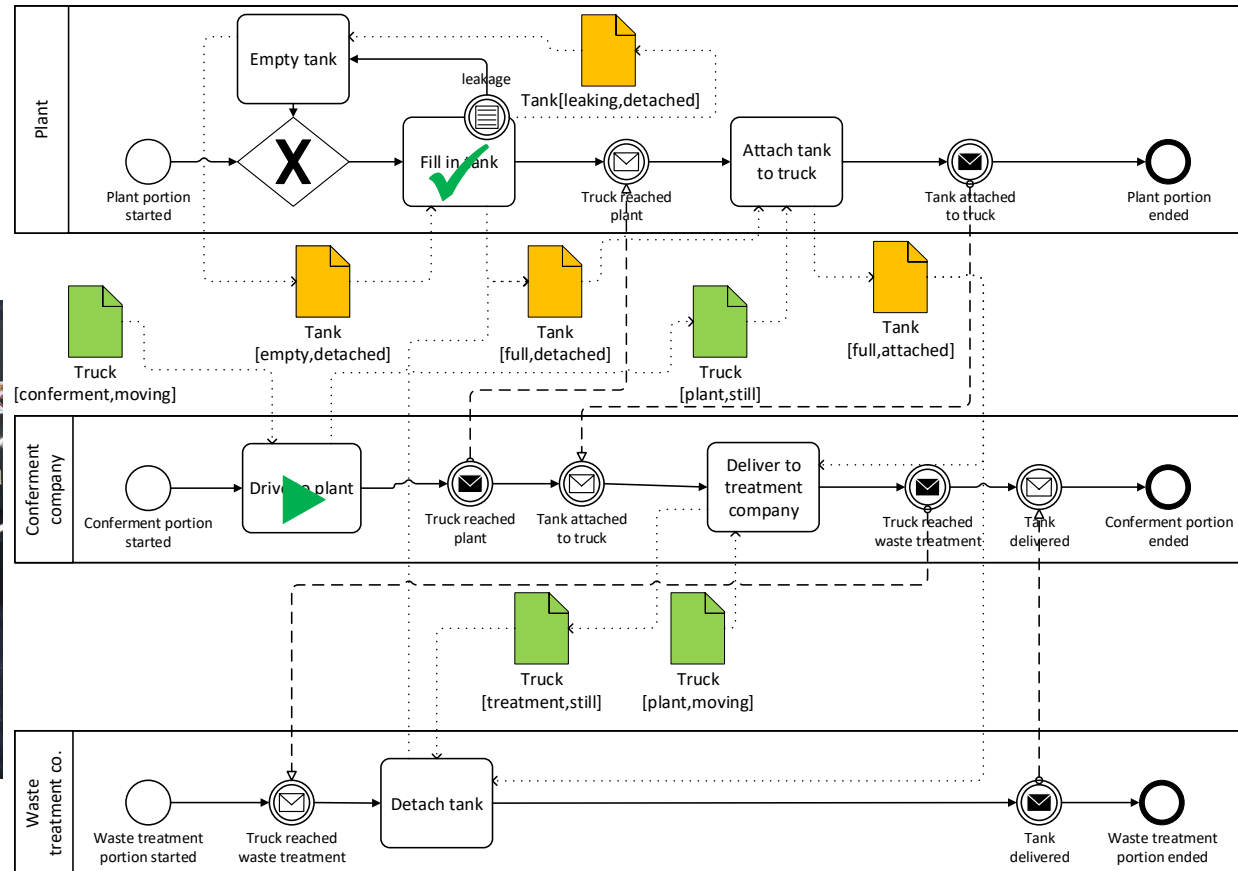
Tank[empty, detached]

Tank[full, detached]

Truck[conferment, moving]



Conferment



Evaluation

Tank[empty, detached]

