

30 March 2023

Opus Suite
Conference

SYDNEY

Technical Deep Dive: Benefits with the latest data model of Opus Suite

Ahmed Hatem
Systecon AB

by Systecon
opus
suite

 **Systecon**

Agenda

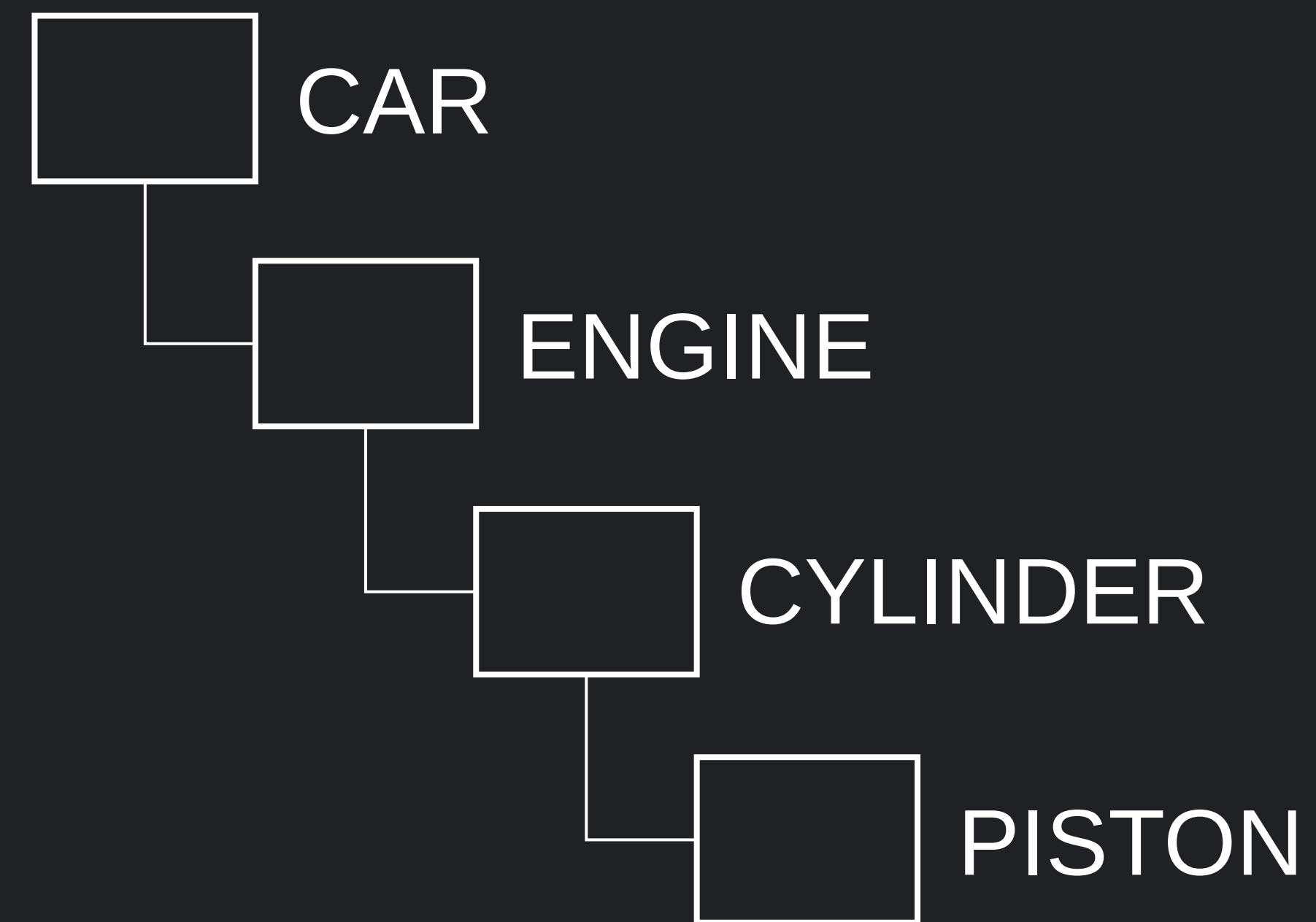
Opus Suite
Conference

SYDNEY

- 1 Product breakdown and replace tasks**
- 2 Maintenance modeling**
- 3 Utilization and failure intensity per position**
- 4 Maintenance without replacements**
- 5 Scheduled maintenance - PM**
- 6 Subitems replaced directly from system**
- 7 Flexible task duration**

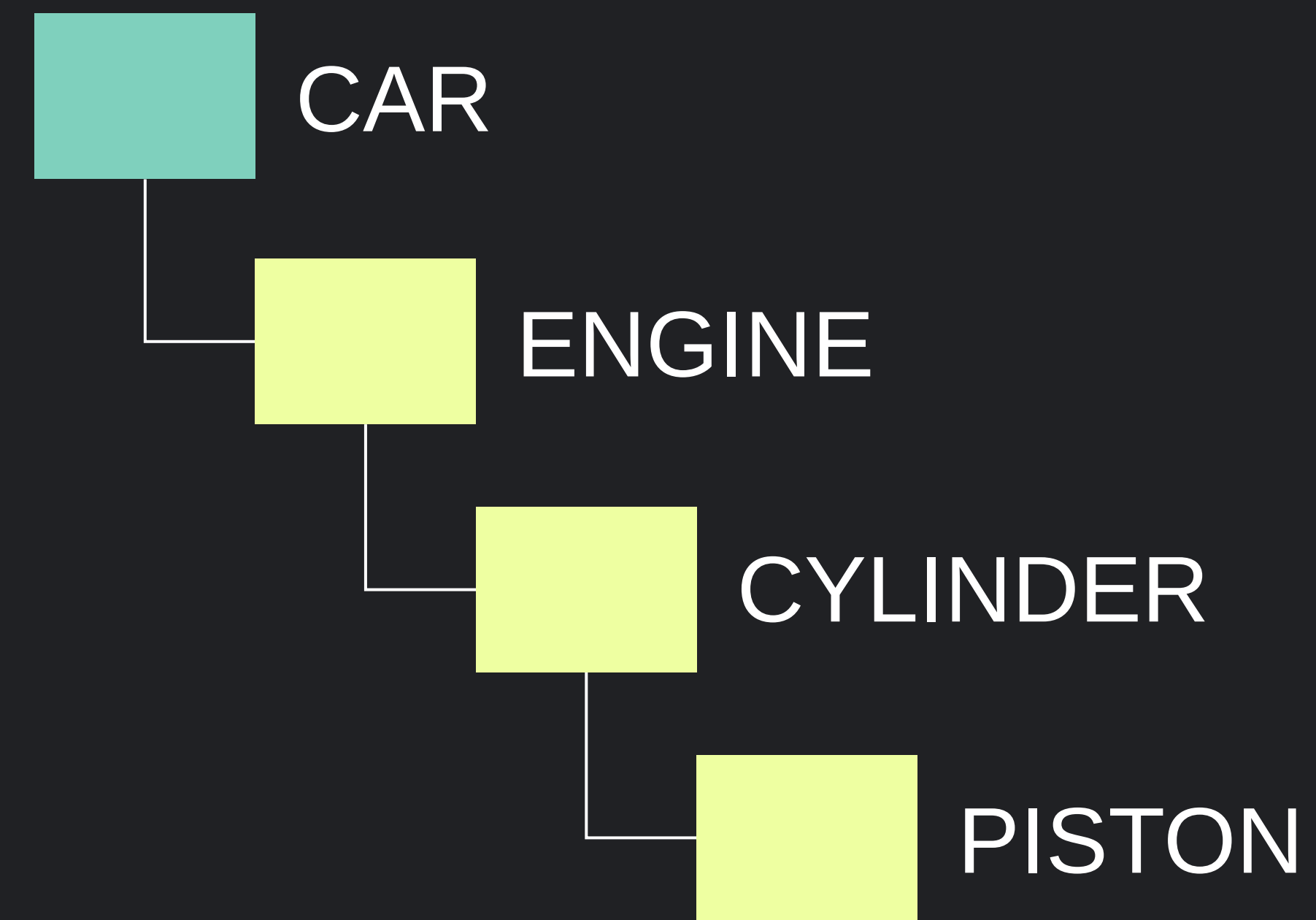
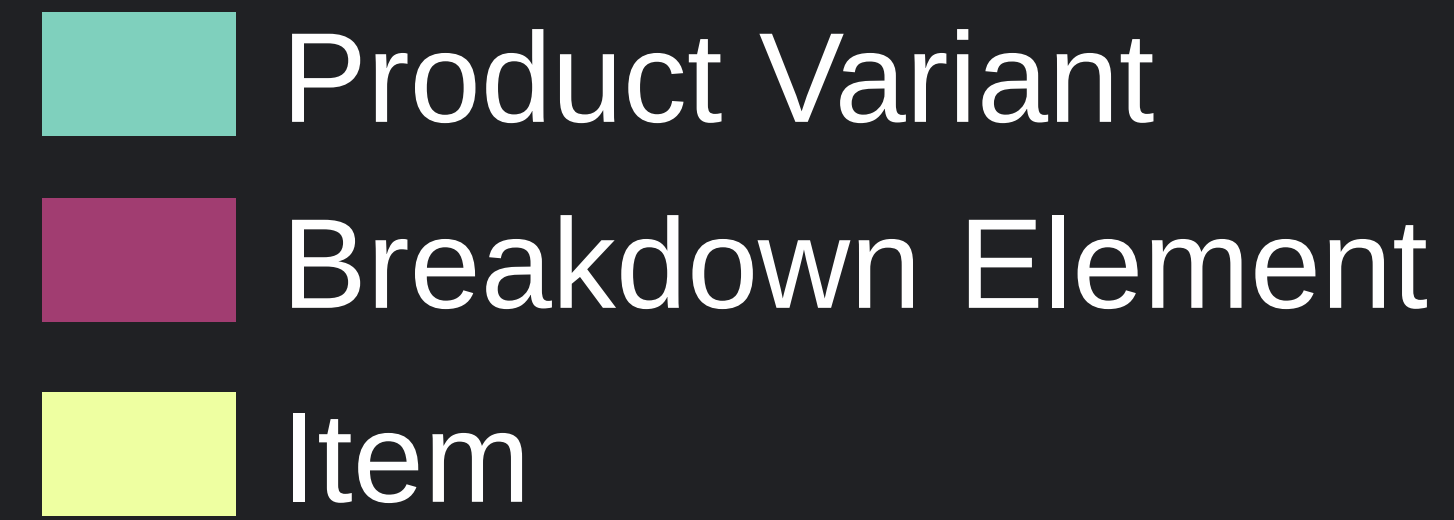
Product breakdown and replace tasks

