



**Java Workflow Tooling (JWT)**

**Release review: JWT v0.6 – part of ‘Galileo’**

**Marc Dutoo (Open Wide, FR)**

**Mickael Istria (Open Wide, FR)**

**Florian Lautenbacher (University of Augsburg, DE)**

**Christian Saad (University of Augsburg, DE)**



# Overview



- Introduction
- About of JWT
- Features in this release
- Non-Code Aspects (Documentation, Communication)
- APIs
- Architectural issues
- Tool usability
- End-of-Life
- Bugzilla
- Standards
- UI Usability
- Schedule
- Communities
- IP Issues
- Project Plan
- Notes

# Introduction



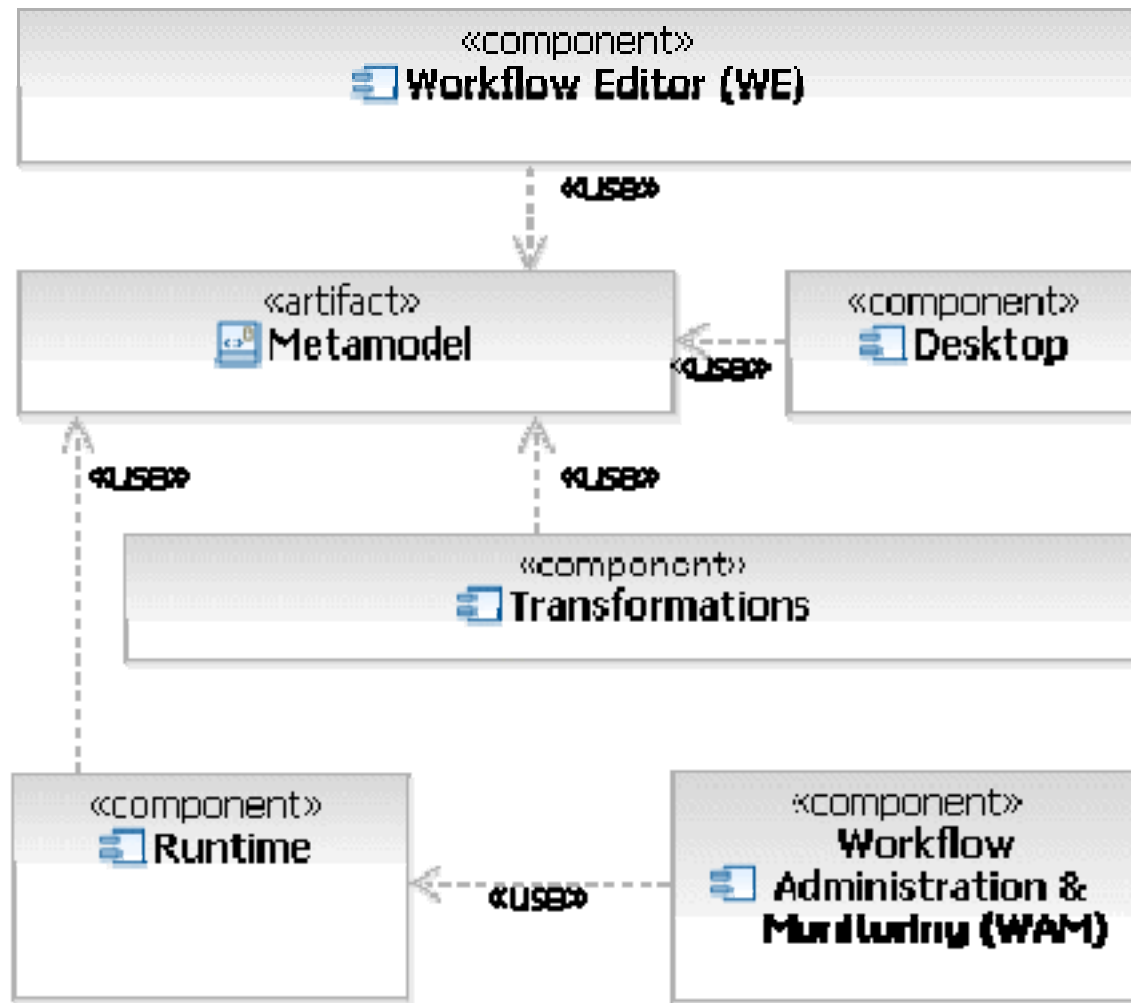
- Java Workflow Tooling (JWT) is a technology sub-project currently in Incubation phase.
- JWT aims at providing a complete Business Process Management (BPM) and workflow tooling platform
  - with a special focus on a unified approach to BPM design, allowing to bridge the gap between BP representation, BP standards, BP engines, BP deployment environments (platforms, Information System, SOA)
- This release (version 0.6) covers improvements to the Workflow Editor, new Transformations and new code in the Runtime component.
- These slides conform to the Eclipse Guidelines for a Release review which can be found under [http://www.eclipse.org/projects/dev\\_process/release-review.php](http://www.eclipse.org/projects/dev_process/release-review.php).