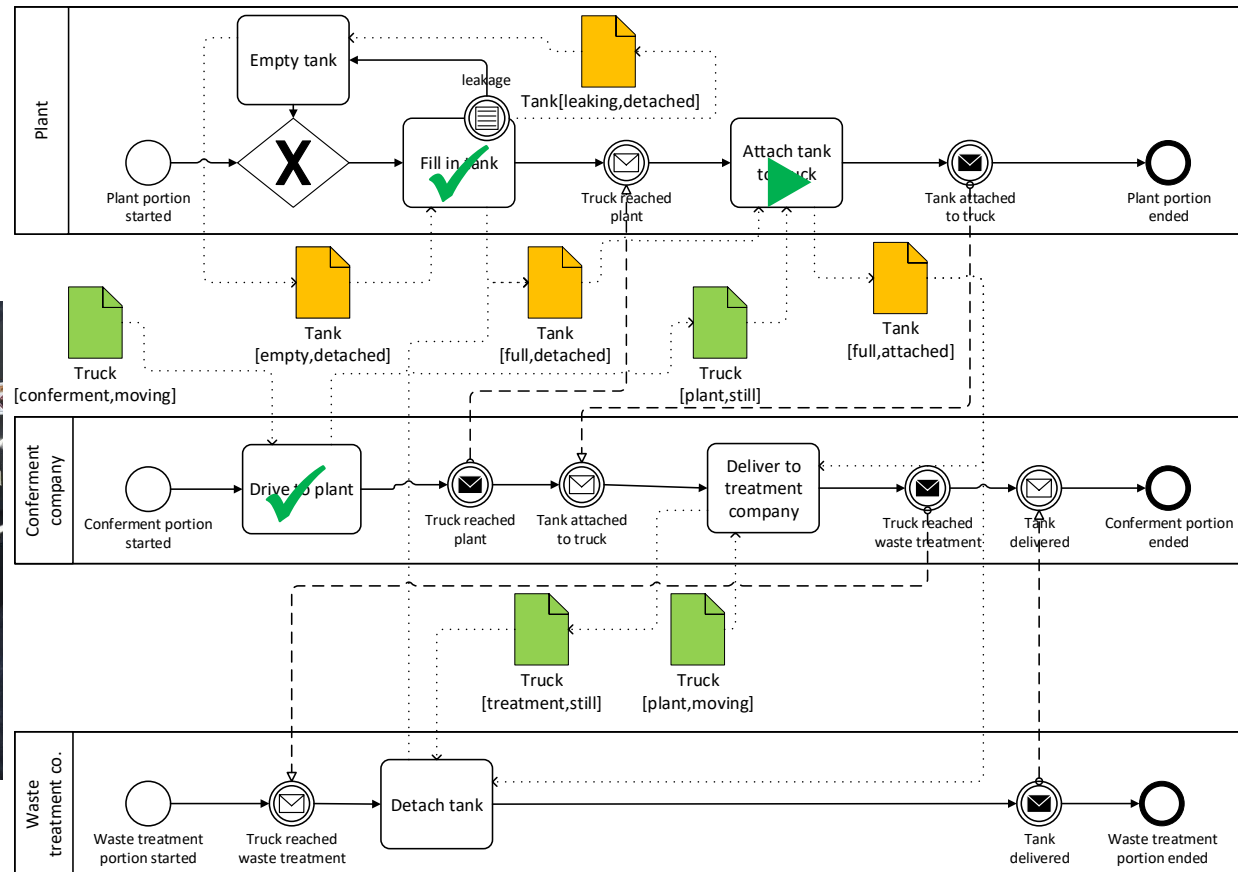
Tank[full, detached]

Truck[conferment, moving]

Truck[plant, still]



Conferment



Evaluation

Tank[empty, detached]

Tank[full, detached]

