hybris software

An SAP Company

hybris software
An SAP Company

hybris Developer Training Part I - Core Platform

Validation Framework

Data Validation Framework

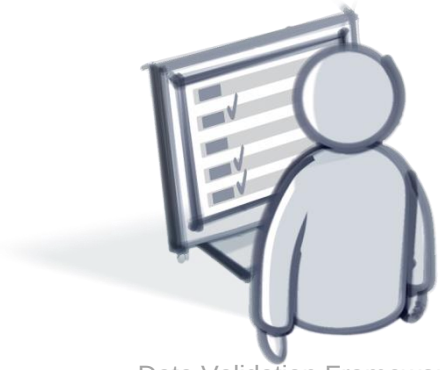
Validation service

Administration cockpit

Cockpit integration

Overview

- The hybris Data Validation Framework is based on the JSR 303 Java validation specification
- It offers an easy and extensible way to validate data
- You can provide user-friendly and meaningful notifications when allowing users to provide or re-enter valid input
- Only validated data is passed on to the persistence layer
- In contrast to the JSR 303 specification, hybris constraints can be created and modified at runtime

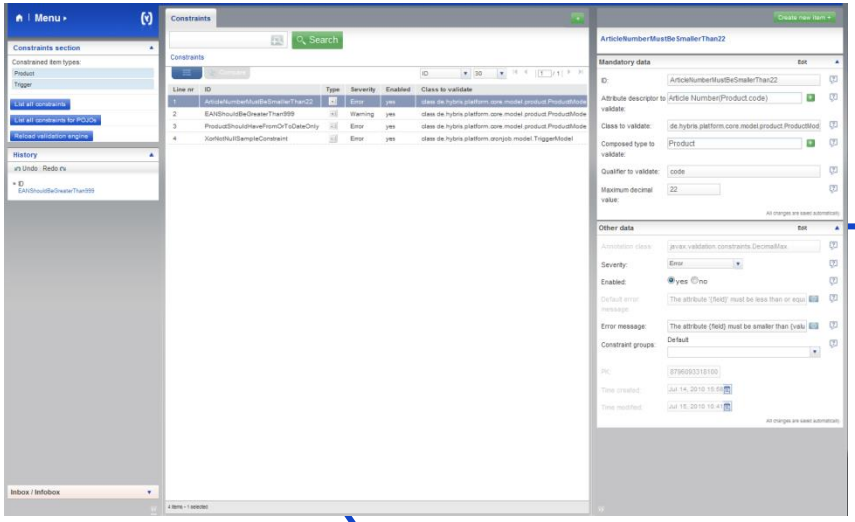


Overview (continued)

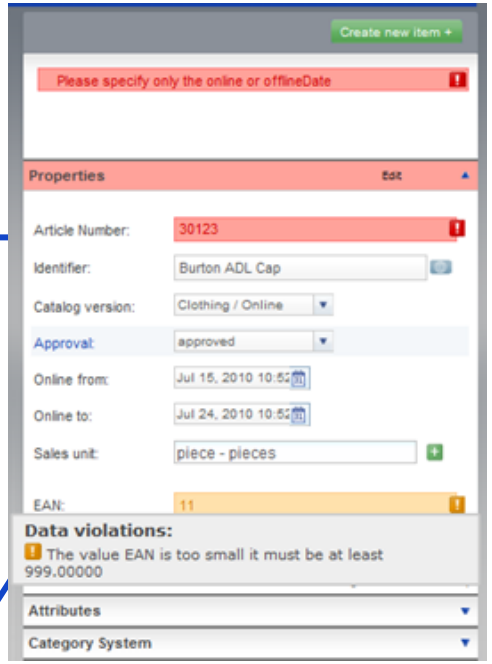
- Constraint definitions are written in Java or XML
- Constraint instances are persisted in the hybris database as items
- You may define your own constraint types
- hybris Data Validation consists of the three following areas:
  - The Validation Service is a service in the `ServiceLayer` which defines constraints and performs data validation
  - The Administration Cockpit allows you to create and manage constraints
  - Cockpit integration provides validation feedback to the user