# About JWT - Goals of JWT



- Support for workflow and business process modeling, deployment, execution and monitoring inside Eclipse
- Provide a complete, flexible, interoperable and usable BPM toolkit
- Set of generic and extensible plugins and APIs
- Extensions allowing support for specific business representations, process language formats, process engines, service platforms, etc.
- Targeting and supporting SOA in close collaboration with the Eclipse STP project

# About JWT - Components of JWT



# Features in this release - Overview



- Workflow Editor (WE)
  - Several new extension points
  - New view & new view editor
  - Example framework and first examples
  - Aspect-oriented extensions
- Transformations
  - BPMN to JWT and vice versa
  - JWT to STP-IM
- Runtime
  - Task Engine Framework

# Features – Workflow Editor



- We improved the JWT Workflow Editor with the following features:
  - It is now possible to import and export not only Data/Roles and Applications but also whole Activities.
  - Exported templates can be imported into existing workflows.
  - Added a graphical overview page in addition to the tree-like one.
  - Many small bugfixes

## Features – Workflow Editor (cont.)



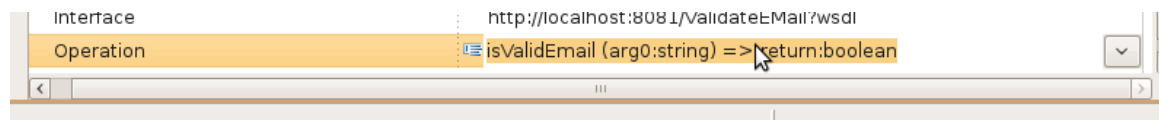
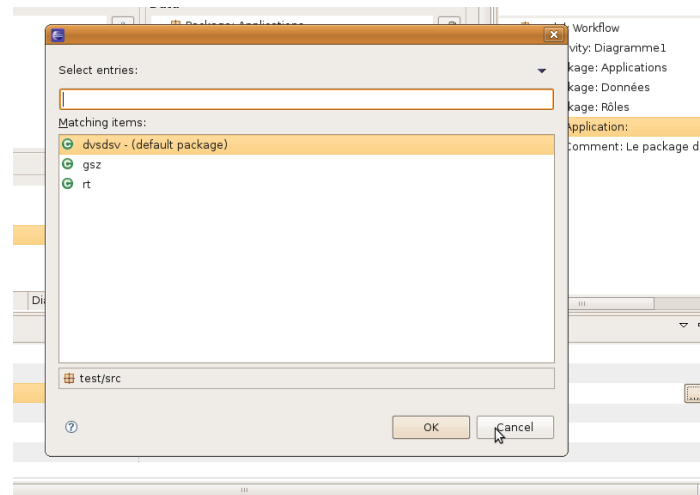
- Added extension points:
  - Extension point to **customize cell editors**: can be used to assist user in entering right values and to give him some feedback (unitary validation).
  - Extension point to **listen to value changed on property editors** and handle them with custom actions.
  - Extension point to **listen to drop events**.
  - Extension point to **listen to double-click event on diagram nodes**.
- Use of these extension points in the Workflow Editor:
  - Drag'n'drop of an existing Java class from the Navigator or Package Browser onto workflow model creates a Java Application in the model.



# Features – Workflow Editor (cont.)



- Use of these extension points in WE release:
  - Integration with Java Nature and JDT for properties that are Java Classes (class selection and introspection, argument guessed automatically...)