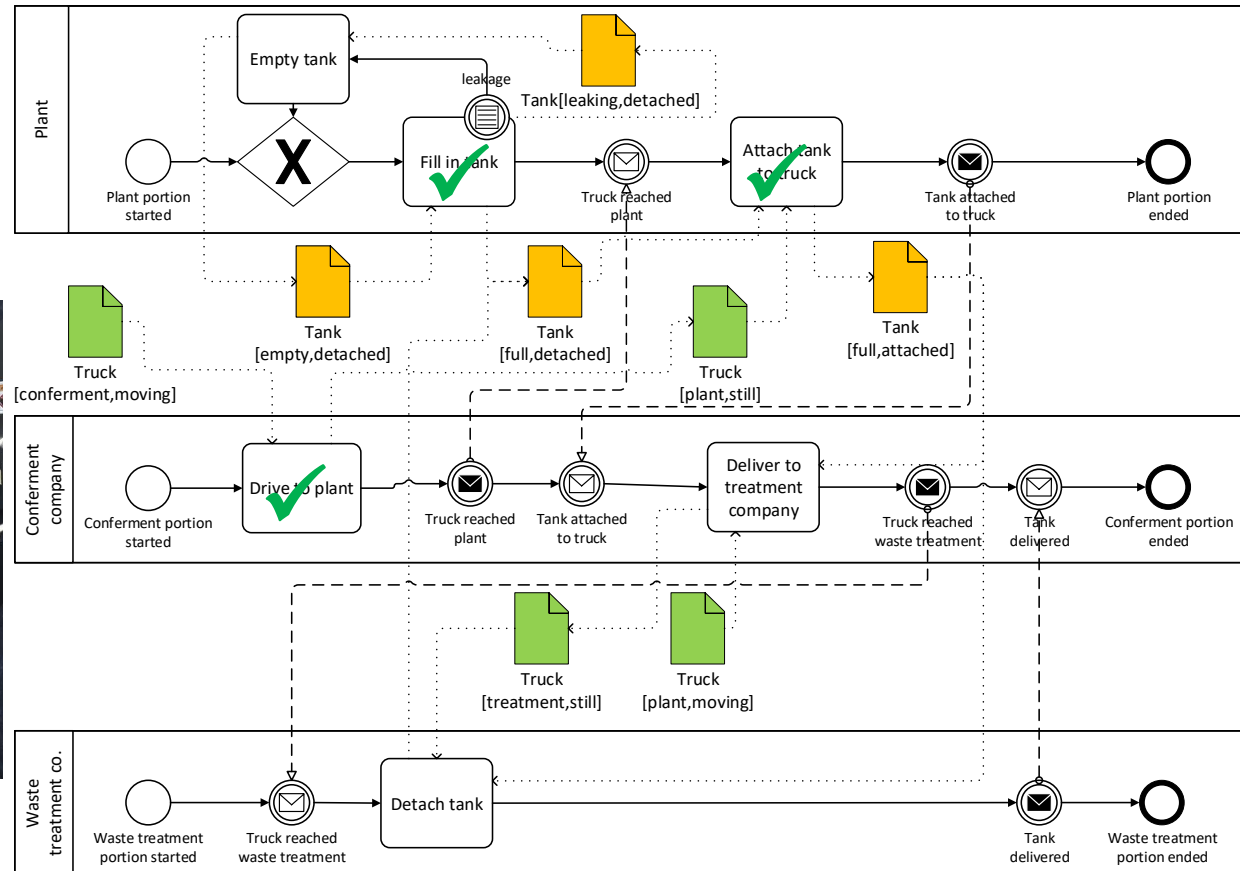
Truck[conferment, moving]

Truck[plant, still]

Tank[full, attached]



Conferment 



Evaluation

Tank[empty,detached]
 Tank[full,detached]
 Truck[conferment,moving]
 Truck[plant,still]
 Tank[full,attached]
Truck[plant,moving]