- Product Variant
- Breakdown Element
- Item



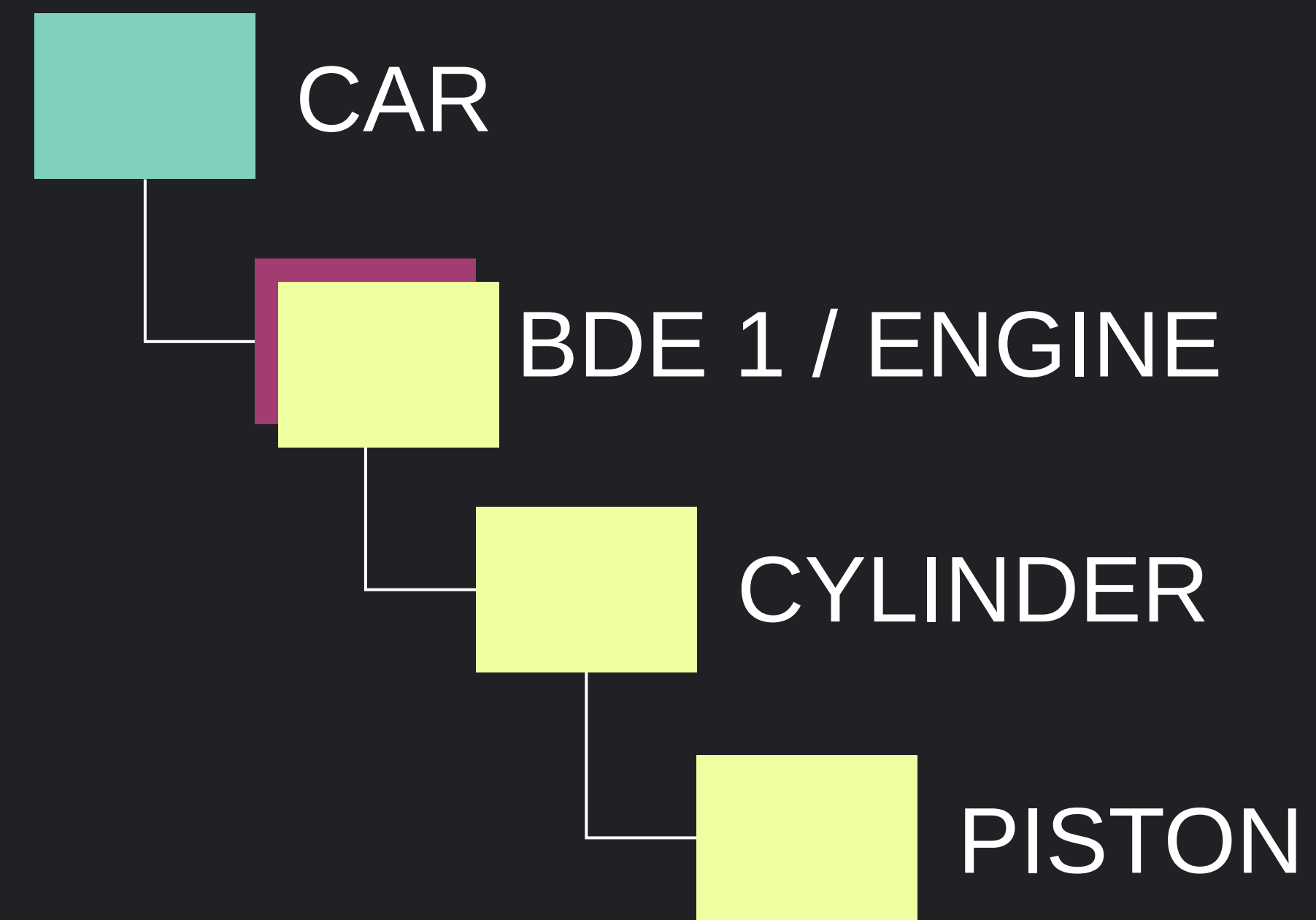
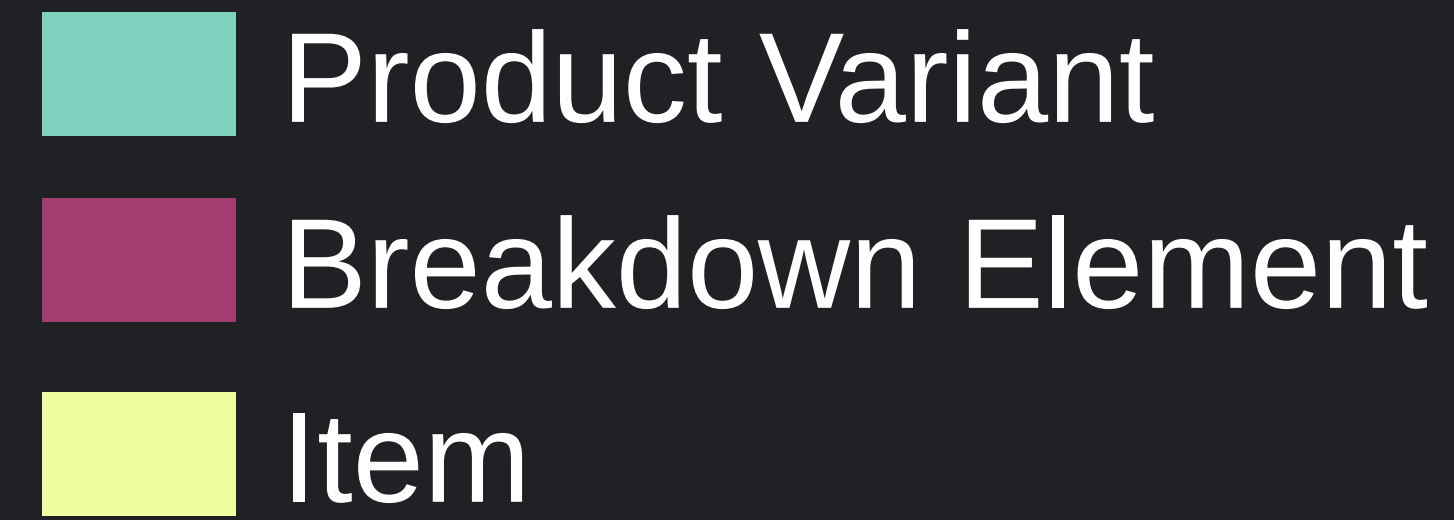
Product breakdown and replace tasks

- Optional no of level(s) with breakdown elements



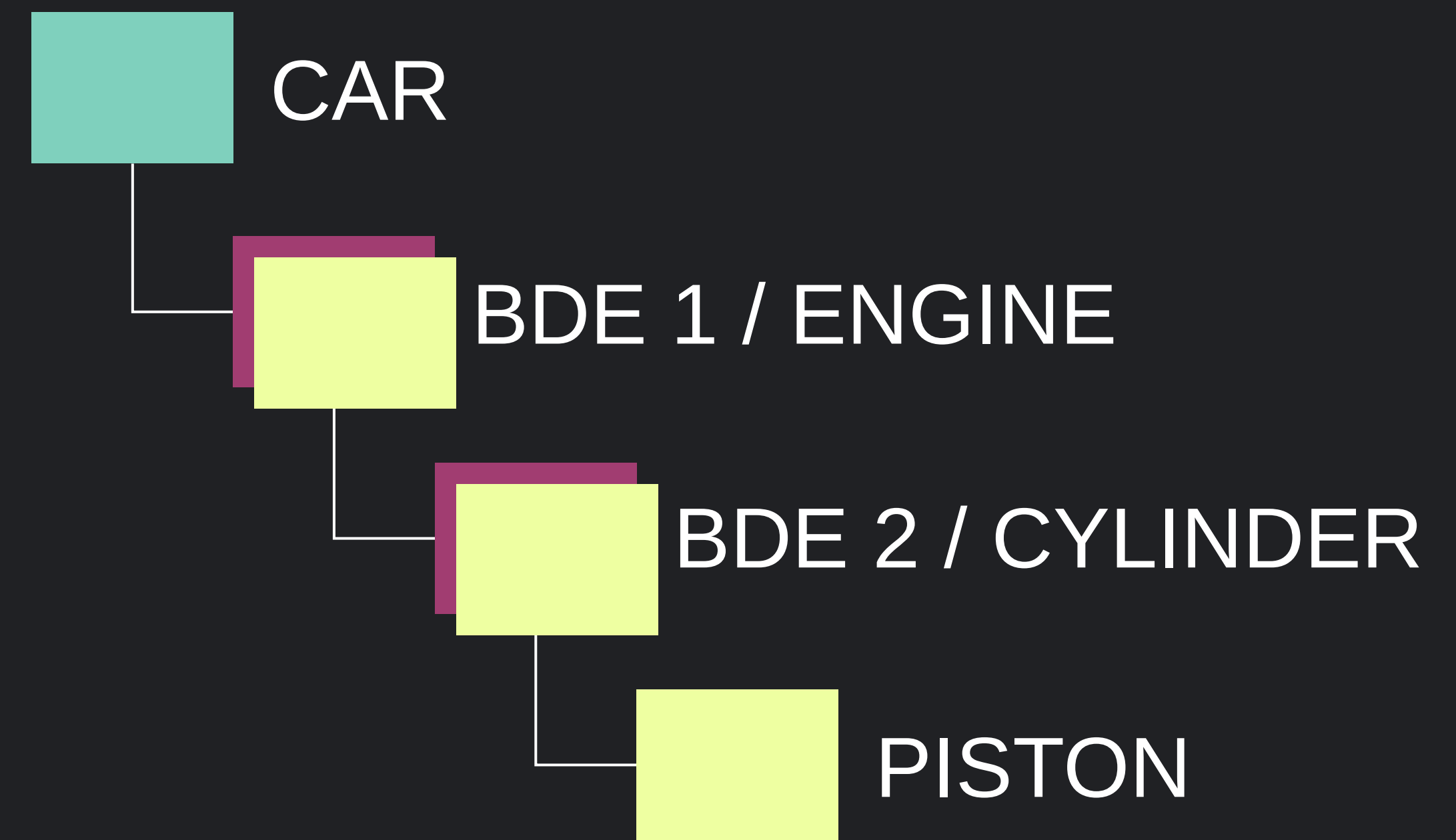
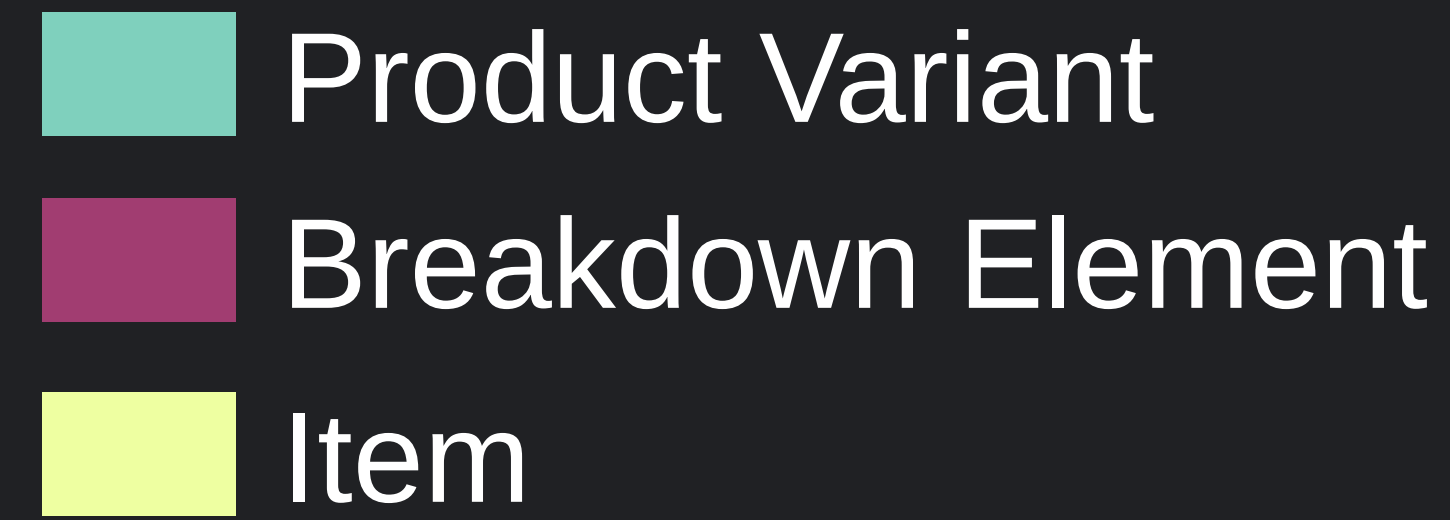
Product breakdown and replace tasks