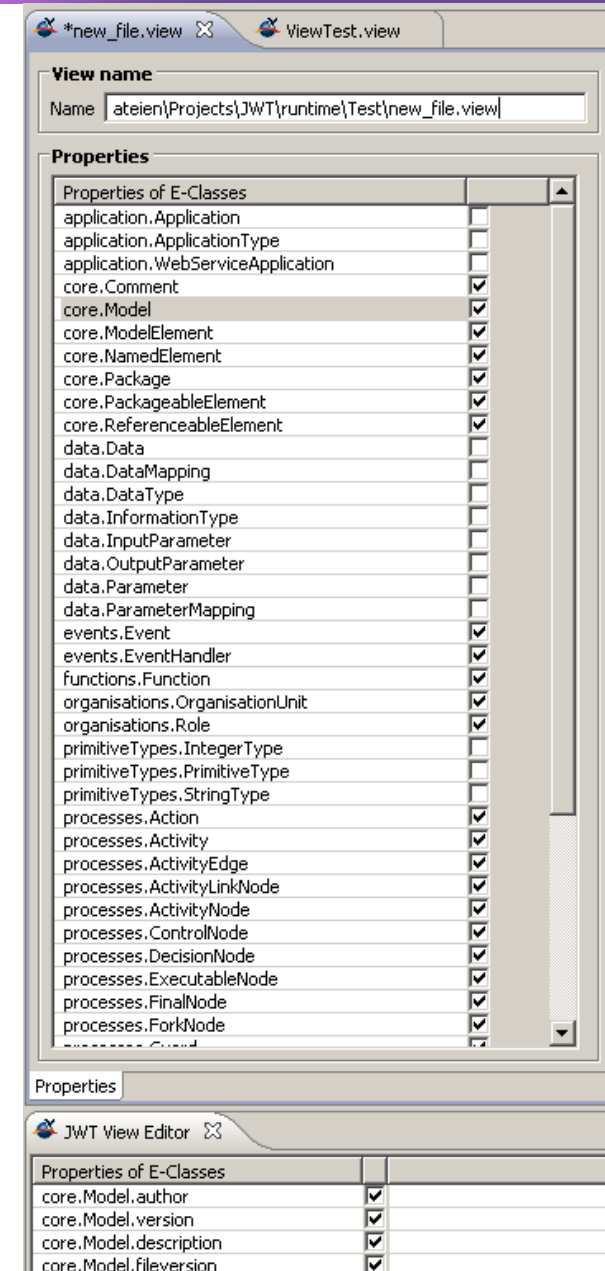
## Features – Workflow Editor (cont.)



- New Aspect-oriented metamodel extensions:
  - Custom workflow features require custom information in the workflow model
  - In addition to the somewhat rigid model extensibility based on EMF inheritance,
  - JWT now provides a mechanism allowing decorate workflow model elements by typed custom information in workflow models
    - thanks to an additional \*\_conf file (not invasive)
    - Profiles describe what kind of additional information (or Aspects) can be stored on model elements.
    - Works for any kind of EMF model!
  - Custom profiles and aspects can be provided by JWT integrators, or designed by JWT power users
  - Examples : all-purpose simple key-value properties, several others in OW2 Scarbo (WS, RMI , script, mail, OWL...)

# Features – Workflow Editor (cont.)

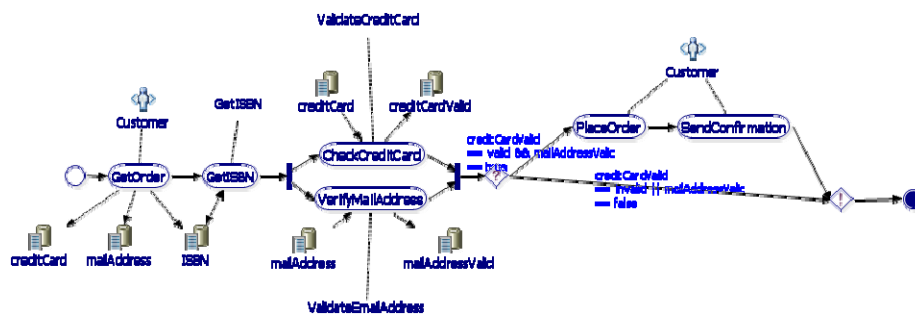
- New view editor:
  - Instead of the old Swing-based view-editor we got the contribution of an own Eclipse-plugin for defining views of a workflow model.
  - This allows to show / hide elements in the Workflow Editor.