Shipment

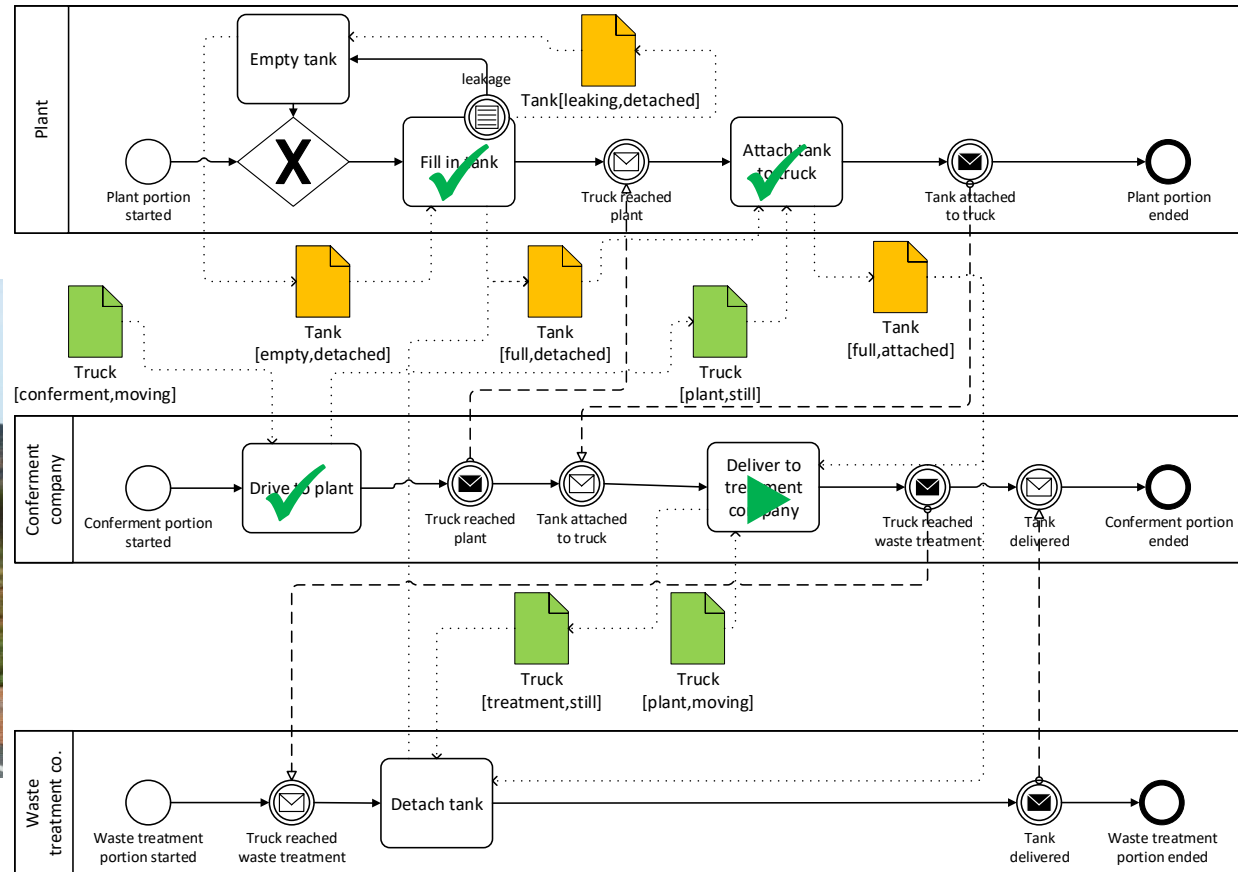


Image source: <https://picryl.com/media/semi-trailers-truck-road-transportation-traffic-17c822>

Evaluation

- Tank[empty,detached]
- Tank[full,detached]
- Truck[conferment,moving]
- Truck[plant,still]
- Tank[full,attached]
- Truck[plant,moving]
- Tank[full,detached]**