- Optional no of level(s) with breakdown elements



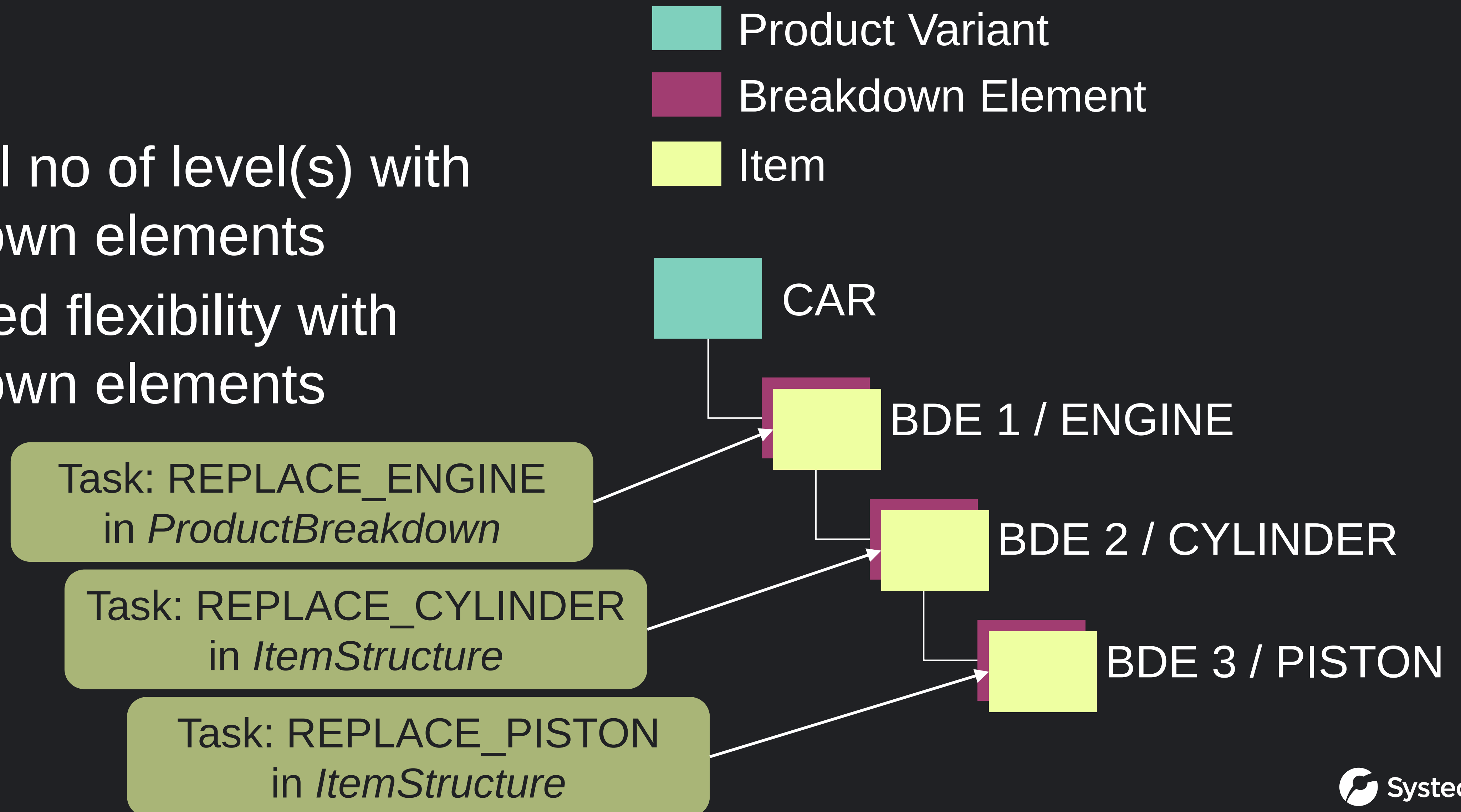
Product breakdown and replace tasks

- Optional no of level(s) with breakdown elements



Product breakdown and replace tasks

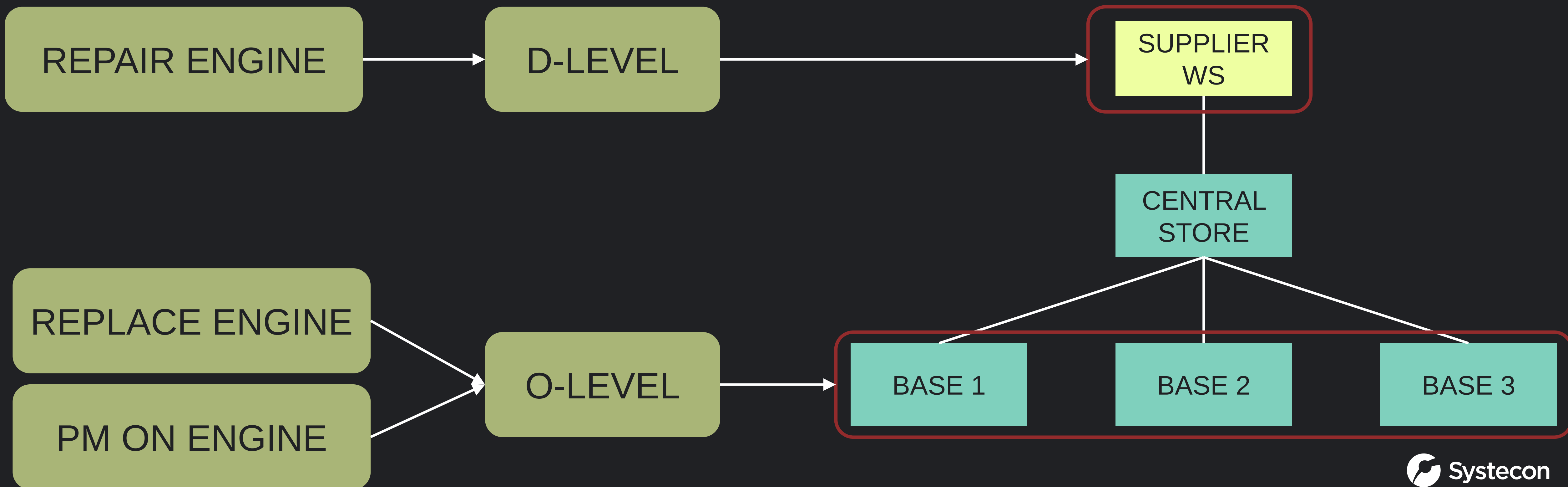
- Optional no of level(s) with breakdown elements
- Increased flexibility with breakdown elements



Maintenance modeling

1. Maintenance Capability

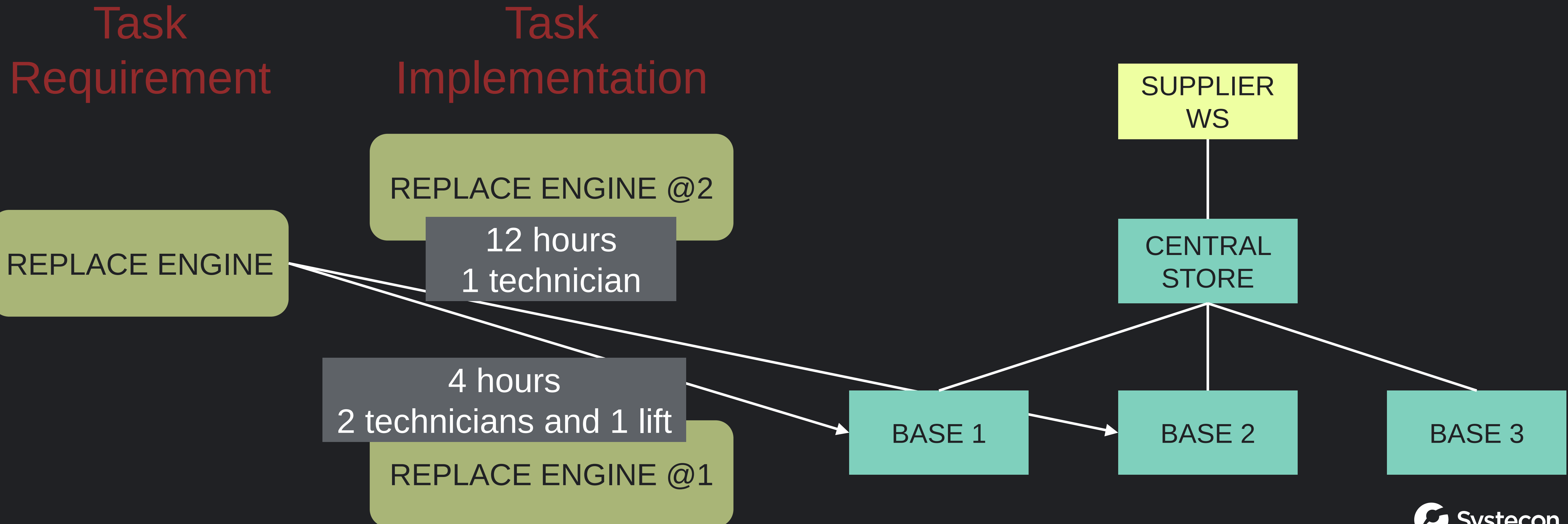
- Covers the most common types of maintenance modeling
- Builds on maintenance levels



