Combining Artifact-driven Monitoring with Blockchain: Analysis and Solutions

Giovanni Meroni and Pierluigi Plebani

CAiSE 2018 BIOC Workshop – Tallinn, 11 June 2018

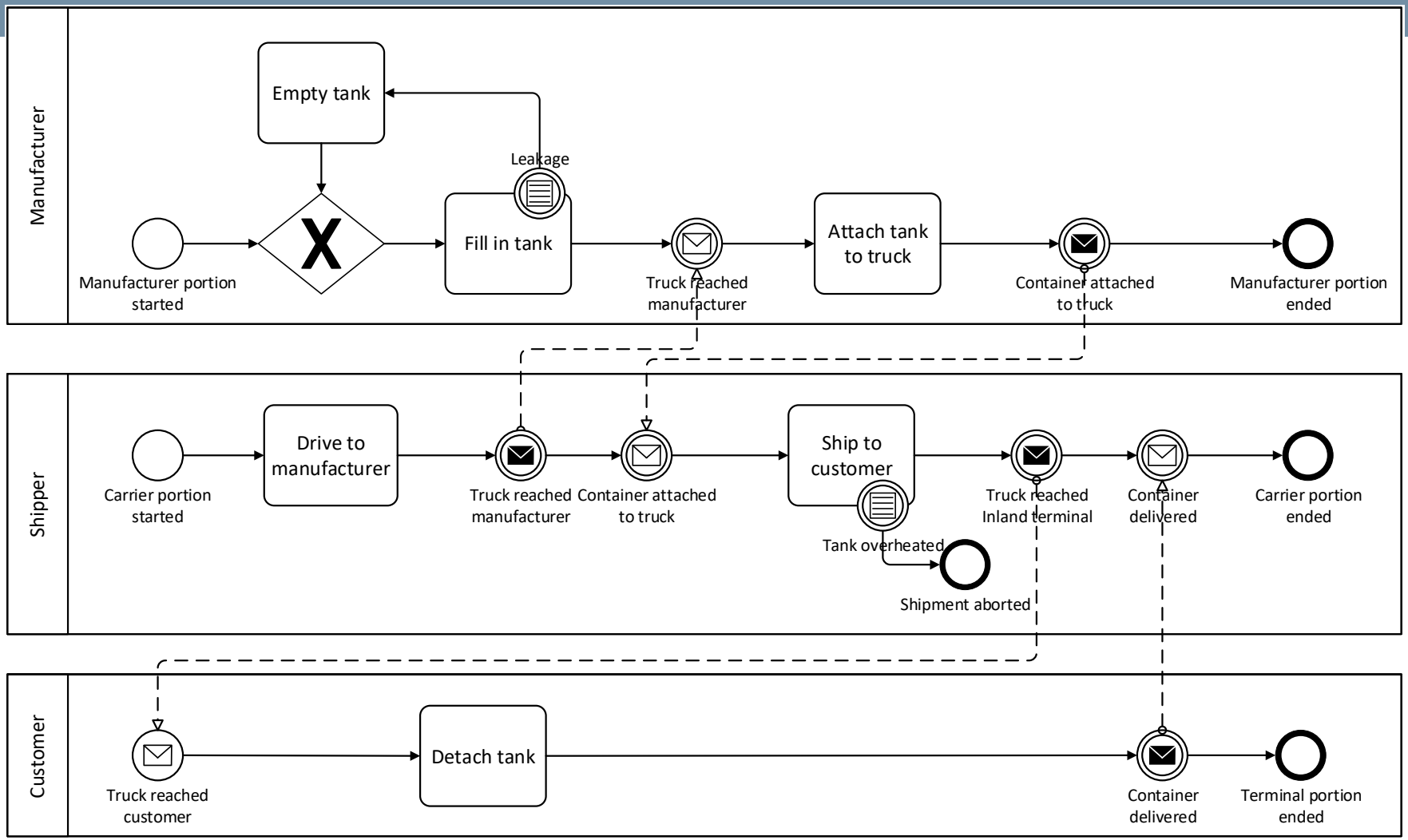
Agenda

- Monitoring multi-party processes
- Artifact-driven monitoring
- The issue of trust
- Exploiting blockchain to achieve trusted monitoring

