Pushing Business Data Processing Towards the Periphery

Pablo Guerrero
GK E-Commerce
Dept. of Computer Science
TU Darmstadt, Germany
pguerrer@gkec.informatik.tu-darmstadt.de

Kai Sachs, Mariano Cilia, Alejandro Buchmann
Databases and Distributed Systems Group
Dept. of Computer Science
TU Darmstadt, Germany
<ksachs,cilia,buchmann>@dvs1.informatik.tu-darmstadt.de

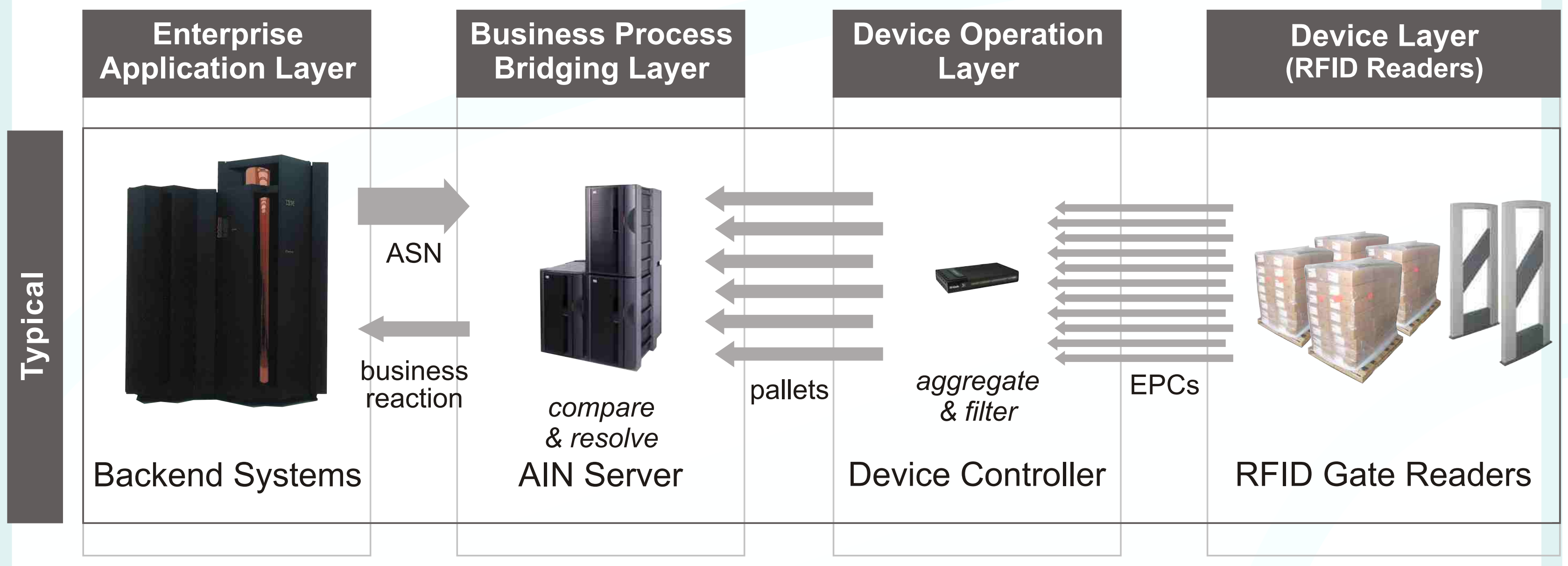
Christof Bornhövd
SAP Research
RC Palo Alto, USA
christof.bornhoevd@sap.com

