



# BOD Architecture Refactoring STAR 6

Anders Puwanant  
Paco Escobar

# Meeting Objectives

- Clarify background and drivers for the proposed STAR BOD refactoring initiative, as a follow-up to the communication sent to members on October 3, 2018.
- Address questions, comments and concerns from members.
- Kick off the 15 day voting period to approve the refactoring initiative according to STAR bylaws:
  - >50% of members voting
  - >75% of votes in support

# Background

- Globalization and JSON identified by members as prioritized objectives for STAR in the 2017 General Session and 2017 STAR Usage Survey.
- Two workgroups were formed in May 2017. Recommendation from both these workgroups was to first perform a refactoring (upgrade to OAGIS 10 and harmonization of STAR objects and components across BODs) prior to adapting BODs to better meet requirements from global users and developing JSON STAR schemas.
- In December 2017 STAR's SC commissioned a STAR 6 Refactoring Workgroup tasked to detail scope and approach for a refactoring initiative. CDK, Ford, General Motors, Motive Retail and Volvo Group participated in this effort.
- A high level proposal for refactoring was presented to and supported by the STAR General Session in March 2018.
- This proposal has now been further detailed and anchored and is ready to be presented to members for approval.

# What is being proposed?

- Start an initiative to refactor and clean-up current STAR XML BODs (STAR 5 based on OAGIS 9), branching off into STAR 6 based on OAGIS 10, following the process outlined in this presentation.
- For STAR, under an extended transition period, to continue maintain the current STAR 5 branch with extensions and updates, following a similar approach as for the transition from STAR 4 to STAR 5 in 2006 (when both branches were supported for 5 years).
  - Length of the transition period to be proposed by the STAR SC, based on input from members, and ratified by a member vote no sooner than 3 years following completion of the refactoring.

# Why?

Secure future relevance of STAR as an industry B2B standard by:

1. Ensuring alignment with the latest major OAGIS release:
  - Improved support for Mobile/Cloud/REST API/JSON expansions of STAR.
  - Compliance with updated UN/CEFACT 3.0 definitions (incl. low level components, type definitions and code lists).

# Why?

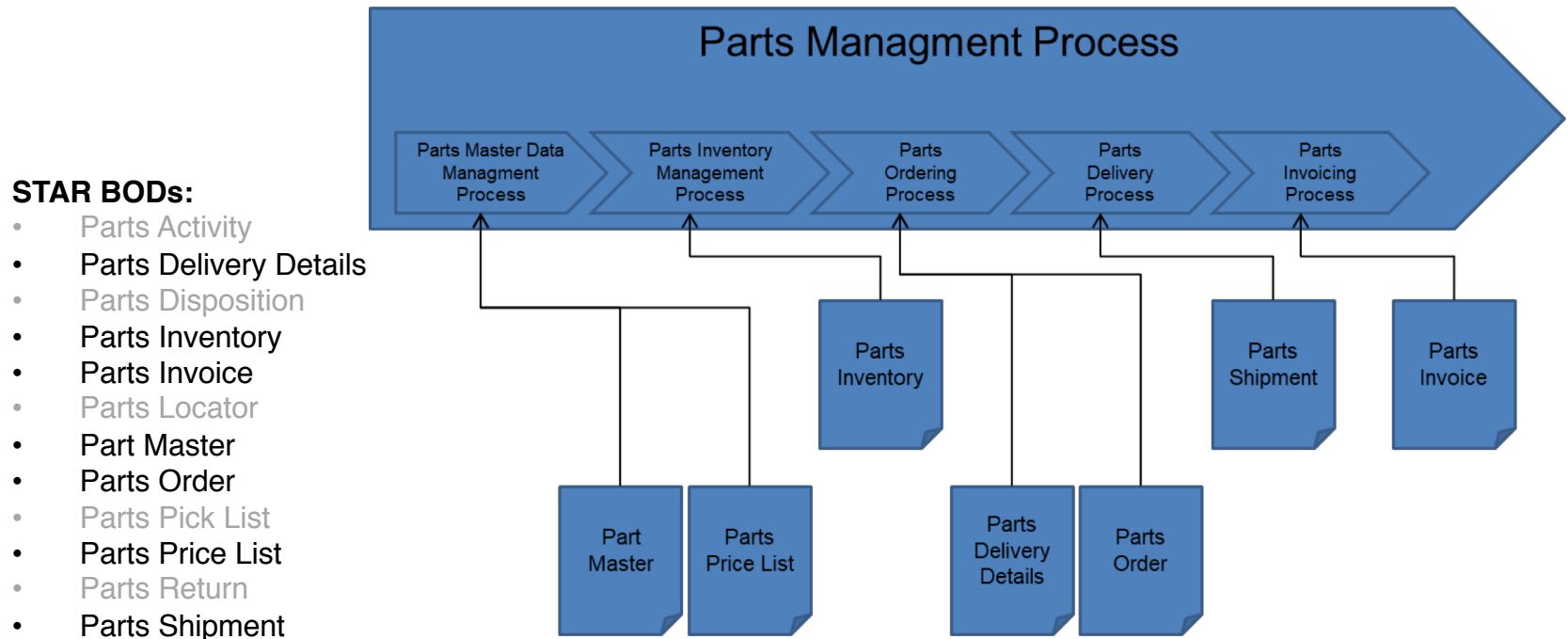
Secure future relevance of STAR as an industry B2B standard by:

1. Ensuring alignment with the latest major OAGIS release:
  - Improved support for Mobile/Cloud/REST API/JSON expansions of STAR.
  - Compliance with updated UN/CEFACT 3.0 definitions (incl. low level components, type definitions and code lists).
2. Cleaning up and harmonizing BODs to make STAR faster to implement and easier to use:
  - Consistency in naming and design rules (eg. OAGIS verb usage and language handling)
  - Increased reusability of components between BODs, eg.
    - naming of nouns (eg. prices in order, shipment and invoice)
    - consistent usage of proprietary elements vs. more general structures (name value pairs).
  - Removal of ambiguous and/or deprecated content, based on current members usage of existing BODs.

# Why? Example...

Parts invoice price matching with prices, tax, charges named and structured differently in the different BODs through out the Parts Management Process

Volvo CE Parts Ordering Use Case – Part Invoice Price Matching



# Why? Example...

## Example of inconsistencies - Prices

- **PartMaster**
  - PartMasterPricing element – PartMasterPricingType
  - CoreChargeAmount element – udt:AmountType
- **PartsPriceList**
  - Price element – PriceABEType
  - Tax element - TaxType
- **PartsInventory**
  - UnitPriceAmount element - udt:AmountType
  - CorePriceAmount element - udt:AmountType
  - AveragePriceCost element - udt:AmountType
- **Parts Order**
  - PartsOrderHeader
    - SubtotalAmount element - udt:AmountType
    - EstimatedFreightCostAmount element - udt:AmountType
    - HandlingAmount element - udt:AmountType
    - TotalAmountLessTax element - udt:AmountType
    - Price element – PriceABEType
  - PartsOrderLine
    - Price element – PriceABEType

## PartsShipment

### PartsShipmentHeader

- AccessoriesDiscountAmount element - udt:AmountType
- OtherDiscountAmount element - udt:AmountType
- TotalDiscountAmount element - udt:AmountType
- Tax element – Taxtype
- Total Amount element - udt:AmountType
- TotalFreightChargeAmount element - udt:AmountType
- TotalHandlingAmount element - udt:AmountType

### PartsShipmentLine

- Price element - PriceABEType
- Tax element - TaxType
- ExtendedCoreAmount element - udt:AmountType
- CoreUnitAmount element - udt:AmountType
- FreightChargeAmount element - udt:AmountType
- HandlingAmount element - udt:AmountType