Context

- Following the “servitization” paradigm, companies tend to externalize activities and goods.
- Many intra-organizational processes are becoming multi-party:
  - Portions of a process are outsourced to external organizations
  - Companies interact with goods without owning them
- Organizations are interested in monitoring the execution of multi-party processes as a whole
  - No guarantee that outsourced activities are performed as agreed
  - No guarantee that goods given to other companies are manipulated as agreed

Motivating example



Motivating example

Image source: https://www.flickr.com/photos/timothywildey/4682999460

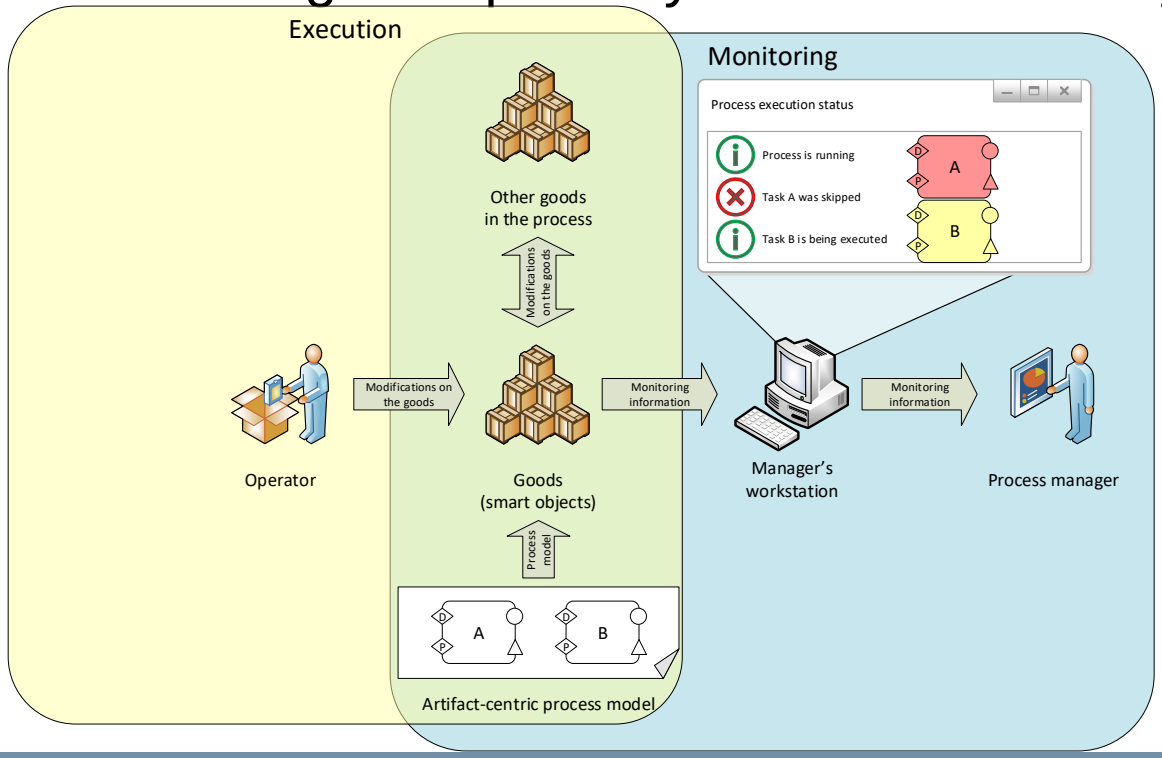
Artifact-driven process monitoring

- Goods participate in multi-party processes
  - Goods belong to a specific organization
  - Goods have visibility on activities interacting with them, regardless of the organization performing the activities
  - The conditions of the goods can be altered by organizations not owning such goods
- Objects participating in a process are named **artifacts**
- Goods can be seen as artifacts
  - They actually are a subset, since artifacts can also be virtual
  - For our purposes, goods = artifacts
- Idea: Artifact-driven process monitoring [1]
  - Monitoring is directly performed on the artifacts
  - The artifact “knows” when its conditions change
  - The artifact “knows” when activities are executed