Databases and Distributed Systems Group

Enabling Technologies for Electronic Commerce

Motivation

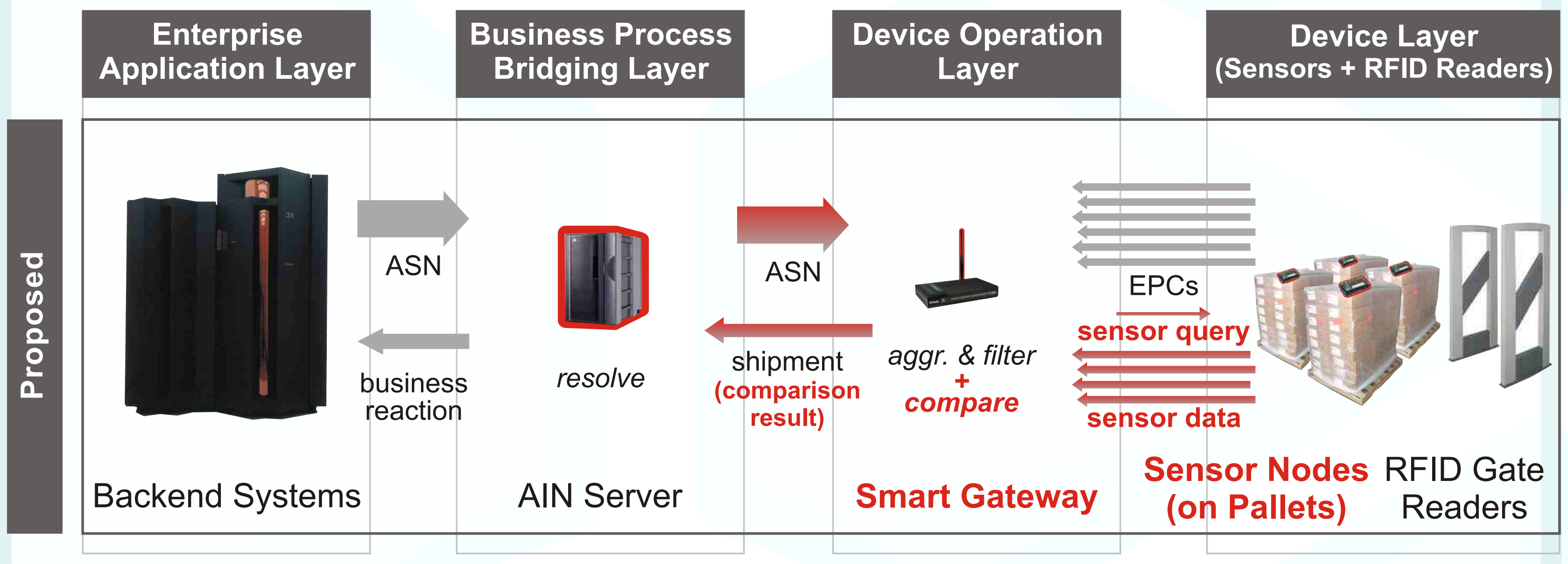
- Usage of **RFID** and **sensor data** in SCM requires the automatic conversion of **large amounts** of raw data into sensible **business** information
- Leads to **performance** and **scalability** issues in existing RFID middleware infrastructures
  - Typical systems are structured in four layers:



- Such a **centralized** system does not scale well with increasing load

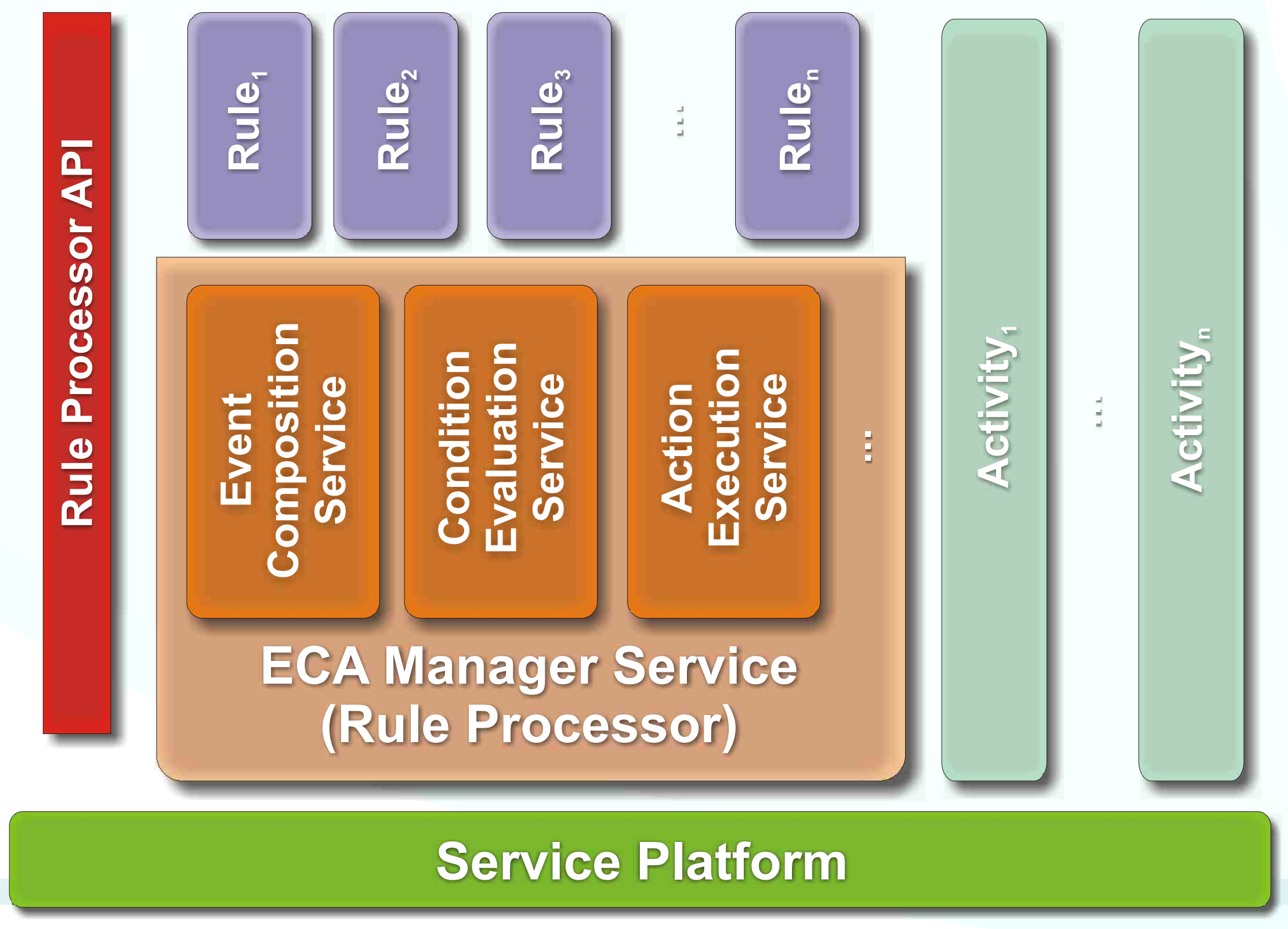
Proposed Approach

- Move from existing monolithic architecture towards a more flexible, distributed architecture
  - **Shift** part of the **business logic** closer to the **point of observation**, exploiting processing power of smart devices
  - React locally for better system scalability and throughput:
    - Improve system **scalability** by reducing network traffic, and
    - Shorten system **response time** by taking actions closer to the edge and hence enabling further interactions