Maintenance modeling

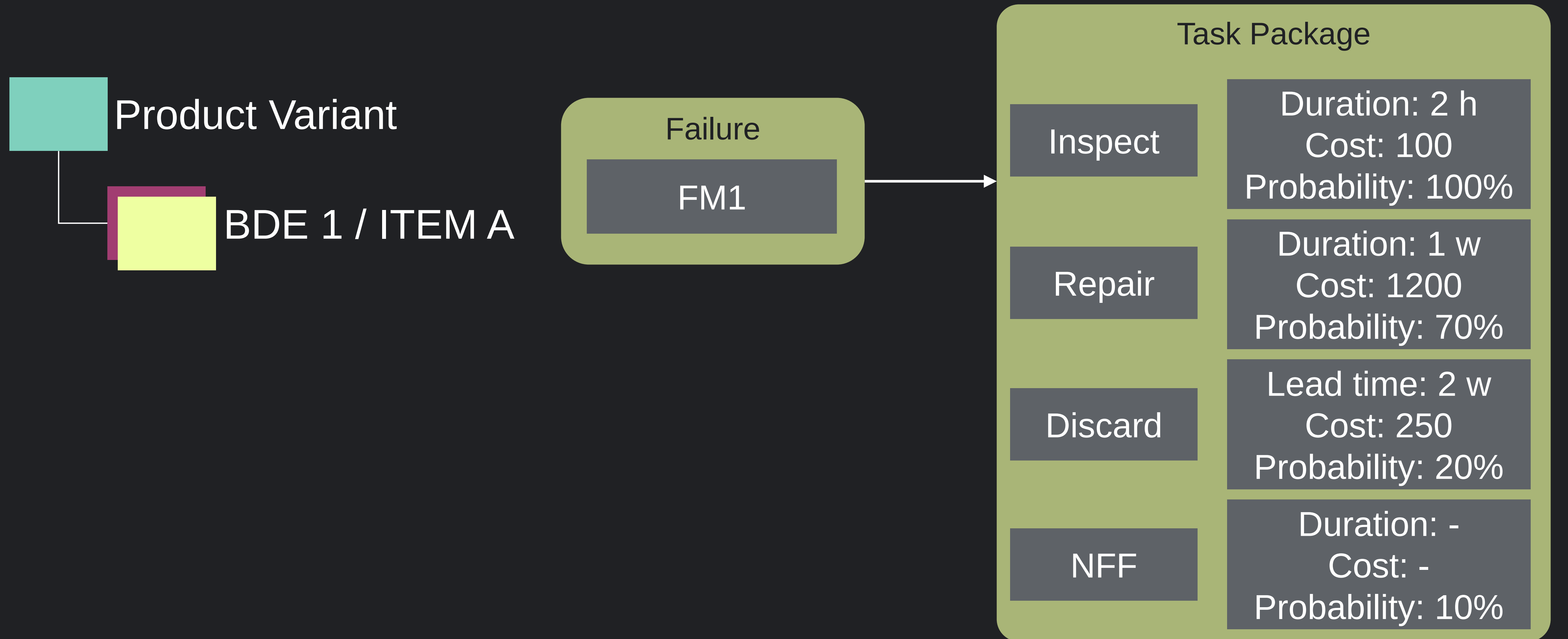
2. Task Implementation

- Handles detailed maintenance modeling



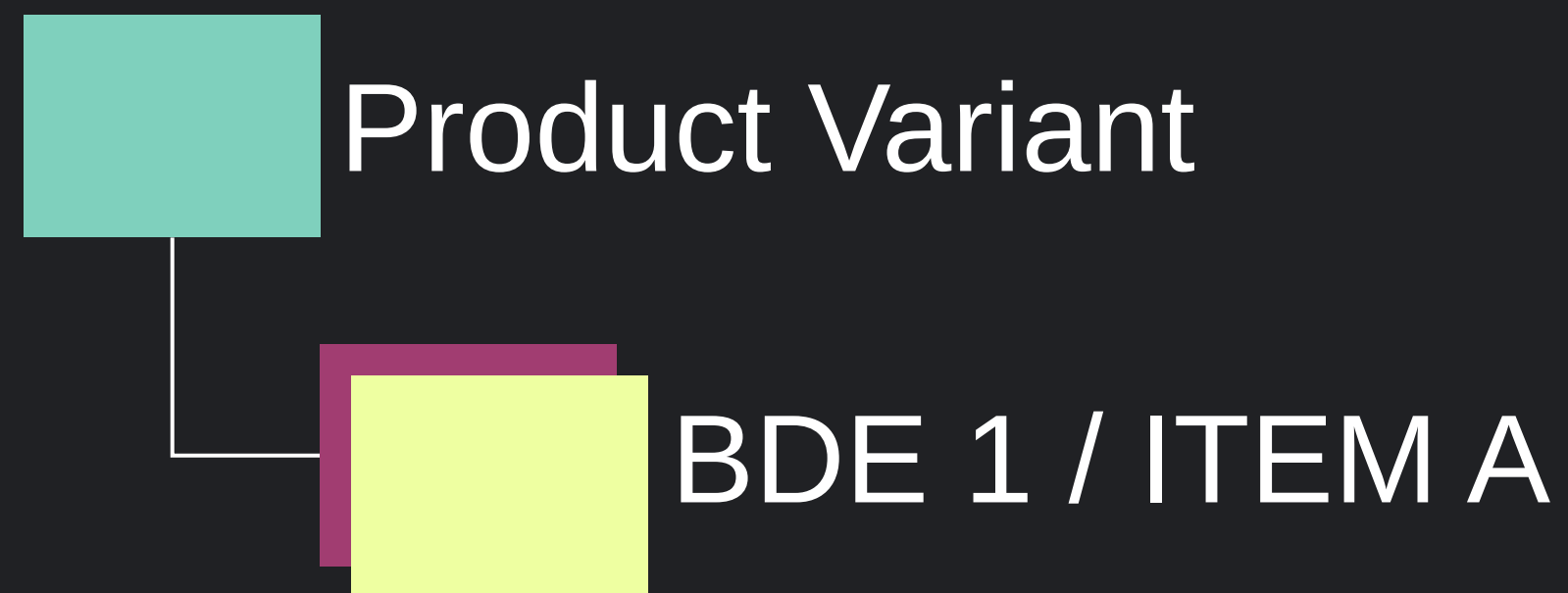
Maintenance modeling

3. Task Package - Multiple tasks for single event

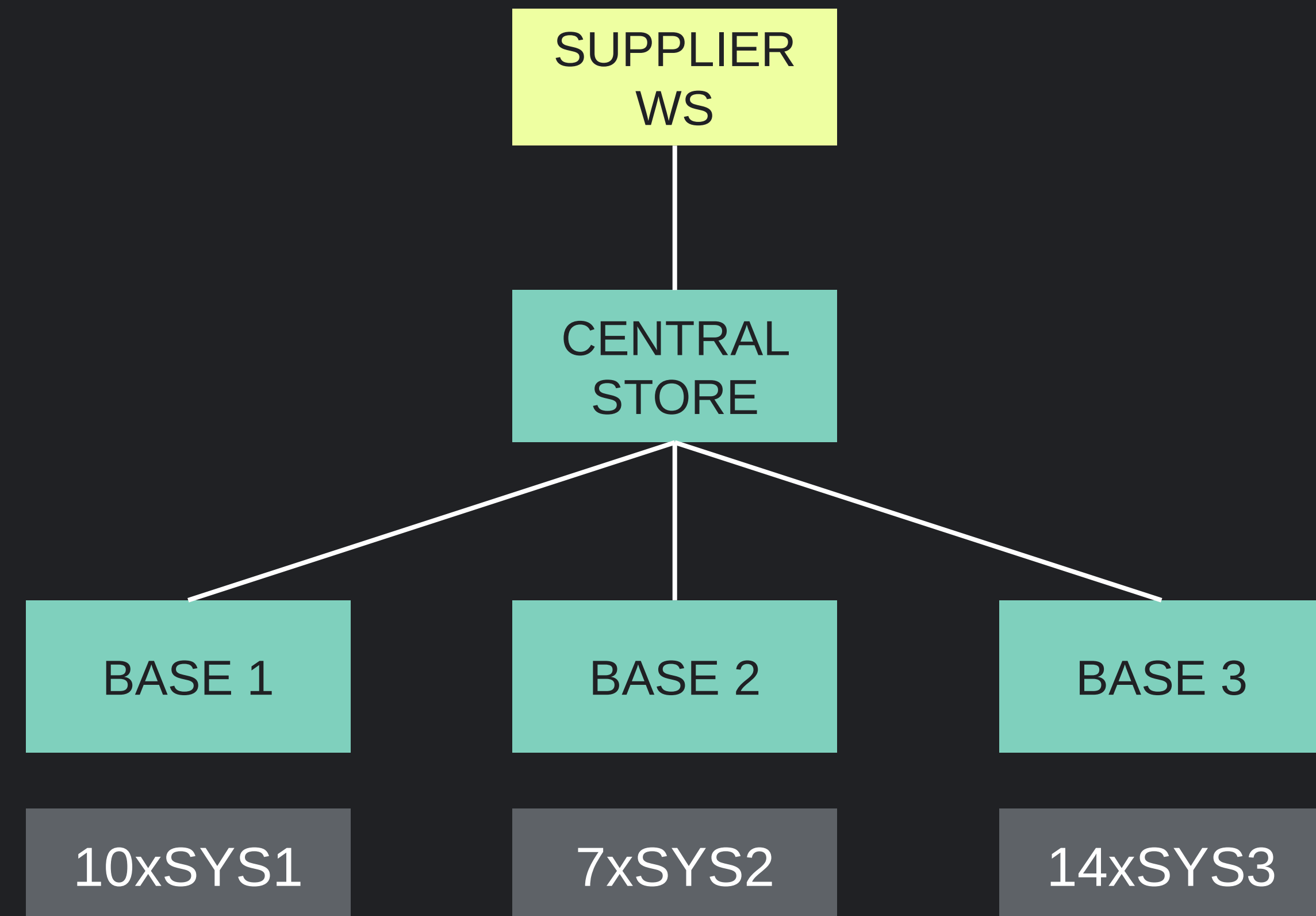


Utilization per breakdown element

- Describe how a component is utilized in separate systems

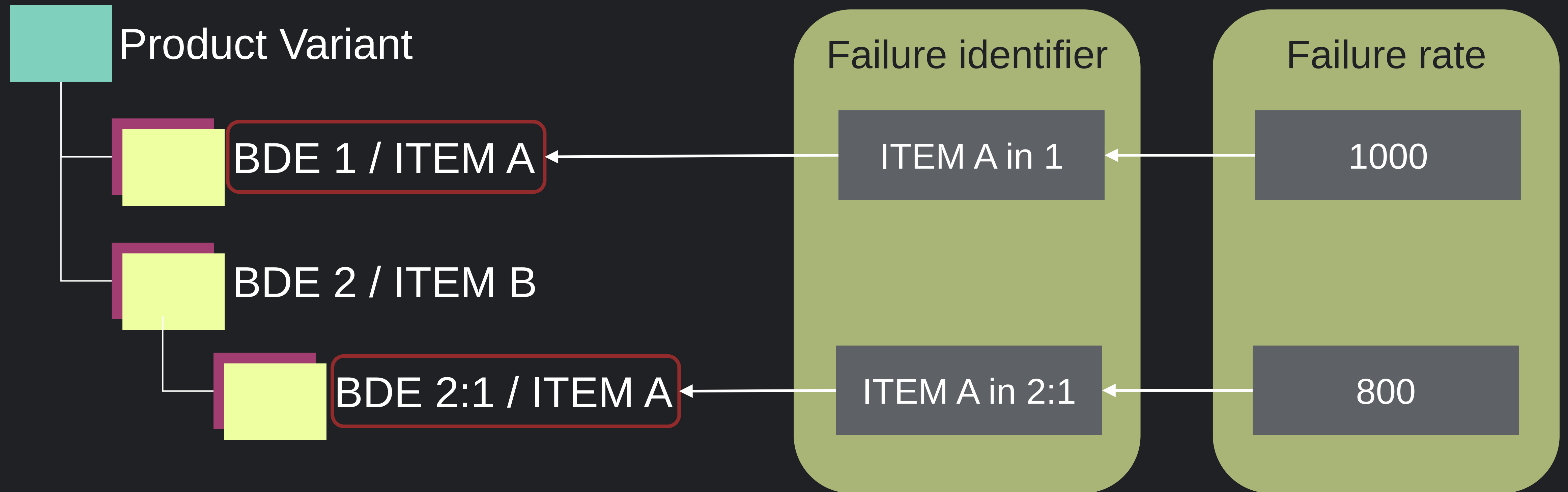


Breakdown Element Usage Rate		
System identifier	Breakdown element	Usage rate
SYS1	BDE 1	1.0
SYS2	BDE 1	0.3
SYS3	BDE 1	0.7