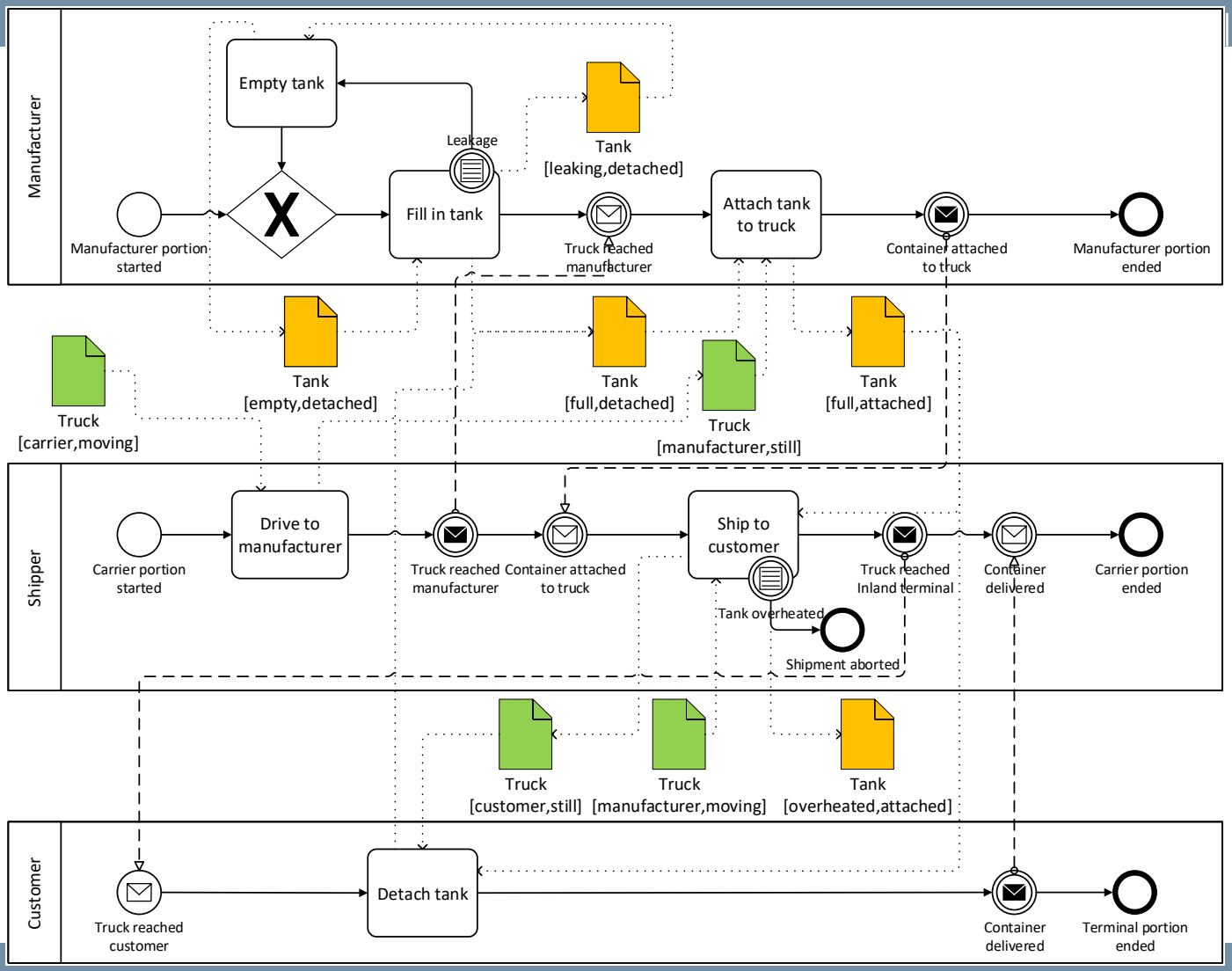
[1] G. Meroni, L. Baresi, M. Montali, P. Plebani - “Multi-party business process compliance monitoring through IoT-enabled artifacts”, Information Systems, Volume 73, 61-78

Artifact-driven process monitoring

- Exploit the Internet of Things to monitor processes
- Make objects aware of the process they participate in
- Perform monitoring transparently and autonomously