Definition and use

Administration Cockpit



Cockpit Integration



Validation

ValidationService	Validation Engine

hybris software
An SAP Company

Data Validation Framework

Validation Service

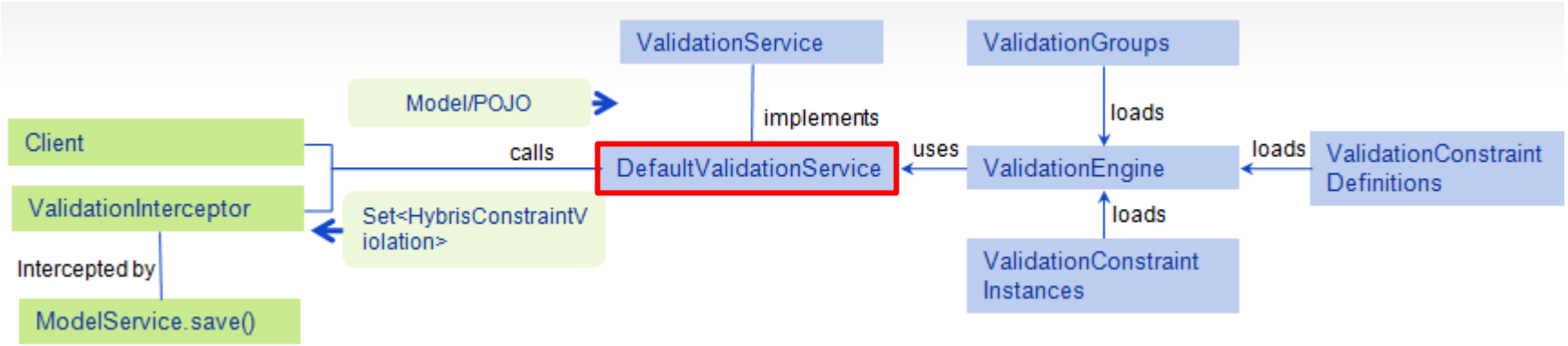
Administration Cockpit

Cockpit Integration

Key Features

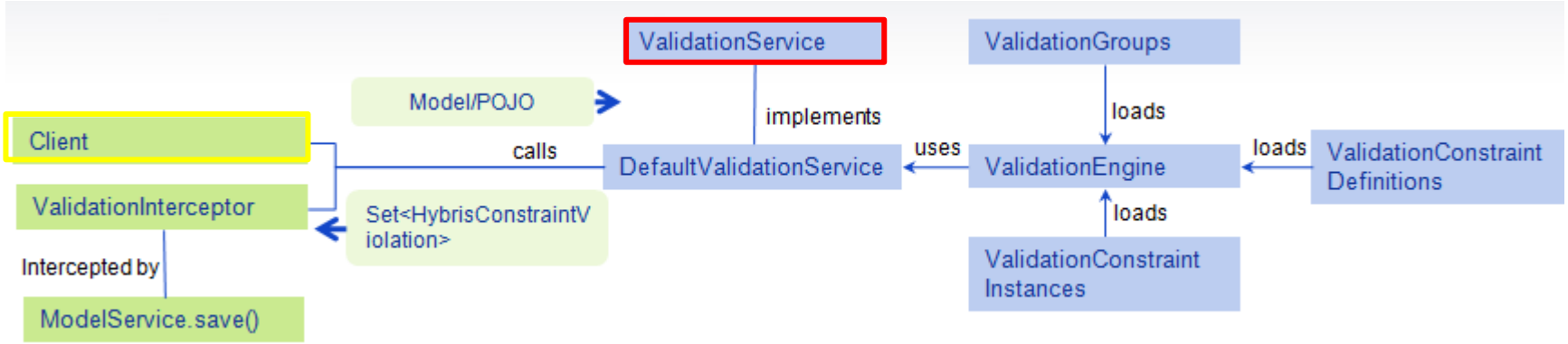
- The Validation Service loads the validation engine with constraints
- Data is validated based on a set of validation constraints
- All `modelService.save` calls are intercepted aborting save operations where necessary
- Validation methods may be called explicitly