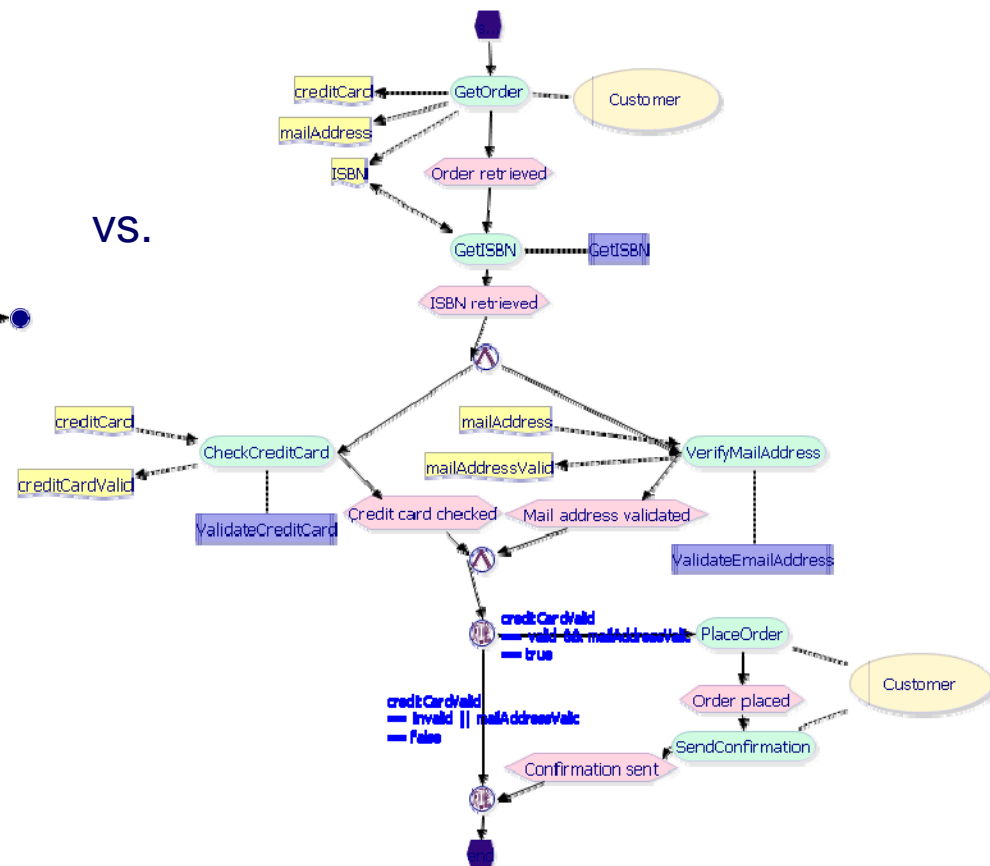
# Features – Workflow Editor (cont.)



- New view for Event-driven Process Chains (EPCs):



VS.



# Features - Transformations



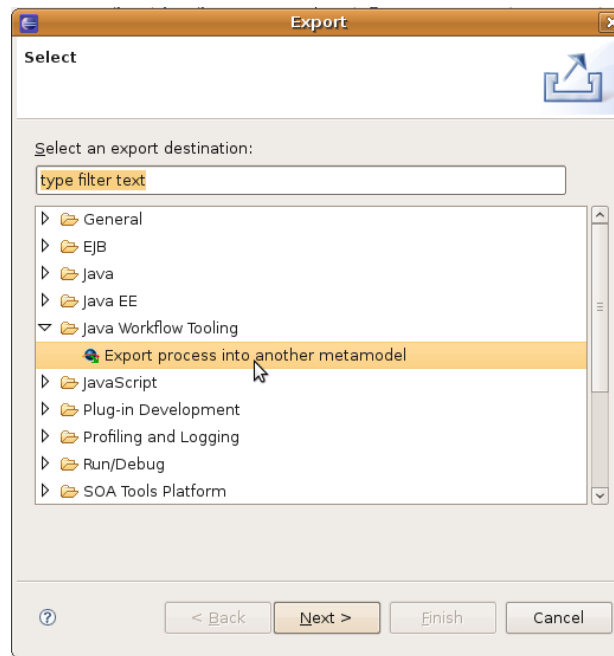
The following transformations now exist:

- Import:
  - STP BPMN-editor to JWT
  
- Export:
  - HTML documentation of the modeled processes
  - JWT back to BPMN
  - JWT to the STP Intermediate Model (STP-IM)
  - JWT to XPDL (e.g. packed in a Bonita Archive)
  - ...