Shipment ✘

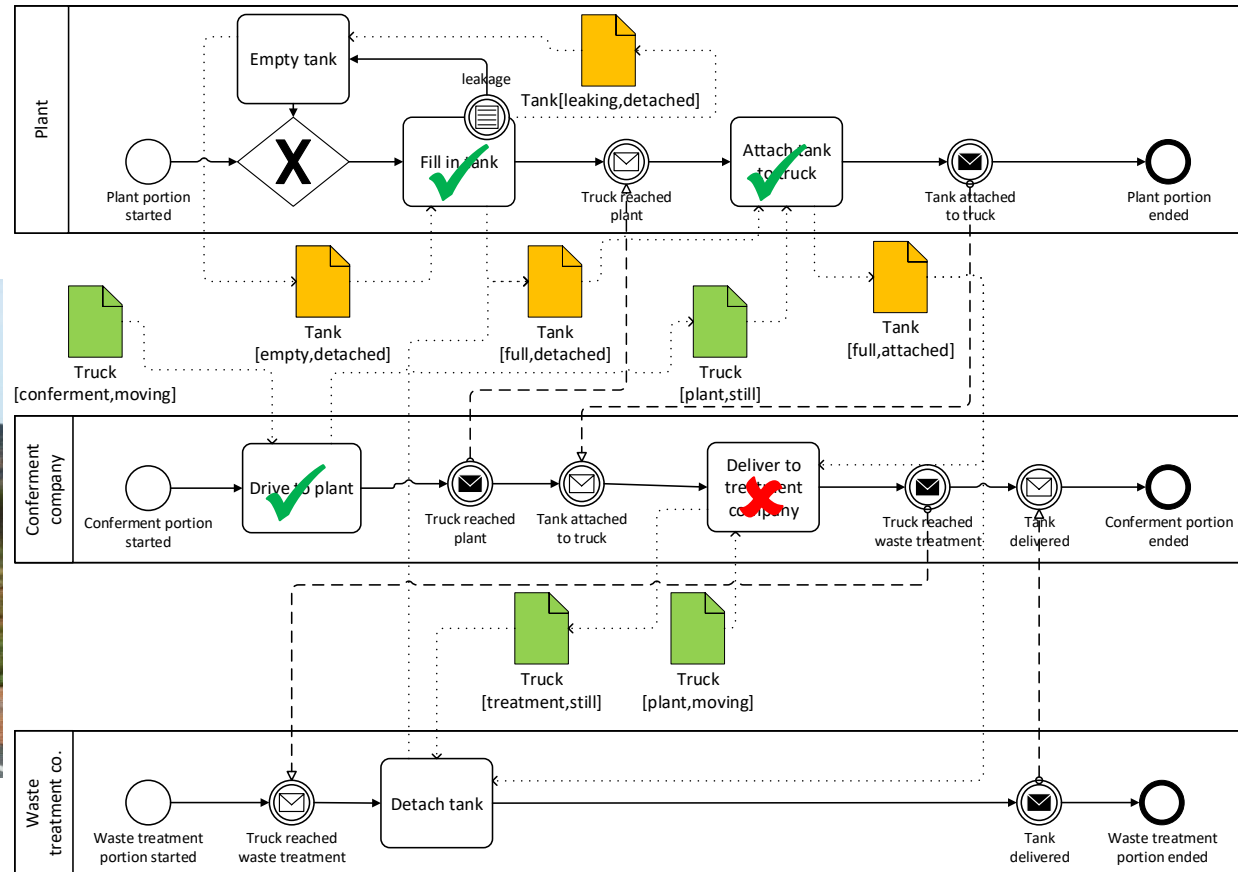


Image source: https://de.wikipedia.org/wiki/Datei:Umweltbelastung_durch_Industrie%C3%B6le_in_Industriebranche_2.jpg

Evaluation

Tank[empty, detached]

Tank[full, detached]

Truck[conferment, moving]

Truck[plant, still]

Tank[full, attached]

Truck[plant, moving]

Tank[full, detached]