Architecture



- Validation is performed by the `DefaultValidationService`
- This service uses a JSR 303-compliant validation engine
- The Validation Service contains two groups of methods:
  - Validation methods to validate items, properties or values
  - Control methods to alter the behavior of the validation engine

Example

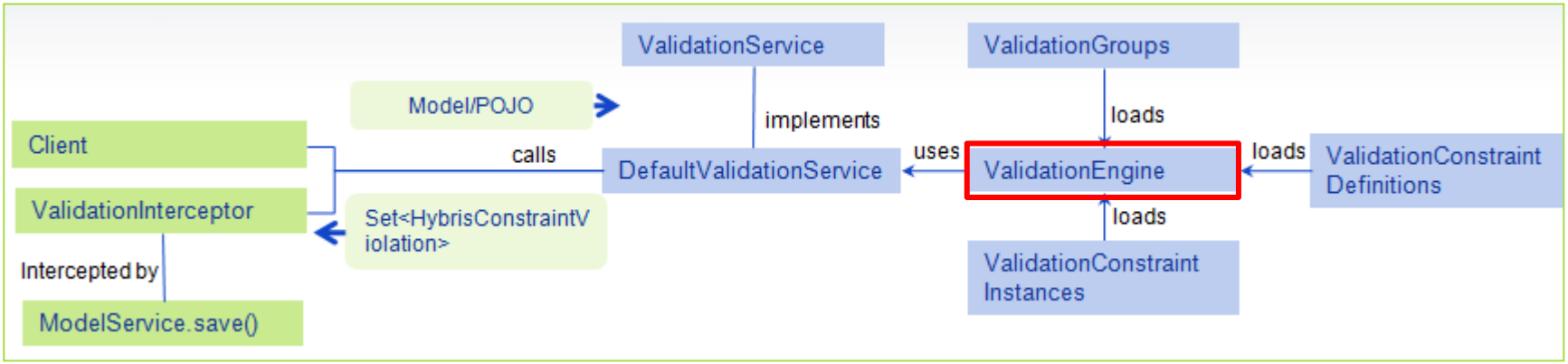



```
interface ValidationService {
void reloadValidationEngine();
    <T> Set<HybrisConstraintViolation> validate(T object,
        Collection<ConstraintGroupModel> g);
    <T> Set<“> validateProperty(T object, String propertyName,
        Collection<ConstraintGroupModel> g);
    <T> Set<“> validateValue(Class<T> beanType,  String propertyName,  Object o,
        Collection<ConstraintGroupModel> g);
    void setActiveConstraintGroups(Collection<ConstraintGroupModel> groups);
...
}
```

Example Client usage:

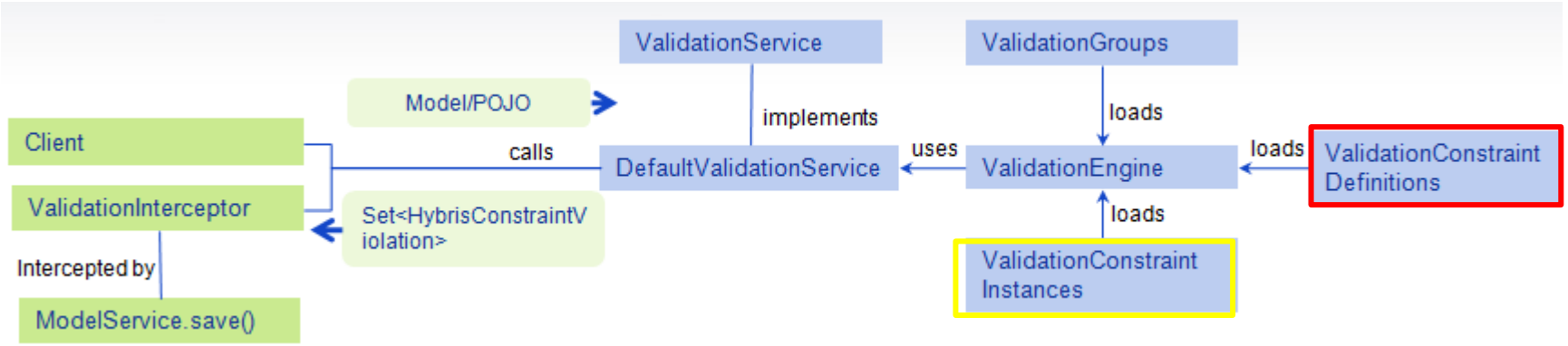
```
ProductModel model = new ProductModel();
Set<HybrisConstraintViolation> constraintViolations = validationService.validate(model);
```