# Features - Transformations



- UI
  - Transformation UI now uses Wizards
  - Import and export wizards can now be found in Import and Export menus



# Features - Runtime



- Task Engine Framework (TEF)
  - API to handle automated tasks at runtime with the same paradigm as the one used at model-time
    - Applications as black boxes
    - Independent from engine
    - Using data mapping
  - Builds a bridge between process engines and human interaction on the one side with automated task implementations on the other.

# Non-Code Aspects - Documentation



- As everything else, documentation, communication and visibility is being geared up in order to best target the Galileo simultaneous release.
- Our website has been changed to conform to the Nova style and has been further improved.
- Several pages on the wiki describe to end users the usage of the Workflow Editor, Transformations, etc. A new tutorial describes the features of the JWT components.
- Several pages on the wiki provide developer documentation. All extension possibilities by external plugins are documented there, and have examples that are available at least in the CVS.
- Such documentation is up to date.



# Non-Code Aspects - Communication



- There has been a presentation about JWT at Eclipse Summit Europe 07, EclipseCon08, Open World Forum 2008, EclipseCon09, Solutions Linux 2009 and Eclipse Forum Europe 2009.
- News have been posted on sites such as [theserverside.com](http://theserverside.com), JWT has now its project in Ohloh's directory, own project-related blog...
- In June 2009 a first publication appears in the German-speaking „Eclipse-Magazin“ describing the different features of JWT.
- NB. See about translation at UI Usability.

# APIs



- The code conforms to the Eclipse Quality.
- The code has been implemented by committers of the project.
- The API has been officially declared (using PDE tooling) in prior to this release.
- Several examples have been tested and several jUnit tests exist.
- Build (includes unit testing) has been fully automated and is easily repeatable.

# Architectural issues



- Workflow Editor
  - jwt-we: Contains Workflow Editor and meta-model
  - jwt-view: Editor for Views
  - jwt-we-conf-model\*: Custom aspect extensions for the meta-model
  - jwt-we-conf-property.\*: Allows to add custom properties to model elements
  - jwt-we-\*-sample: Several examples on how to use extension points
  - jwt-we-view-\*: Additional views for the Workflow Editor
  - jwt-we-plugins: Several additions to the Workflow Editor
  
- Workflow Administration and Monitoring
  - jwt-wam-api: API to invoke a workflow service
  - jwt-wam-monitoring: Generic monitoring plugin

# Architectural issues (cont.)



- Several extension points available that are already used by plugins in order to extend the JWT Workflow Editor.



# Tool usability



- JWT enables a user to model his/her processes and workflows and use these models not only for documentation, but also for execution.
- The model can be used to generate code in different languages (such as XPDL)
- The model can be transformed to other models (e.g. STP BPMN).
- It already provides several extension points where others can build on it and extend several parts of the editor.
- If necessary, the meta-model can be adapted to the needs of each user.
- In the next release we will include a new view mechanism which allows different positions of model elements in different views.

# End-of-Life



- There are no features that are end-of-life'd in this release.

# Bugzilla



- There are already 143 closed and fixed bugs for JWT (mostly the Workflow Editor).
- Bugzilla currently knows 54 open bugs where most of them are feature requests for version 0.7.0.

## Closed:

Component	Version				Total
	0.4.0	0.5.0	0.6.0	0.7.0	
Releng	.	.	3	.	3
Transformations	.	.	10	1	11
WE	32	40	52	1	125
Yearly Release	.	.	4	.	4
<b>Total</b>	<b>32</b>	<b>40</b>	<b>69</b>	<b>2</b>	<b>143</b>

## Open:

Component	Version		Total
	0.7.0	unspecified	
Desktop	5	1	6
Releng	3	.	3
Transformations	3	.	3
WE	34	4	38
Wam	1	.	1
Yearly Release	.	3	3
<b>Total</b>	<b>46</b>	<b>8</b>	<b>54</b>

# Standards



The relationship to existing standards has been achieved by the following:

- Addition of a UML Activity diagram view
  - Addition of a EPC model view
  - Transformation to the Eclipse STP BPMN modeler
  - Codegeneration of XPD L possible
- 
- Existing transformation to the STP-IM, which aims to bridge different standard oriented tools within the STP project, like BPMN and SCA editors or BPEL.