Improper disposal

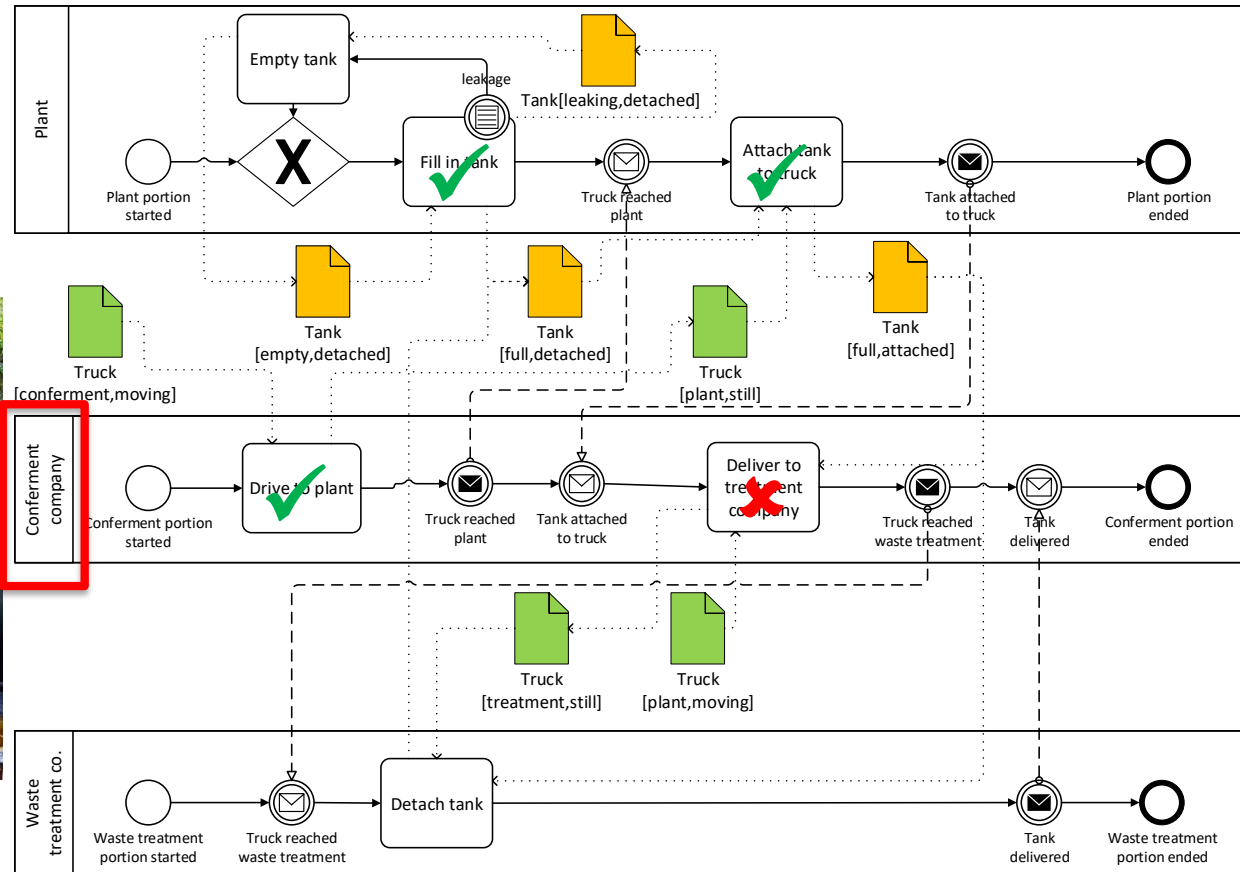


Image source: https://de.wikipedia.org/wiki/Datei:Umweltbelastung_durch_Industrie%C3%B6lle_in_Industriebranche_2.jpg

Trusted artifact-driven process monitoring

Fully blockchain-based monitoring platform

```
1  contract Blockclient {
2      string processModel; //process model
3      struct State { //event
4          uint id;
5          address sender;
6          string artifact;
7          string status;
8          string timestamp;
9          string data; }
10     mapping(uint => State) public states; //list of events
11     uint stateCounter;
12     struct participant {
13         bytes32 encodedArtifact;
14         address addr; }
15
16     mapping(address => participant) public participants; //participants
17     mapping(uint => address) participantsIndex;
18     uint participantCounter;
19 }
```