Validation Engine



- The Validation Engine is loaded with JSR 303-compliant constraint instances at run-time
- It can be reloaded with new or modified constraint instances
- All JSR 303 constraint types are supported
- New constraint types may be created but require new files, recompilation and a system initialization or update.

Constraints



All JSR 303-compliant validation constraints can be used:

NotNullConstraint

Java annotation:

```
public class User {
  @NotNull
  private String firstname;
}
```

OOTB Constraints

- The Data Validation framework provides implementations of all constraint types defined in the JSR 303 specification and few additional ones:
  - `PatternConstraint`
  - `PastConstraint, FutureConstraint`
  - `MinConstraint, MaxConstraint`
  - `DecimalMinConstraint, DecimalMaxConstraint`
  - `SizeConstraint`
  - `AssertTrueConstraint, AssertFalseConstraint`
  - `IsNullConstraint, IsNotNullConstraint`
  - `NotEmpty, NotBlank`
  - `XorNotNull Annotation`
  - `Dynamic Annotation`

Constraint types

- Severity
  - Each constraint can be assigned an Error/Warning/Info severity level
  - Error constraints cause a save operation to be aborted
  - Users may choose to ignore Warning constraints and save anyway
- Groups
  - Constraints can be assigned to zero or more groups (zero = default group)
  - The engine can be told which constraint group(s) to use for validation
- There are attribute, type and dynamic constraints