## Standards (cont.)



- JWT has been enriched on the runtime side also:
  - Runtime process APIs allowing workflow engines in a standard way
    1. to integrate with any Java workflow engine
    2. To provide service orchestration features using any service platform. They have been contributed in the WAM and Runtime component.
  - The new OW2 Scarbo project provides its reference implementation, on top of the XPDL-compliant Bonita Engine and the SCA standard compliant Frascati service platform.
    - Because of licensing issues – in addition to Scarbo being a complete solution and not only tooling - this development has been contributed to OW2 as a new project called Scarbo, see <http://scarbo.ow2.org/>.

# UI Usability



- The Workflow Editor supported originally several languages such as English, German and French. For this release we collaborated with Babel so that much more languages now exist.
- Several wizards exist to ease the usage of the Workflow Editor.
- All transformations are easily selectable via Import and Export menus thanks to the transformation base.
- The Eclipse User Interface Guidelines have been followed.
- A UI walkthrough is planned for the following release 0.7.

# Schedule



- The originally planned schedule was respected, at least „in the spirit“, but not „in the letter“.
- Two releases mentioned in the previous project plan have been skipped :
- 0.6 "Integrators" (because Bull and jBoss' schedule changed)
- 0.7 "SOA" (it became OW2 Scarbo, see Communities).

# Communities



- JWT now has a mentor : John Graham (RedHat), former DTP Lead
- Active bugzilla usage by the committers and others.
- Many discussions on the mailing list, inside JWT as well as with partners from the STP projects (STP IM, BPMN, etc.).
- Newsgroup not used too much.
- End user involvement is less rare. More meaningful are discussions with several companies (Bull, jBoss) who express their need for workflow tooling within Eclipse have already produced some collaboration (ex. FactoryRegistry), are going further on and feed JWT's requirements.
- JWT Integrations include now OW2 Scarbo in addition to AgilPro.
- Coordination with several OW2 projects such as Bonita, Orchestra, FraSCAti, Spagic, Scarbo.

# IP Issues



- The code has been committed by individuals who are either committers of the project themselves or their foundation. Code contributions from others have undergone a CQ.
- The contribution questionnaires for the all components have been completed.
- The legal information has been inserted into the source code as described in the Eclipse IP Policy.
- The code has been approved by the EMO-IP-Team under CQ 1936, 2039, 2041, 2042, 2977, 3008, 3106, 3107, 3108, 3109 and 3110 as described on our IP Log  
[http://www.eclipse.org/projects/ip\\_log.php?projectid=technology.jwt](http://www.eclipse.org/projects/ip_log.php?projectid=technology.jwt).
- There are no outstanding CQs and the IP log has been committed in time for this release.

# Project Plan



- The next release (version 0.7.0) is scheduled for September, 24th 2009.
- The next release will consist of
  - an updated version of the workflow editor with a new view mechanism,
  - the JWT metamodel as a separate plugin,
  - new transformations (e.g. JWT to jPDL),
  - improved runtime and WAM component,
  - a first implementation of the desktop component as well as
  - minor bugfixes.
- After that we aim for a graduation (version 1.0.0) and move release. We are already discussing with PMC members of existing top-level projects where we might fit in best.

# Notes



- The Eclipse development process document and the Guidelines document have been read and approved by the project leads and committers of the JWT project.

# Thanks for reading this document!



- Marc Dutoo (Project Co-Lead) - Open Wide, FR
- Florian Lautenbacher (Project Co-Lead) – University of Augsburg, DE
- Mickael Istria (Release Engineer) – OpenWide, FR
- John Graham (Mentor) – JBoss, a division of RedHat, US
- Christian Saad – University of Augsburg, DE
- Stéphane Drapeau – Obeo SA, FR
- Alain Boulze – INRIA / OW2, FR
- Miguel Valdez Faura, Marc Blachon – Individuals, FR
- Guillaume Decarnin – Open Wide, FR
- Pierre Vigneras, Steve Egbert – Bull, FR
- Maxime Porhel – Obeo, FR
- and many others...

## The JWT project team