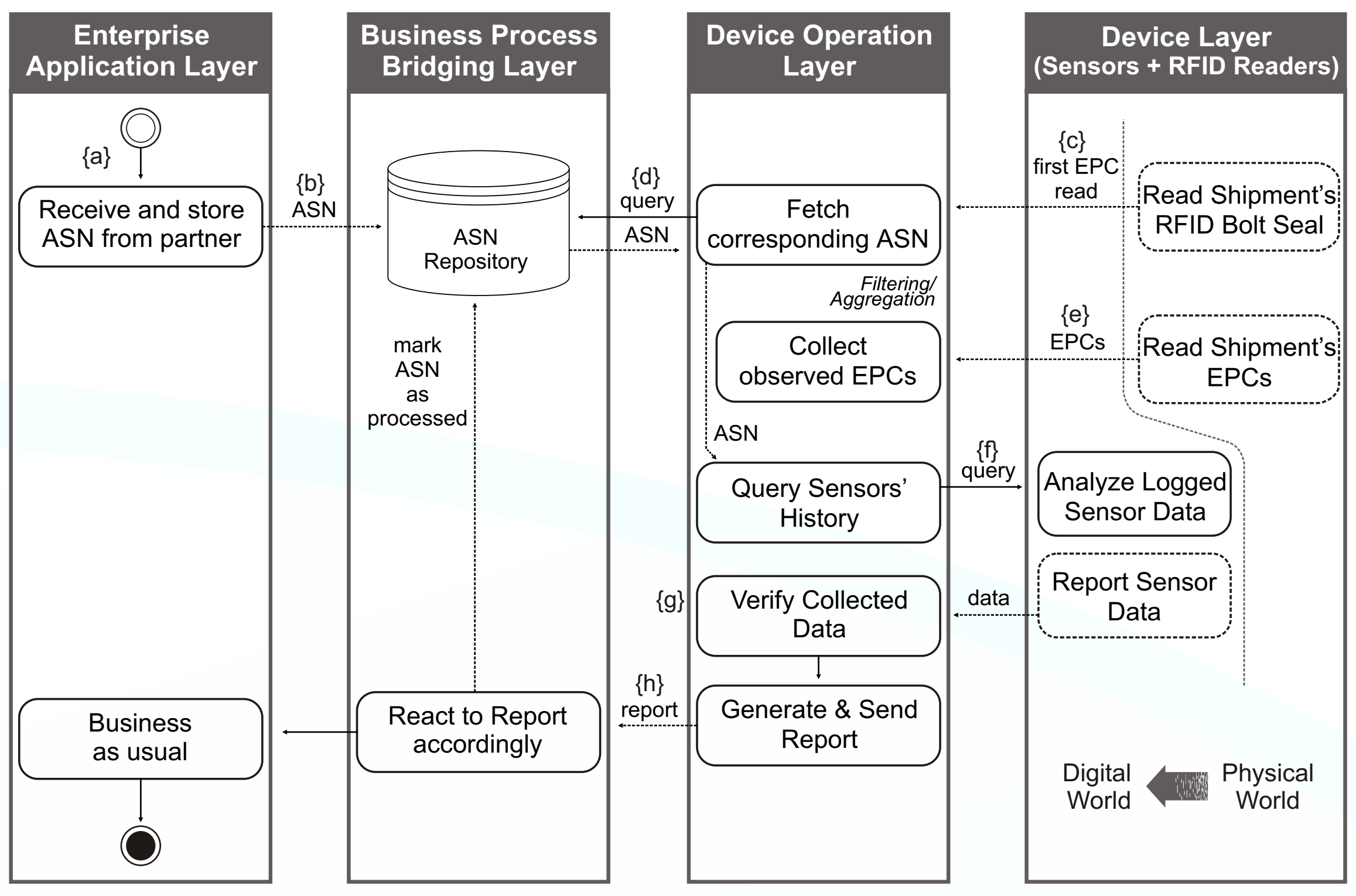
System Architecture

- Business rules are defined in terms of **Event-Condition-Action** (ECA) rules - a system and platform **independent** format
- Small footprint **rule engine** that runs on smart devices
- Component-based design, composed by *elementary services*
  - Basic event detection and event composition,
  - Condition evaluation
  - Action execution
- Use of a **service platform** offering component life-cycle, remote deployment and management