Trusted artifact-driven process monitoring

Fully blockchain-based monitoring platform

```
19
20 function Blockclient(string _processModel, address[] _addrs, bytes32[]
    _encodedArtifacts) payable public {
21     for (uint p = 0; p < _addrs.length; p++) { //add participants
22         participants[_addrs[p]].addr = _addrs[p];
23         participants[_addrs[p]].encodedArtifact = _encodedArtifacts[p];
24         participantsIndex[p] = _addrs[p];
25         participantCounter++;}
26     processModel = _processModel;} //store process model
27
28 function writeState(string _artifact, string _status, string _timestamp,
    string _data) payable public {
29     if (participants[msg.sender].encodedArtifact == stringToBytes32(_artifact
        ) { //check identity of sender and ownership of artifact
30         stateCounter++; //increment state counter
31         states[stateCounter] = State(stateCounter, msg.sender, _artifact,
            _status, _timestamp, _data); //store state data
32         LogWriteState(stateCounter, msg.sender, _artifact, _status, _timestamp,
            _data); }}} //emit a new event
33 }
```

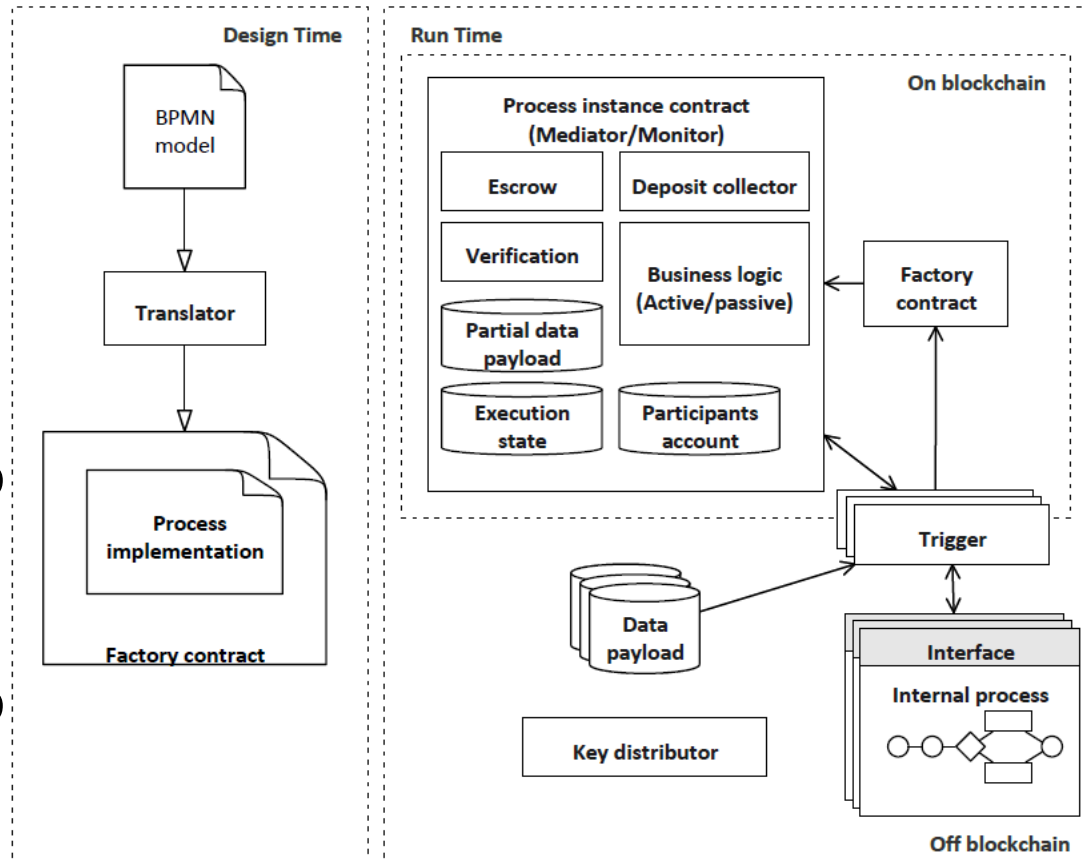
Trusted artifact-driven process monitoring DFS-blockchain hybrid monitoring platform

```
1  contract IPFSblockclient {
2      string processModelHash;
3      struct StateHash {
4          uint id;
5          string mHash;
6      }
7      mapping(uint => StateHash) public hashes;
8      uint hashCounter;
9
10     function writeHash(string _topic, string _mHash) payable public {
11         if (participants[msg.sender].encodedTopic == stringToBytes32(_topic)) {
12             //check identity of sender and ownership of artifact
13             hashCounter++; //increment state counter
14             hashes[hashCounter] = StateHash(hashCounter, _mHash); //store state
15             hash
16             LogWriteHash(hashCounter, _mHash); }}} //emit a new event
17 }
```