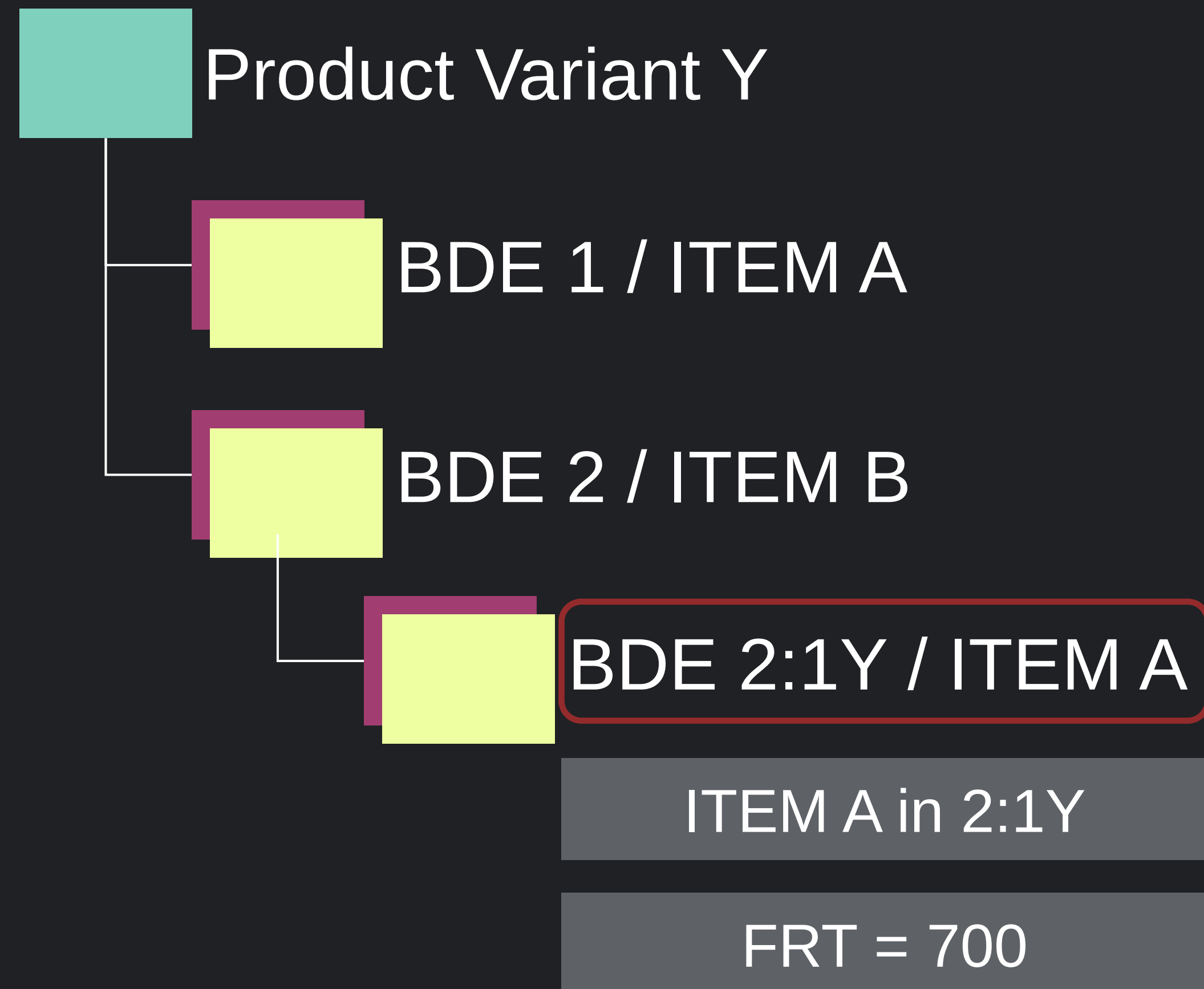
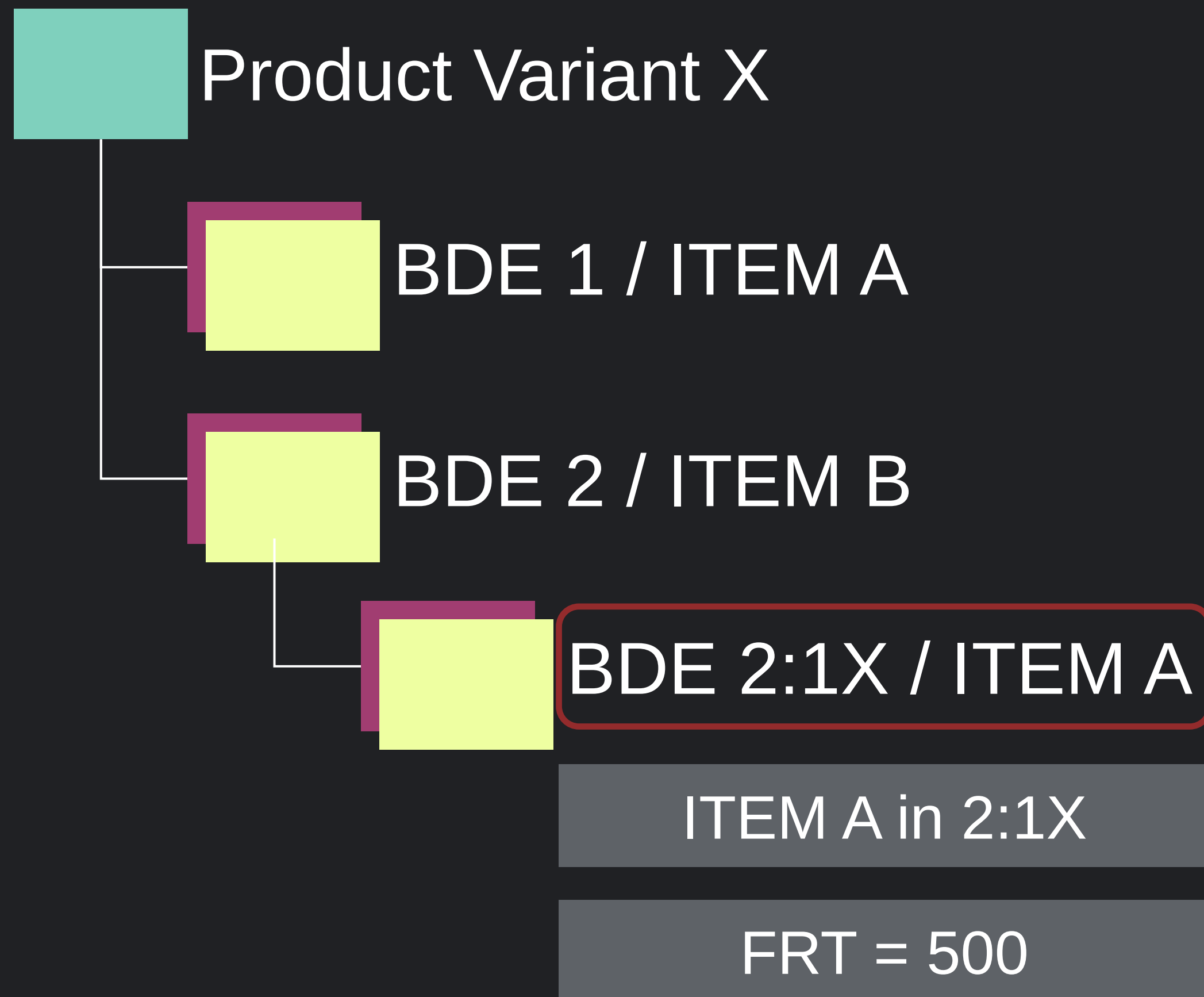
Failure rate per breakdown element

- Handle situations where the FRT differs for an item *within a variant*



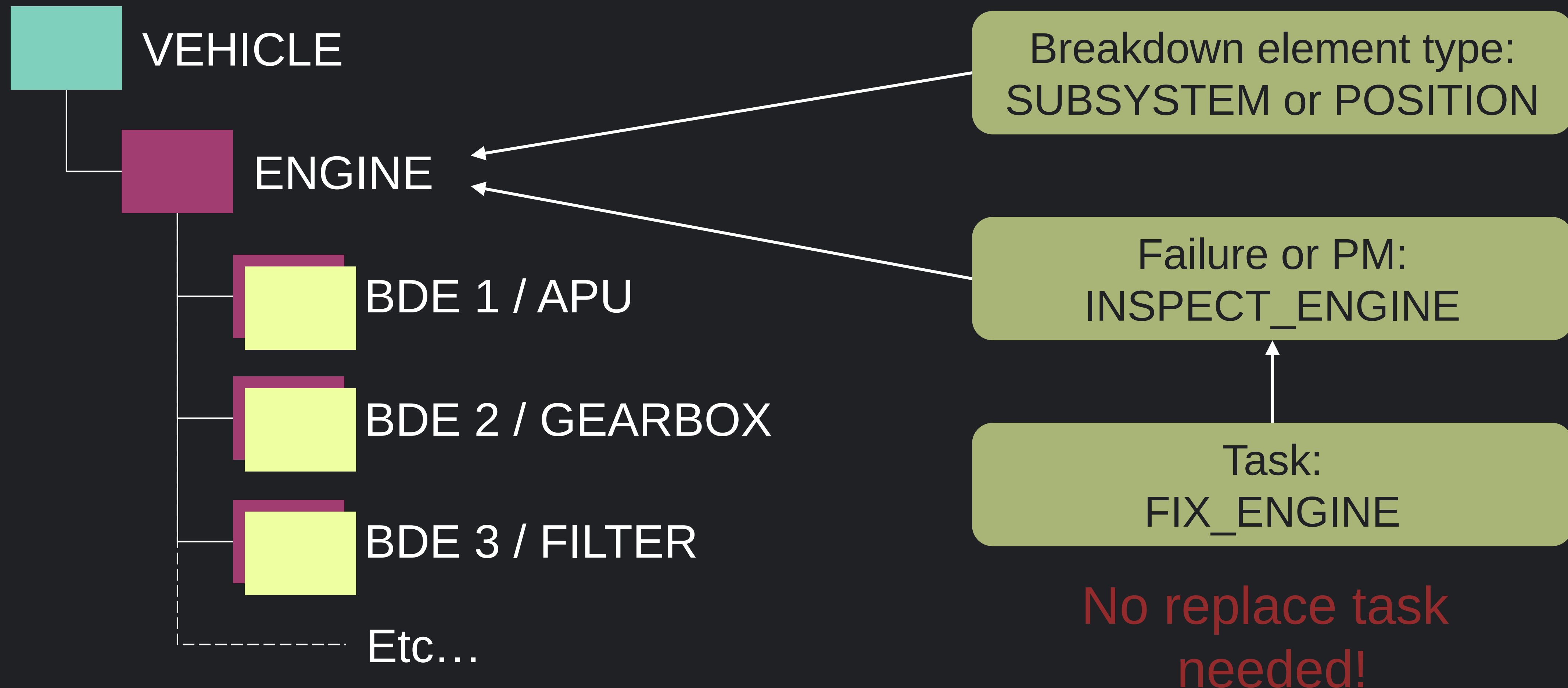
Failure rate per breakdown element

- Handle situations where the FRT differs for an item *between variants*



Maintenance without replacements

- Events for breakdown elements without realization



Scheduled maintenance - PM

- Extensive user documentation available

The screenshot displays the OPUS10 user documentation interface. The window title is 'OPUS10'. The top navigation bar includes icons for 'Hide', 'Back', 'Forward', 'Home', 'Print', and 'Options'. Below this is a tabbed interface with 'Contents', 'Index', 'Search', and 'Favorites'. The 'Contents' tab is active, showing a hierarchical tree of documentation topics. The 'Maintenance' section is expanded, and 'Maintenance - General overview' is selected. The main content area on the right displays the 'Maintenance - General overview' page, which includes a 'Top Previous Next' navigation bar. The page text describes the process of maintenance after an event, mentioning tables like [Repair](#), [PMSchedule](#), [overview](#), [breakdown element](#), [BDEID](#), [Failure](#), and [PMPlan](#). It also discusses task requirements, task implementation, task packages, and maintenance activities, with references to tables like [TaskResource](#), [TaskPackageContent](#), and [MaintenanceActivityContent](#). A note at the bottom states that modeling examples can be found [here](#).