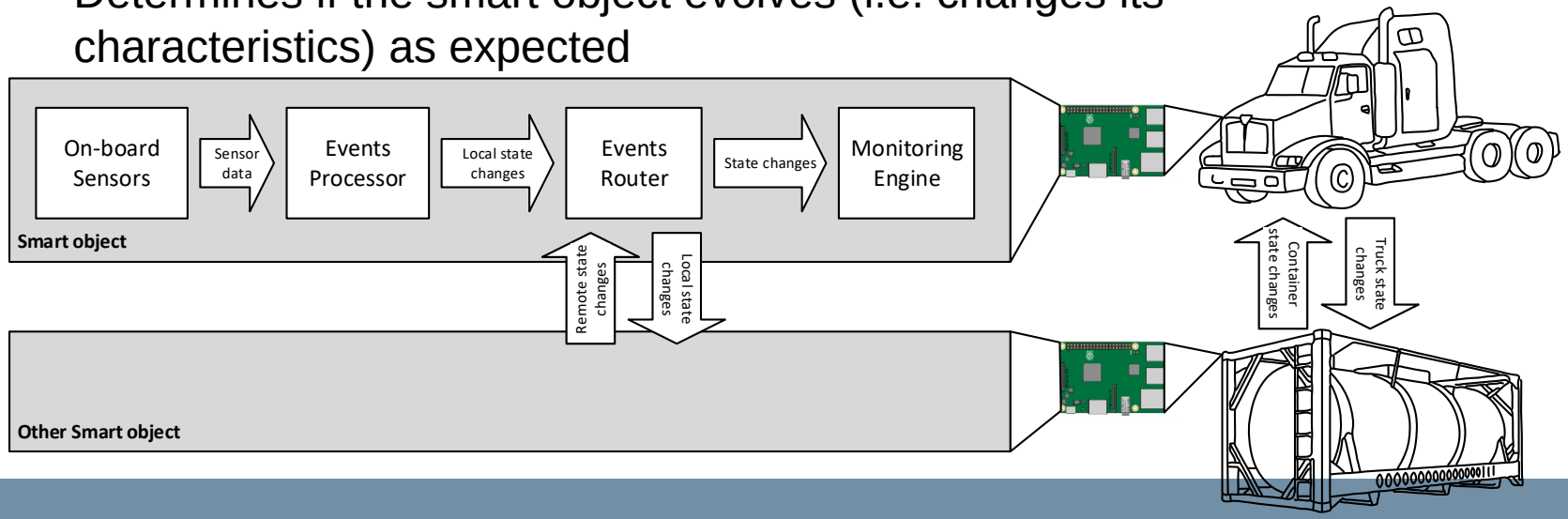


Back to the example



Reference architecture of a monitoring platform

- **Events processor**: Determine a discrete state of the smart object based on its conditions, obtained through sensors
- **Events router**: Routes events relevant for other smart objects and receives external events in a Machine-to-Machine (M2M) fashion.
- **Process engine**: Monitors the process:
  - Determines if activities are executed according to the proces definitions
  - Determines if the smart object evolves (i.e. changes its characteristics) as expected



The issue of trust

- Artifact-driven monitoring alone does not entirely solve the problem of trust.
- The configuration of the smart objects is up to the single organizations:
  - They configure the smart objects with the process model
  - They define rules to determine from sensor data the state of the smart object
  - No guarantee that smart objects are configured as agreed

Exploiting blockchain

- Blockchain are an effective way to let untrusted entities trust each other:
  - Information is encapsulated into blocks
  - A block must be validated by multiple independent entities before being stored
  - Blocks are persistent and immutable
  - Agreements can be formalized with smart contracts