Monitoring with blockchain

Blockchain-powered BPMS

- Weber et al. [2] first to use blockchain for process monitoring
- They redesigned a BPMS to use Ethereum
 - Smart contracts to automate process execution
 - Smart contracts to monitor the process

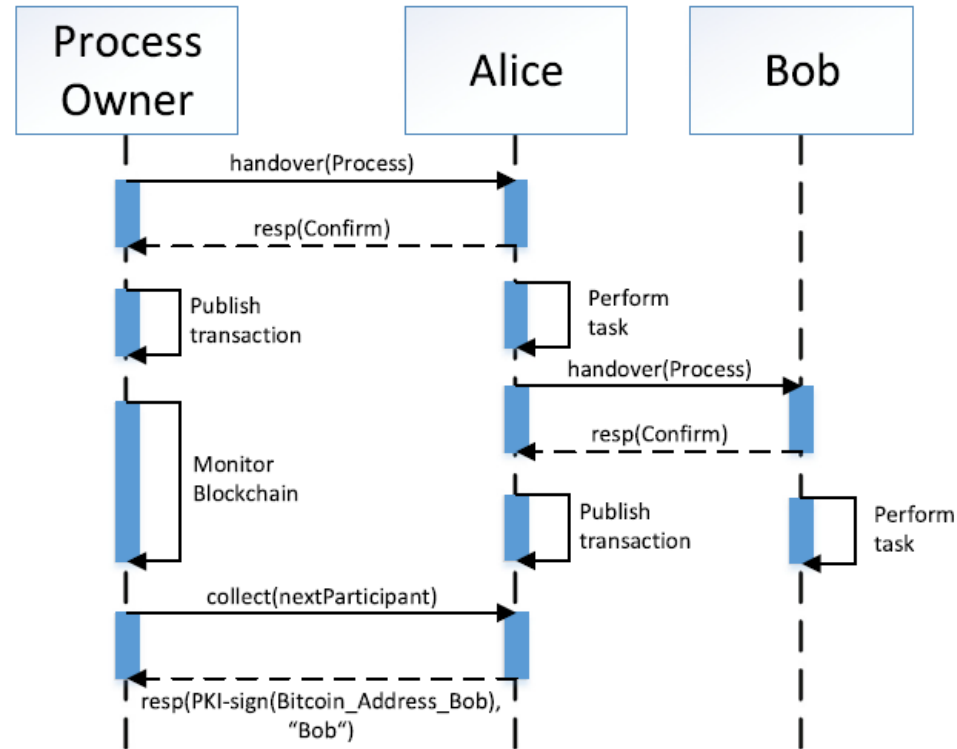


[2] Weber I., Xu, X., Riveret, R., Governatori, G., Ponomarev, A., Mendling, J.: Untrusted Business Process Monitoring and Execution Using Blockchain, BPM 2016

Monitoring with blockchain

Blockchain-powered process event log

- Prybila et al [3] rely on Bitcoin to monitor processes
- Blockchain used as general ledger
 - No smart contracts
 - Handover mechanisms for authorized writes
- Better for unknown or changing participants



[3] Prybila, C., Schulte, S., Hochreiner, C., Weber, I.: Runtime verification for business processes utilizing the Bitcoin blockchain, FGCS 2017

Process monitoring with blockchain

Current limitations

- Trust issues not entirely addressed
 - Notifications are still provided by human operators
 - No verification if notifications are correctly sent
- Focus on choreography
 - Internal activities not monitored
 - Artifacts not monitored