Maintenance - General overview [Top](#) [Previous](#) [Next](#)

After an [event](#) has occurred, the [system](#) is typically rendered inoperable. To get the system up and running again, some kind of maintenance must be performed. What action to take to handle the event is specified in the table [Repair](#) (for corrective maintenance) and the table [PMSchedule](#) (for preventive maintenance). When maintenance is to be performed on an [item](#) it needs to be removed from the system or the parent item. A more detailed description of replacements is given in the [overview](#). When maintenance is performed on a [breakdown element](#) of SUBSYSTEM or POSITION type it is required that failure / pm event triggering the maintenance is connected to the same breakdown element ([BDEID](#) column in tables [Failure](#) and [PMPlan](#)).

Typically a maintenance activity gives rise to a need to perform a [task](#) (the task *requirement* in the Opus Suite maintenance model). How a task is performed (the task *implementation*) can depend on the component and the [capability](#) of the maintaining station. A task is an action with limited scope and it should be viewed as a concrete instruction/assignment that can be performed by an individual or equipment in the repair facility, e.g., "run failure detection" or "clean cylinder". In some cases a task is dependent on a resource, such as some kind of test equipment. The dependency on resources is given in the table [TaskResource](#). A task can be divided into [subtasks](#) and each subtask can also require [resources](#).

In some cases the event results in a sequence of tasks, all of which should be performed on the [same component](#). This type of situation is handled using a [task package](#) and the tasks in the task package are specified in the table [TaskPackageContent](#).

In the case the event results in a sequence of tasks that should be performed on *more than one* [component](#) a [maintenance activity](#) should be used. A maintenance activity can consist of a task, a task package or another maintenance activity, all according to the data given in the table [MaintenanceActivityContent](#).

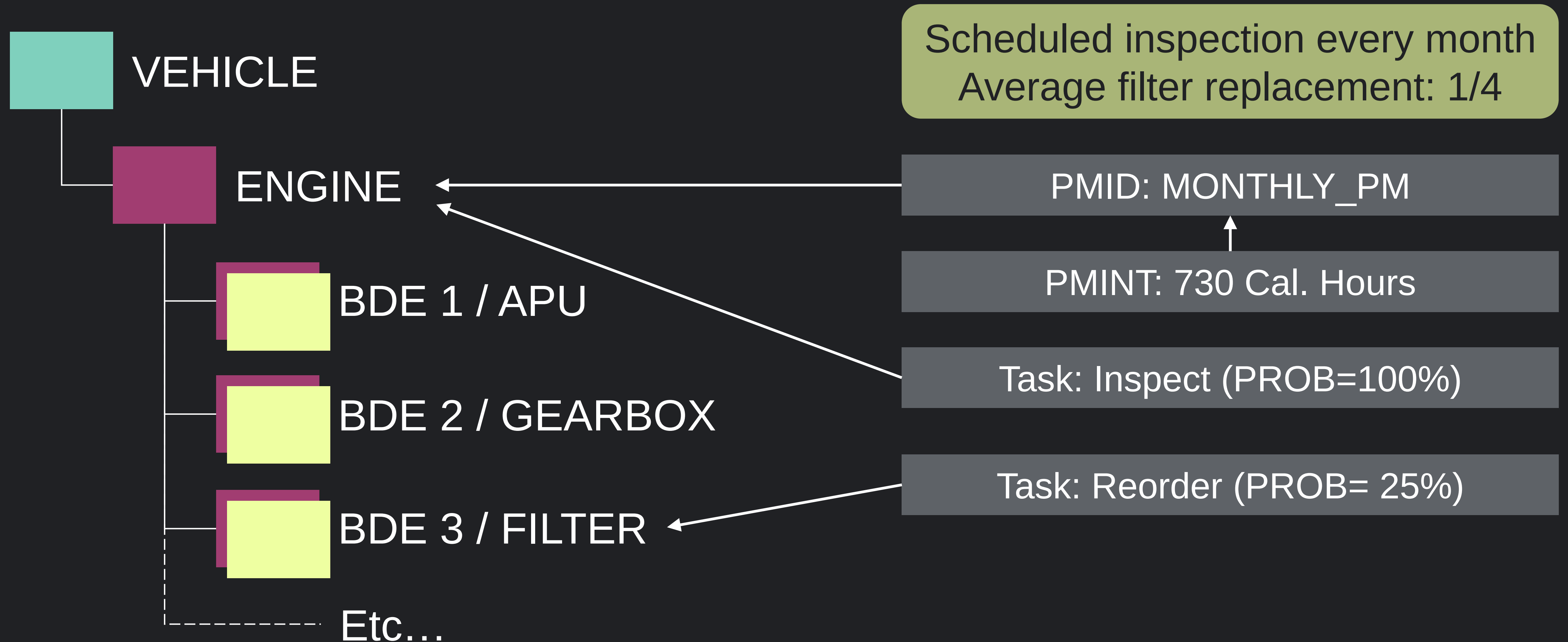
Note that there are some restrictions on what can be included in a maintenance activity. First of all, each activity must include at least one explicit task, i.e. it is not allowed to have an activity that solely links to other maintenance activities. This can be accomplished by including either a single task, or a task package. Second, the maintenance activity must have at least one task associated with the failed component or item below it. It is possible to add other tasks performed on sub-components of the failed component and/or tasks on ancestors of the failed component, such as fault localization on the parent component. It is, however, not allowed to perform tasks on "siblings" (that is components on the same indenture level as the failed component). Moreover, for each event there can only be *one* way to rectify the event. This means that it is not allowed to link another maintenance activity to the same component for the same event.

In the case of corrective maintenance the event is handled using either a [Task](#), a [TaskPackage](#) or a [MaintenanceActivity](#) while in the case of preventive maintenance the event needs to be handled by a [MaintenanceActivity](#).

Modeling examples can be found [here](#).