Demonstration Use Case

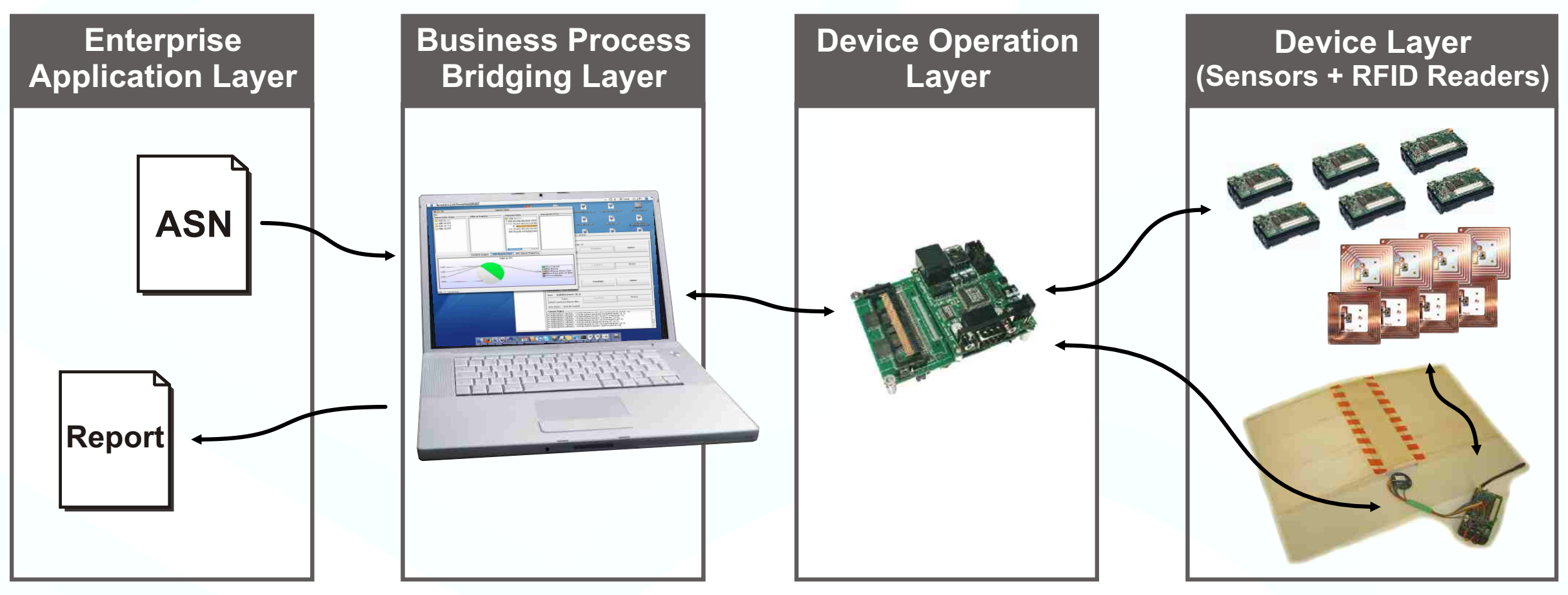
- **Advance Shipping Notice (ASN)**



- Representative SCM example:
  - Multi-partner interaction
  - Involves intensive processing
  - Has a clear output

Demonstration Setup

- Smart pallets
  - RFID-tagged with EPC SGTIN 64 encoding, ISO 15693 tags
  - Some with sensor nodes, running a custom TinyOS data-collection sw.
- RFID reader attached to a sensor node *(for experimentation only!)*
  - 13.56 Mhz, ISO 15693 mini reader
- Stargate platform running Linux BSP and IBM's J9 JVM
- Rule Engine implemented in Java as OSGi (Oscar) bundles
- 4 ECA Rules (following an XML Rule Definition Language)
  - `IncomingEPC`
  - `IncomingSensorData`
  - `EndOfShipment`
  - `EPCException`
- AutoID Node (repository) running on notebook
- Remote deployment and monitoring of rules (visual inspection)

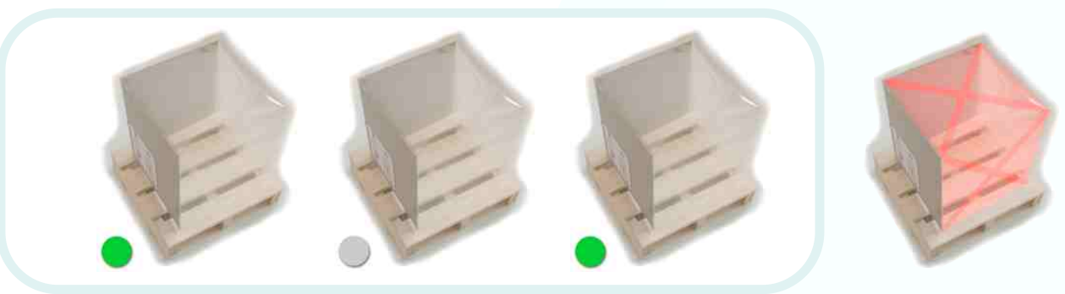


Enterprise Application Layer
Business Process Bridging Layer
Device Operation Layer
Device Layer (Sensors + RFID Readers)
ASN
Report

Scenarios

- Complete, correct shipment
  - *2 pallets: 1 with sensor node*

- Incomplete shipment
  - *3 pallets: 1 missing, 1 unexpected*



- Violation of shipping conditions
  - *1 pallet: values out of range*

- Sensor data unreachable
  - *2 pallets with sensors, 1 is off*

Conclusions & Beyond

- Off-loaded processing from Business Process Bridging Layer
  - Intuitively we have achieved better performance and scalability
  - Extensive rule engine profiling being carried out
- A first step towards a higher level business rule language
  - Richer rule definition language currently being developed
- Exploration of a high-performance ECA Rule Engine
  - e.g. through the usage of clustering techniques