



First Workflow Orchestrator User Developer Meeting

Hans Trompert

22 September 2025



| Agenda

- Meetings
- Member overview
- Projects
- International code sprints
- Core feature updates
- Backlog / Roadmap



| Meetings

- WFO Software Development Sprint
 - February 2025, Berkeley, 5-day sprint working on core issues
- WFO User Meeting
 - June 2024, Rennes, collocated with TNC24
 - June 2025, Brighton, collocated with TNC25
- WFO User Developer Days
 - September 2025, Amsterdam

| Member overview

Partner



Graduate



nomios

Sandbox



REANVZ



Projects

Projects under the GitHub workfloworchestrator organisation:

- orchestrator-core
- oauth2-lib
- orchestrator-ui-library / example-orchestrator-ui
- pydantic-forms
- pydantic-forms-ui
- lso
- example orchestrator
- supa + polynsi
- gnmi-cluster

See <https://github.com/workfloworchestrator>



| International code sprints

- Joint sprints to work on WFO features
- Two week sprint every four weeks
- Each WFO partner contributes half an FTE
- Prioritise and refine the backlog (currently done by SURF)
- General development direction guidance from WFO programme board
- Feature requests from all WFO users
- Sprint meetings:
 - 1x issue refinement, 3x standup, 1x sprint end
- Slack channel for discussions and bridging time zones



| Core feature updates 1/8

Orchestrator-core from version 2.2.2 to 4.4.0 highlights

- Add AuthManager for configurable AuthN and AuthZ
 - Authentication (default OIDCAuth)
 - HTTP authorization (default OPAAuthorization)
 - GraphQL authorization (default GraphQLOPAAuthorization)
 - Override any by using:
 - `app.register_authentication()`
 - `app.register_authorization()`
 - `app.register_graphql_authorization()`
 - <https://workfloworchestrator.org/orchestrator-core/reference-docs/auth-backend-and-frontend>

| Core feature updates 2/8

- Reliable WebSocket communication between backend and frontend
- Add Python 3.13 support, and make it the minimal supported version
- Add PostgreSQL 17, remove PostgreSQL 14 support
- Removal of re-imports of pydantic-forms (breaking change)
 - Replace import from orchestrator.types to pydantic_forms.types
 - A migration script is available
 - <https://workfloworchestrator.org/orchestrator-core/migration-guide/3.0/>
- Add callback progress update endpoint for callback step's
 - Used by callback jobs to provide progress updates during execution
 - The progress updates are stored in the process state under `callback_progress` key
 - On completion the `callback_progress` key is removed, and final result can be found under `callback_result`

| Core feature updates 3/8

- Add `@ensure_provisioning_status` decorator
 - For step(s) that modify the subscription
 - Ensure that the step(s) are only executed while the subscription lifecycle state is **PROVISIONING**
- Add support for pydantic-forms 2.x (deprecating 1.4.0)
 - For single-value fields **ReadOnlyField** has been replaced by **read_only_field**
 - For list fields **ReadOnlyField** has been replaced by **read_only_list**
 - The argument **default_type** is no longer needed to specify the type for lists
- Redis domain model caching removed (breaking change in orchestrator core 4.0.0)
 - Made possible by greatly speeding up domain model related database queries
 - Note that all workflows must have run to completion before upgrading



| Core feature updates 4/8

- Add new workflow target VALIDATE
 - Breaking change in orchestrator core 4.0.0
 - New target to distinguish between validation workflows and system tasks
 - Add is_task field to workflows table and isTask field to the API
 - Both VALIDATE and SYSTEM workflows are labelled with is_task set to True
 - <https://workfloworchestrator.org/orchestrator-core/migration-guide/4.0/>
- Show UI components and orchestrator core software versions in frontend
 - Add GraphQL endpoint
 - Show in orchestrator-ui



| Core feature updates 5/8

- Add Prometheus compatible metrics endpoint `/api/metrics`
 - Various subscription, workflow and task statistics
 - Extensible with custom metric collectors
- Add example Grafana dashboard for metrics endpoint
- Make workflow description mandatory
 - Will break user deployments with workflows without description
 - Description can now be changed through the API and orchestrator-UI
- Application settings overview endpoint `/api/settings/overview`
 - settings are grouped by name
 - sensitive values are masked for security reasons



| Core feature updates 6/8

- Add role based access control for running, resuming and retrying workflows
 - More details in the Deep Dive RBAC session
- Expose RBAC through isAllowed field in StartOptions endpoint
- Expose RBAC through process and subscription detail endpoints
- Deprecate workflow_target field in ProcessSubscriptionTable
 - Workflow target is stored in the workflows table
 - Deprecate target parameter in step store_process_subscription()
- Improve step timestamping logic
 - Show actual step run time in UI instead of the difference between start time of consecutive steps
 - Add new started_at field to table and API
 - Rename executed_at field to completed_at in table and API

| Core feature updates 7/8

- Add lastValidatedAt GraphQL field to subscriptions
 - Make it easy to identify subscriptions that have not been validated for a certain time
- New reconcile workflow decorator for subscription synchronization with external systems
 - Add new workflow target RECONCILE
 - Only one reconcile workflow allowed per product
 - Add reconcile_workflow() decorator
 - Reconcile workflows only contain the steps that update information in external systems
- Add integration tests for Celery functionality in the orchestrator-core
 - tests for core workflow operations (NEW_TASK, NEW_WORKFLOW, RESUME_TASK, RESUME_WORKFLOW)
 - Add worker configuration for test environment
 - Added a CI GitHub workflow to run the Celery tests

| Core feature updates 8/8

- Replace schedule with APScheduler
 - Persistent job store
 - Use `@scheduler.scheduled_job()` instead of `@scheduler`
 - Triggers: Interval, Cron, Date, CalendarInterval, And, Or
 - Add `scheduledTasks` GraphQL endpoint



| Backlog / Roadmap 1/2

- Add NSI support to example orchestrator
 - Add NSISTP and NSIP2P products
 - Add PolyNSI and SuPA with SURF/WFO backend
- Extend schedules functionality
 - Upgrade schedules API endpoint to support CRUD
 - Add list/create/edit/delete schedule to UI
- Make demo orchstrator more robust and usable
- Replace Uniforms with pydantic-forms
 - Add more components and compossibility
 - Easy to extend existing components or to add new ones
- Open Source some of the SURF specific UI components




| Backlog / Roadmap 2/2

- (Better) Integration tests for example orchestrator and example orchestrator UI
- Improve documentation structure
- Revive self-paced orchestrator introduction workshop
- Pydantic V3 (upgrade is expected to be much less drastic than from version V1 to V2)
- Sub or parallel workflows
 - Make the executor more fine-grained, run per step instead complete workflow
 - Implement parallel steps?
 - Implement workflows per product block?
 - Implement parallel (sub) workflows?
- Add AI copilot to orchestrator
- Add Agentic AI to orchestrator



**Thank you for
your attention!**

 Hans Trompert

 [hans.trompert\(@\)surf.nl](mailto:hans.trompert(@)surf.nl)

 <https://discord.gg/TZZz2bGw>

SURF