Exploiting blockchain

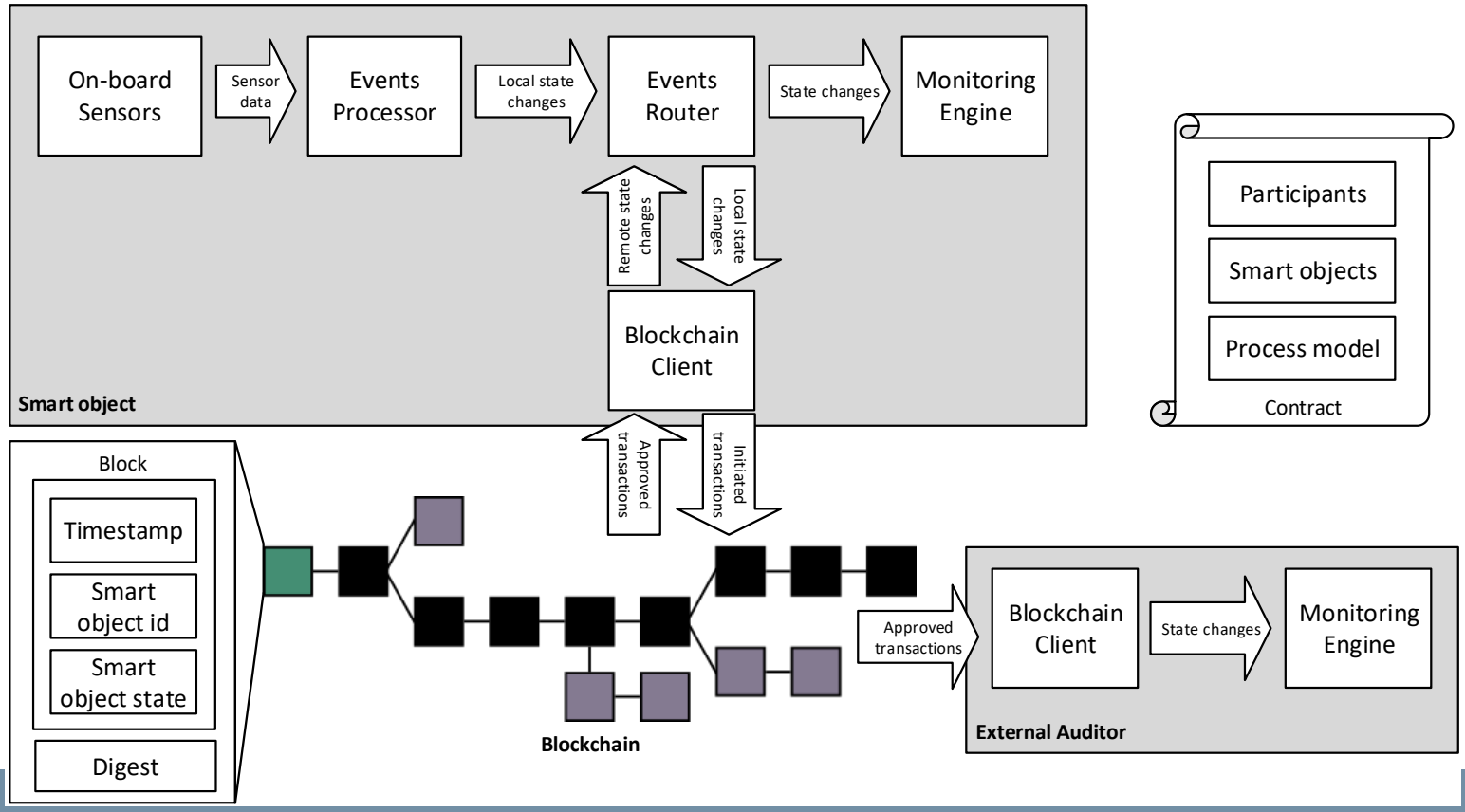
- We propose two possible modifications to artifact-driven monitoring architecture to include blockchain:
  - State-oriented block
  - Sensor-oriented block
- Both approaches are based on permissioned blockchain:
  - Maintains the process confidential to the participants
  - No need to implement computational-heavy block generation algorithms

State-oriented block

- Before monitoring starts, the process model is formalized as a smart contract
  - It must be approved by all participants to be valid
- A new block is written when a smart object detects a change in its state
- To validate the block, the identity and ownership of the smart objects are verified:
  - The smart object producing the block must be the same as the one whose state changed
  - The smart object must be owned by an organization participating in the process

State-oriented block

- Advantages: easy to monitor by external auditors
- Disadvantages: cannot determine if states are correct