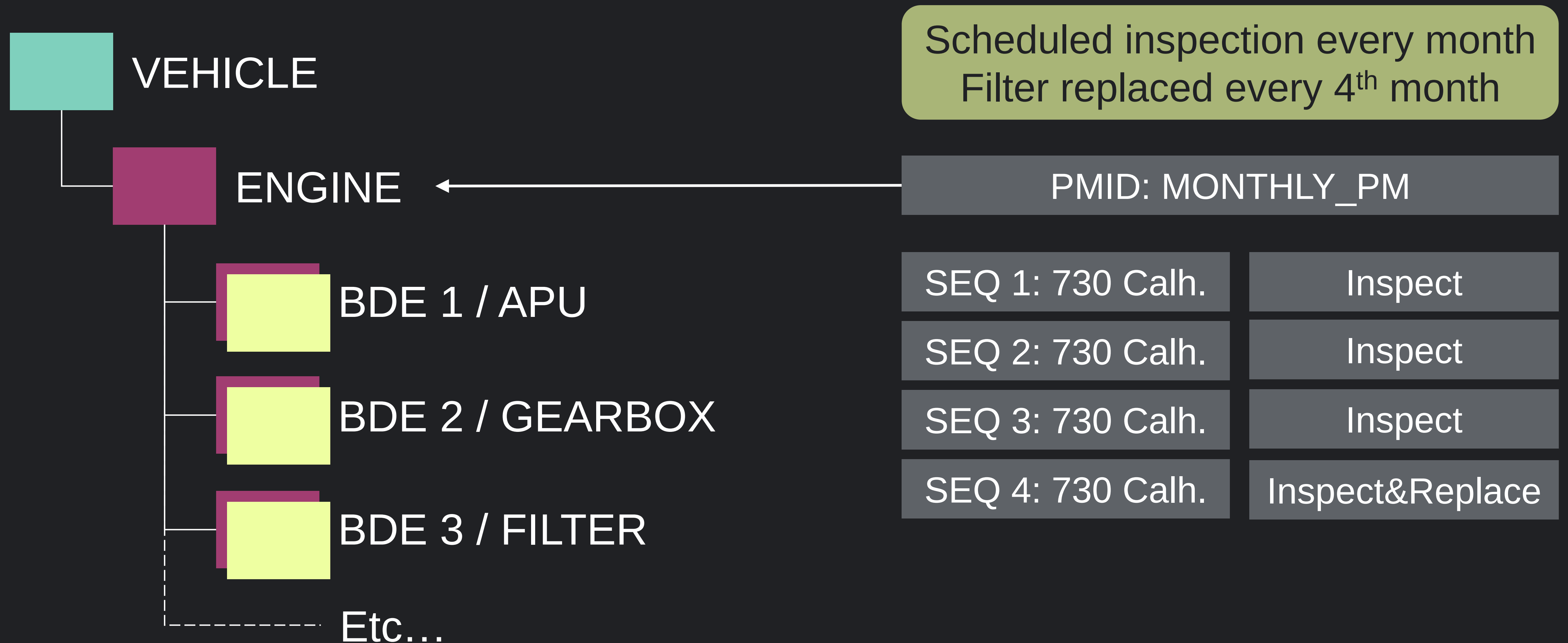
Scheduled maintenance - PM

- Random item replacements

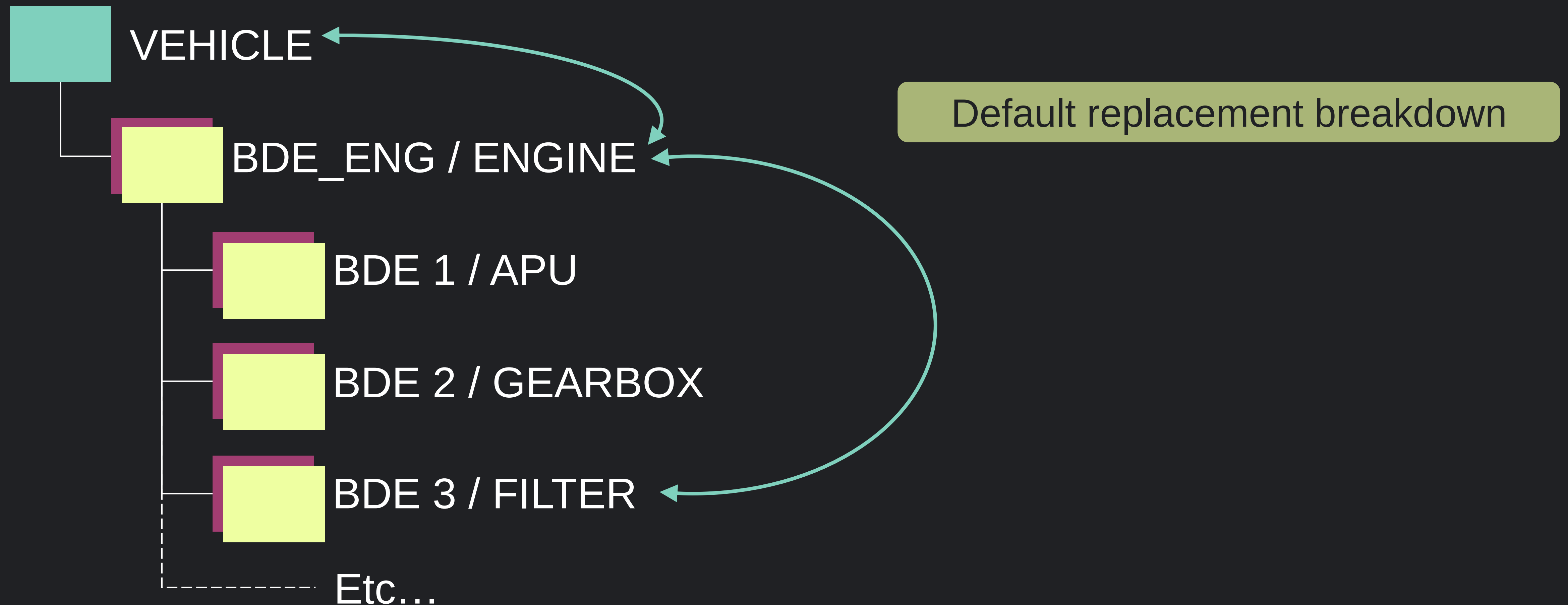


Scheduled maintenance - PM

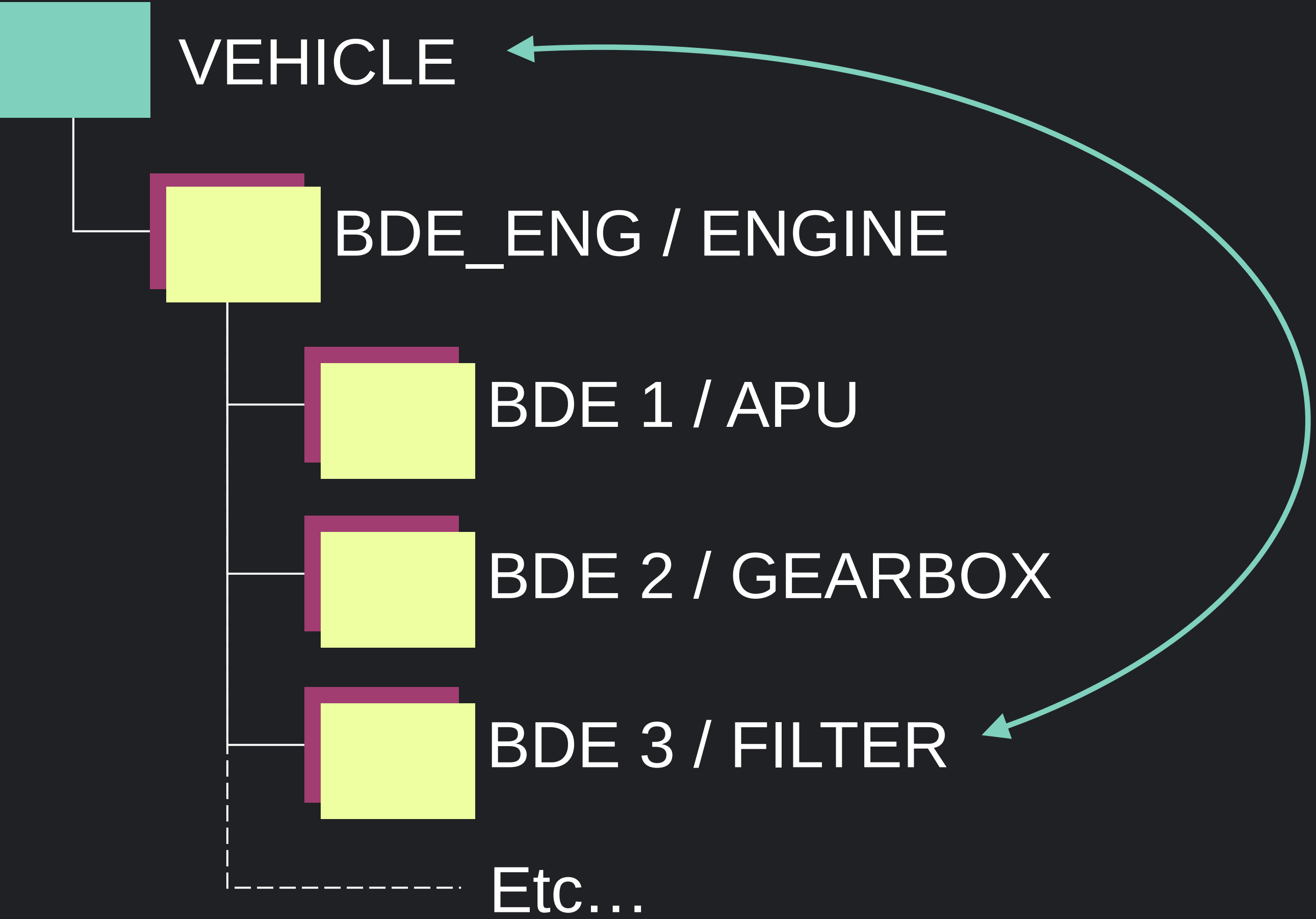
- Scheduled item replacements



Subitems replaced directly from system

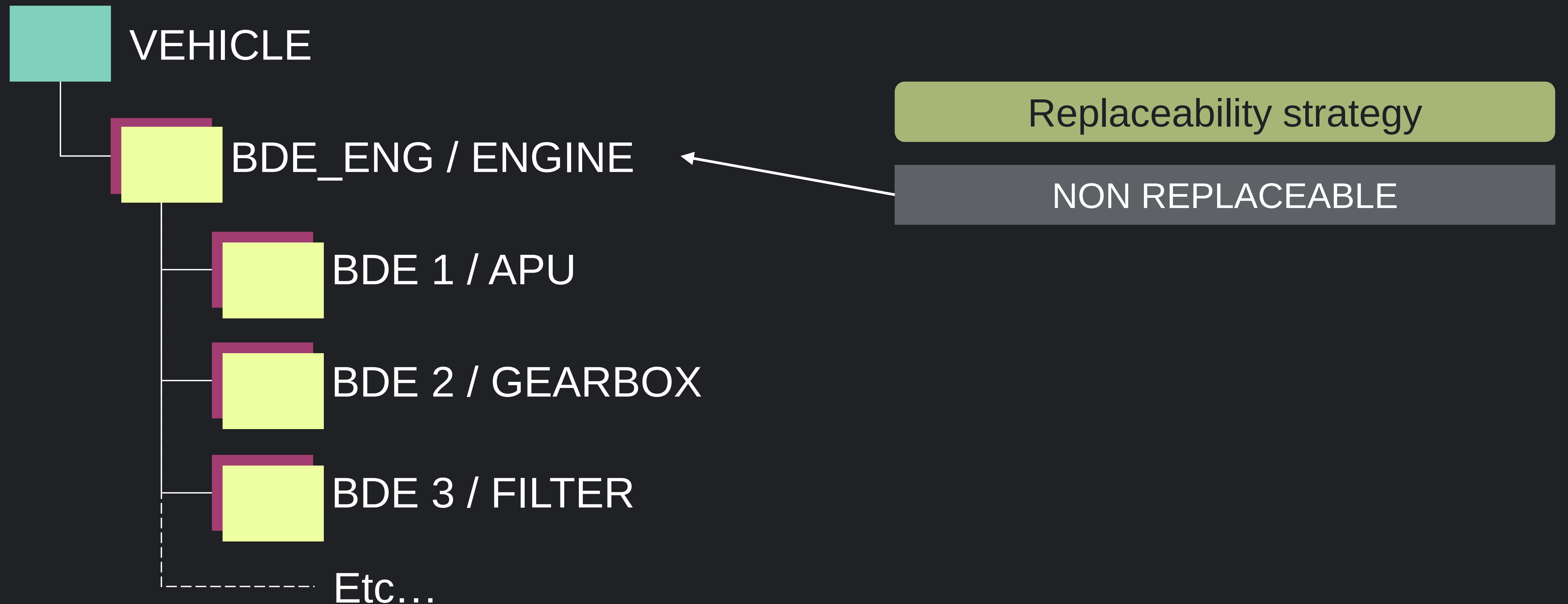


Subitems replaced directly from system

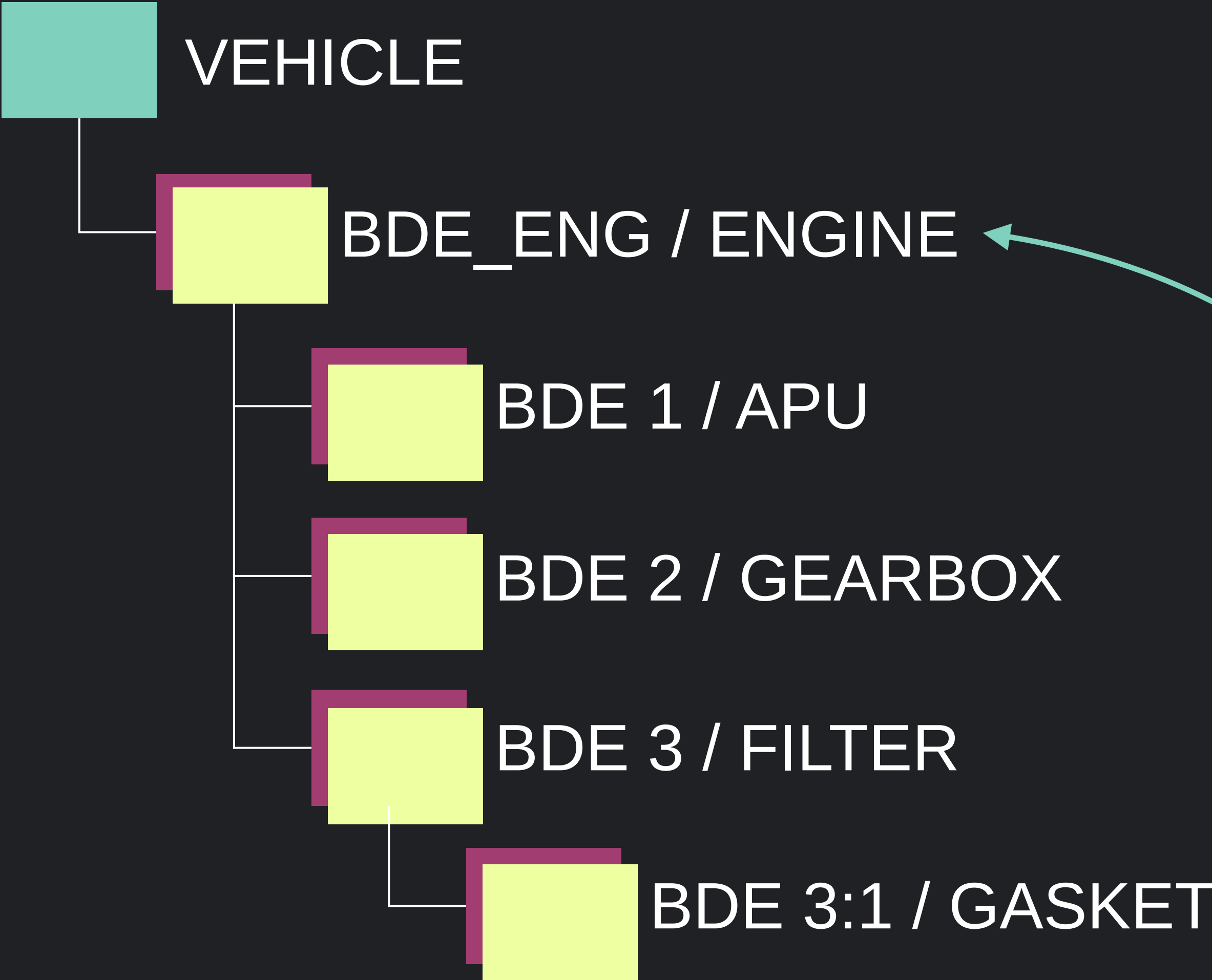


Product replacement breakdown	
Product Variant	Breakdown element
VEHICLE	BDE 3

Subitems replaced directly from system



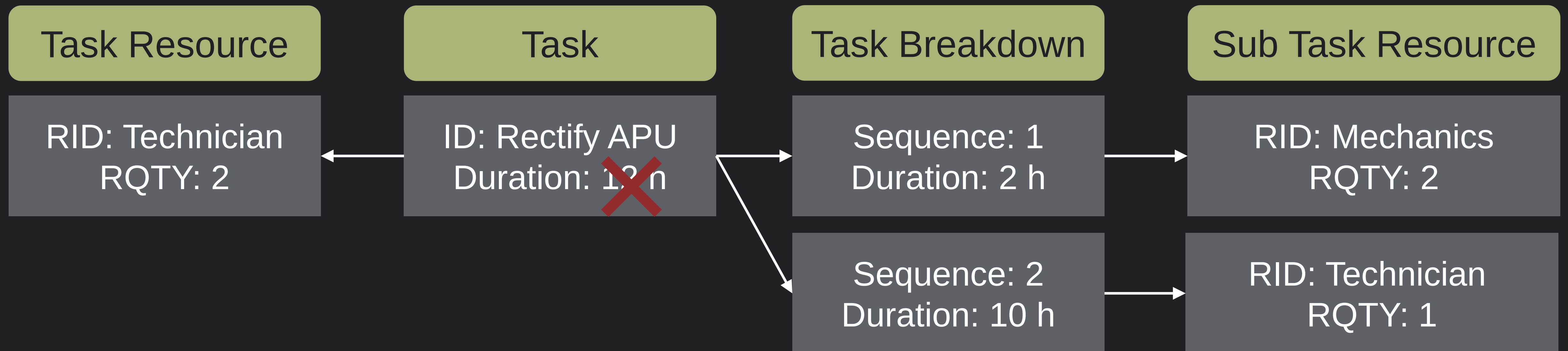
Subitems replaced directly from parent



Item replacement breakdown	
Sub item identifier	Mother item Identifier
GASKET	ENGINE