# Why? Example...

Example of inconsistencies - Prices

- **PartsInvoice**

PartsInvoiceHeader

- stockOrderNetAmount element - udt:AmountType
- StockAdjustmentValueAmount element - udt:AmountType
- TotalPartsAmount element - udt:AmountType
- TotalAccessoriesAmount element - udt:AmountType
- TotalOtherAmount element - udt:AmountType
- SubtotalBeforeDiscountAmount element - udt:AmountType
- PartsDiscountAmount element - udt:AmountType
- OtherDiscountAmount element - udt:AmountType
- TotalDiscountAmount element - udt:AmountType
- SuntotalIncludingDiscountAmount element - udt:AmountType
- ExtendedAmount element - udt:AmountType
- Allowance element – AllowanceType
- TotalMiscellaneousExpense element - udt:AmountType
- Charges element – ChargesType
- Tax element – TaxType
- Price element - PriceABEType

# Why?

Secure future relevance of STAR as an industry B2B standard through:

1. Ensuring alignment with the latest major OAGIS release:
  - Improved support for Mobile/Cloud/REST API/JSON expansions of STAR.
  - Compliance with updated UN/CEFACT 3.0 definitions (incl. low level components, type definitions and code lists).
2. Cleaning up and harmonizing BODs to make STAR faster to implement and easier to use:
  - Consistency in naming and design rules (eg. OAGIS verb usage and language handling)
  - Increased reusability of components between BODs, eg.
    - naming of nouns (eg. prices in order, shipment and invoice)
    - consistent usage of proprietary elements vs. more general structures (name value pairs).
  - Removal of ambiguous and/or deprecated content, based on current members usage of existing BODs.
3. Incentivizing former members to re-engage with STAR.

# How? Refactoring Process

- Refactoring to be performed by a core Refactoring Workgroup, supported by the STAR Enterprise Data Architect, in two steps:
  1. Identify and refactor common components (e.g. Vehicle)
  2. Refactor BODs based on the following priority:
    - Requests from members for refactoring a specific BOD
    - Refactor more commonly used BODs before less used BODs
- All members are invited to participate in the Refactoring Workgroup (target profile data/information architects with experience working with STAR implementations),
- Refactored BODs will be made available for members to review to identify gaps in relation to their implementations and provide feedback to the Refactoring Workgroup.
- Refactoring guiding principles:
  - Consistency in naming and design rules
  - Common components across BODs for common objects
  - Consistency in usage of proprietary elements vs. general structures
  - Removal of ambiguous and deprecated content
  - **Leading to no backwards compatibility between STAR 6 and STAR 5**

# How? Review Process

Approval process for refactored common components and BODs:

1. Signoff of all members in the Refactoring Workgroup (incl. STAR Enterprise Data Architect)
  - Common components will be signed off in group, BODs individually.
2. A request for review will be sent to all current STAR members, who will have **60 days** to provide comments on the refactored group of common components or BOD.
  - If needed, the STAR Enterprise Data Architect will reach out to the member for clarification in case of unclear comments.
  - The STAR Enterprise Data Architect will provide recommendations to the Refactoring Workgroup based on comments provided by members.
3. After **60 days** the Refactoring Workgroup will consider all comments from members and release a final draft version of the common component and/or refactored BOD to all members.
4. The refactored BOD will be included in next upcoming yearly release of STAR on the following 4<sup>th</sup> of July.
5. Once published, the same change request process for STAR 5 BODs will apply to STAR 6 BODs for members.

# Next Steps

- **STAR to distribute a voting package to all members following this meeting.**
- **Members invited to participate on two levels:**
  - 1) Provide direct support by participating in the Refactoring Workgroup (target profile data/information architects with experience working with STAR implementations).
  - 2) Review refactored components or BODs and provide feedback to the Refactoring Workgroup.

# Questions, comments or concerns?

If later, please submit to Paco Escobar at [pescobar@starstandard.org](mailto:pescobar@starstandard.org)

Questions and answers will be posted on the [refactoring Q&A section of the STAR web site](#)