hybris software
An SAP Company

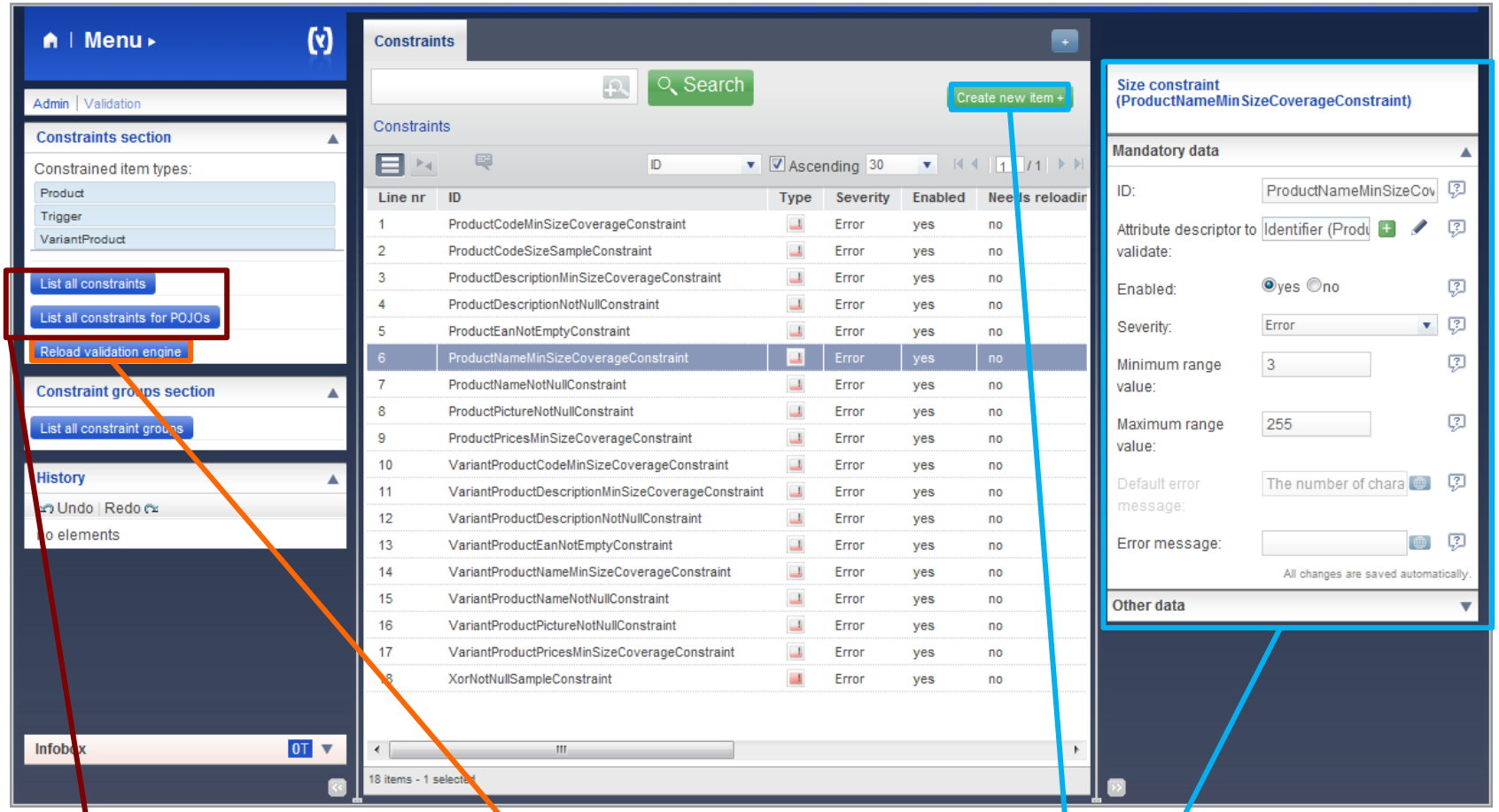
Data Validation Framework

Validation Service

Administration Cockpit

Cockpit Integration

Administration Cockpit overview



List constraints

Reload the validation engine

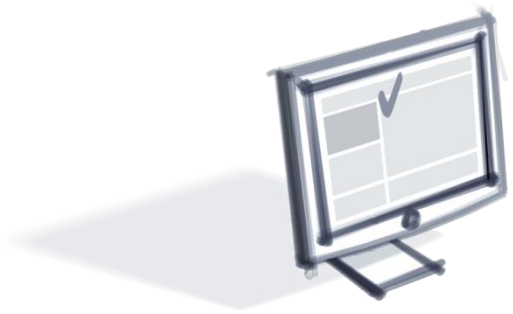
Create and manage constraints

Features

With the administration cockpit you can:

- Create and manage constraints and constraint groups
- Filter constraints
- List constraint declarations
- Reload the validation engine