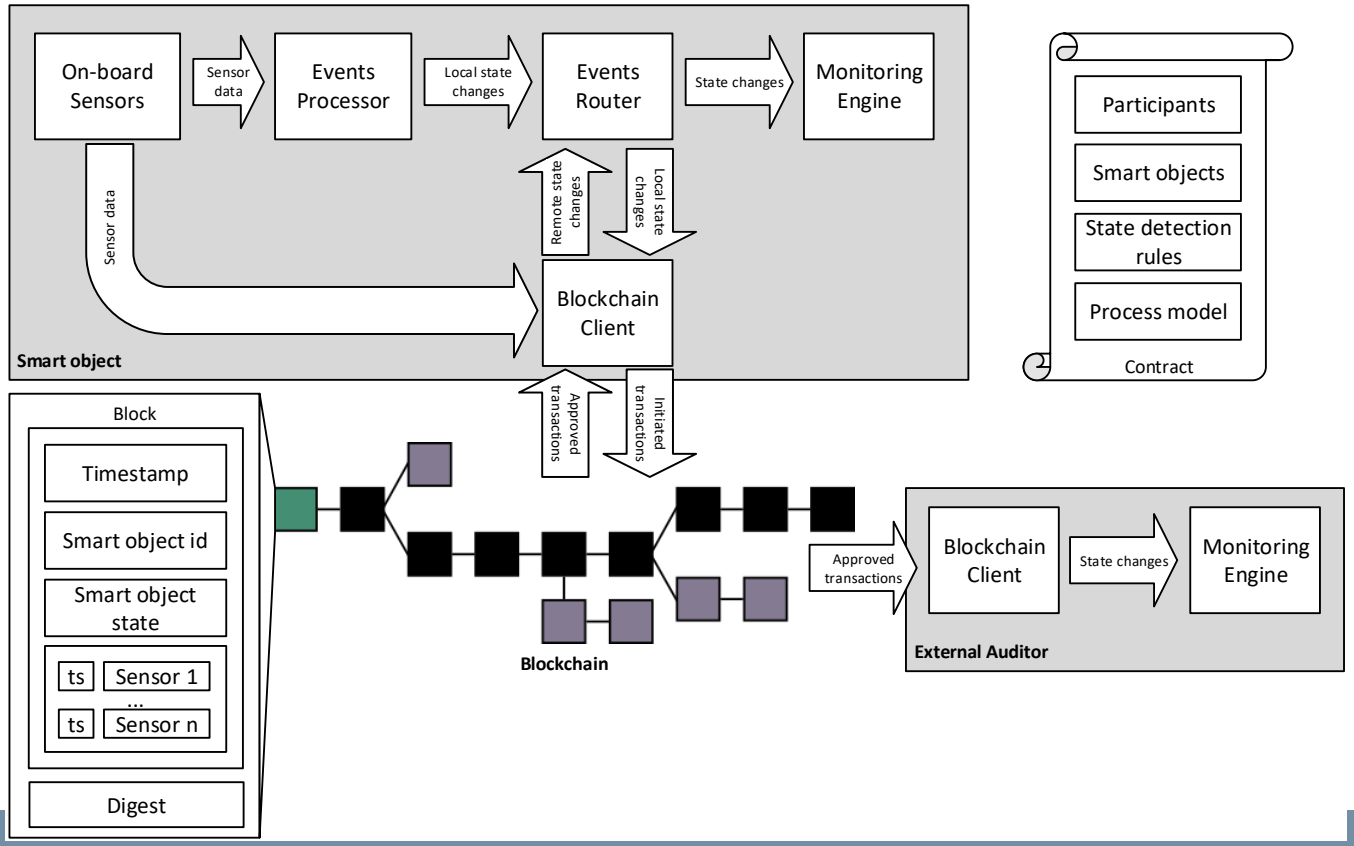


Sensor-oriented block

- Smart contract includes both process model and rules to derive the state of the smart objects from sensor data
  - All participants must also agree on how the states are derived
- A block includes also all sensor data that caused a new state to be detected
- Only blocks that satisfy the smart contract are considered correct:
  - If the state inferred from sensor data differs from the one indicated in the block, then the block is invalid
  - An invalid block is ignored

State-oriented block

- Advantages: even greater level of trust in monitoring data
- Disadvantages: much more intensive use of blockchain



Final remarks

- The synergy between blockchain and artifact-driven monitoring increases the trust among cooperating organizations
- This solution still has limitations:
  - The initial setup of a blockchain can be cumbersome
  - Small-sized permissioned blockchain can be taken over by a single organization
  - The validation of blocks is slow, thus unsuited for monitoring processes timely

TRASPORTI ITALIA 2020

UNIONE EUROPEA
Fondo europeo di sviluppo regionale

PON Ricerca e Competitività 2007-2013

Ministero dell'Istruzione, dell'Università e della Ricerca

POLITECNICO
MILANO 1863

Thanks for your attention

This work has been partially funded by the Italian Project ITS2020 under the Technological National Clusters program