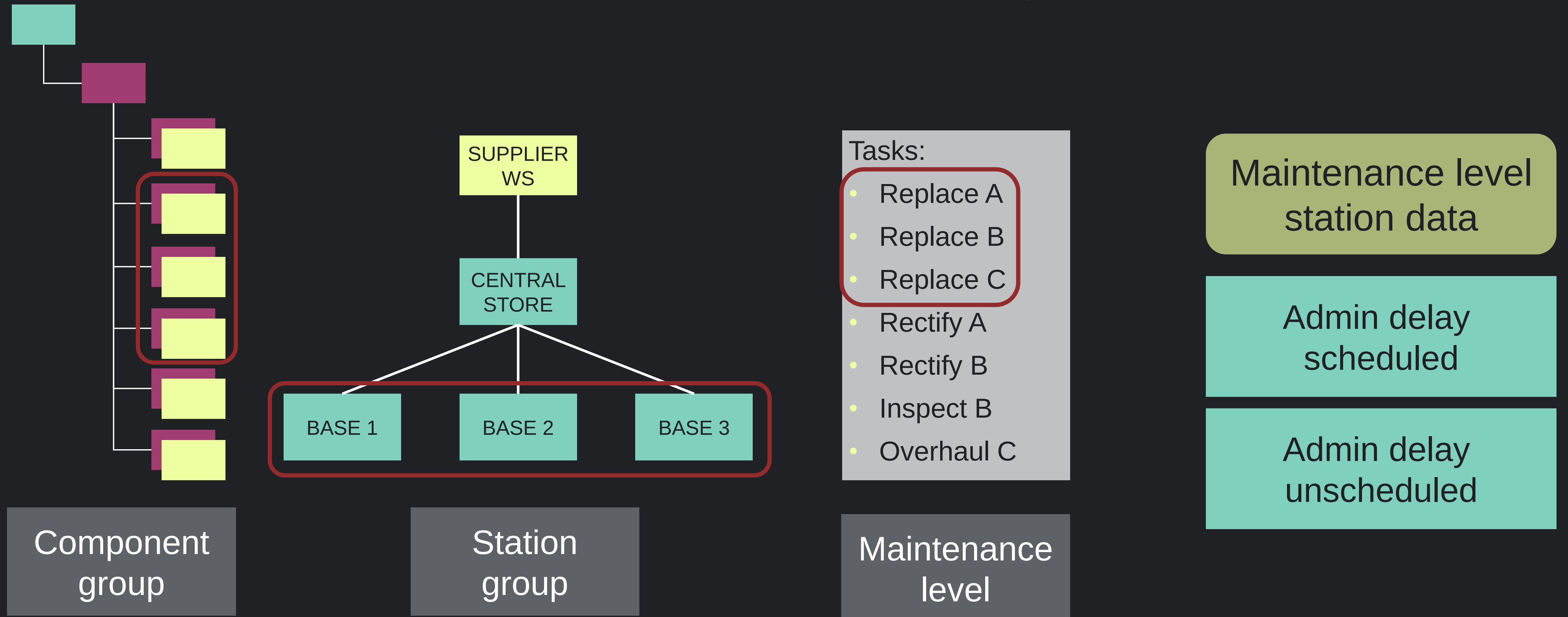
Flexible task duration

- Optional level of detail



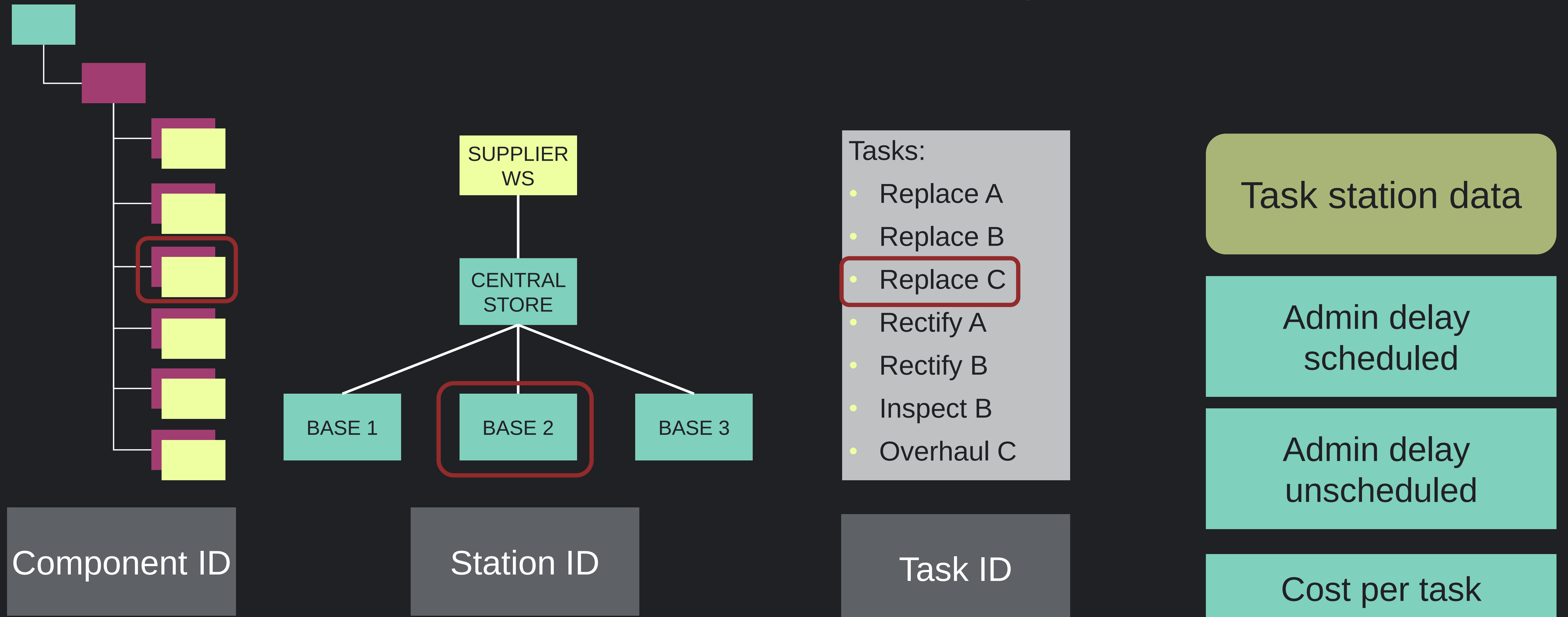
Flexible task duration

- Administrative delays



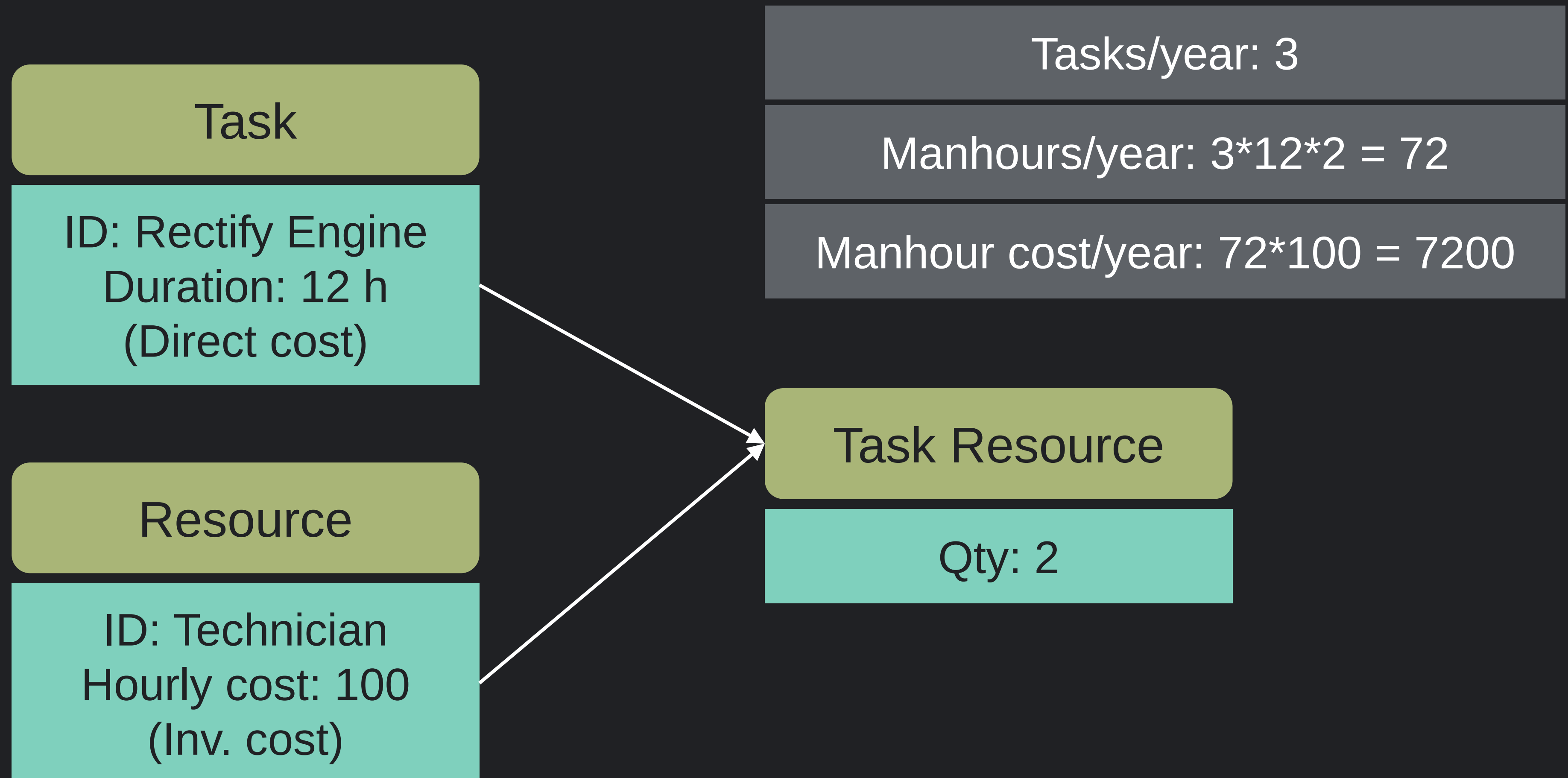
Flexible task duration

- Administrative delays



Flexible task duration

- Task and manhour costs



Thank you.