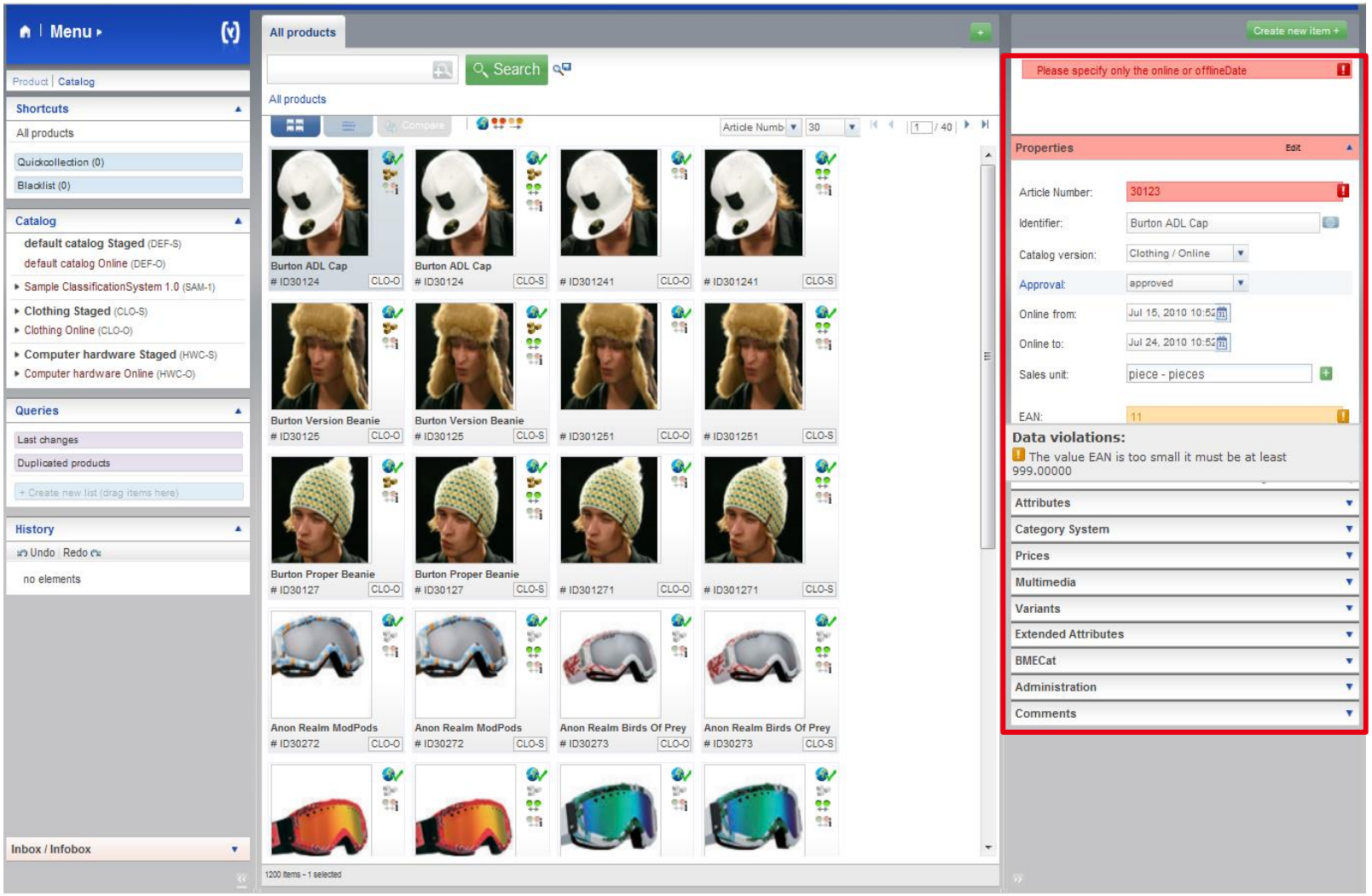
Although this can also be done with the hMC, we recommend using the administration cockpit which is more user-friendly and intuitive.



hybris software
An SAP Company

Data Validation Framework
Validation Service
Administration Cockpit
Cockpit Integration

Cockpit Integration



Data validation is integrated into the editing area of cockpits

Users are notified of all validation issues

Quiz questions

1. Explain the main features of the hybris Validation Framework and the three areas of which it consists.
2. Name the specification on which the hybris Validation Framework is based.
3. Give a short description of the Validation Service architecture.
4. Is it possible to create your own constraint types? If so, how do you declare a custom constraint type?
5. What happens to a save operation if a constraint violation occurs?
6. What is the Administration Cockpit? What are its primary uses?

References

- wiki.hybris.com/display/release5/Data+Validation+Framework
- jcp.org/aboutJava/communityprocess/final/jsr303/index.html
- wiki.hybris.com/display/release5/Administration+Cockpit+-+Business+Guide