F-Air 4.0

Orchestration of *any* air quality model

Configuration-driven, modular, extensible model execution service

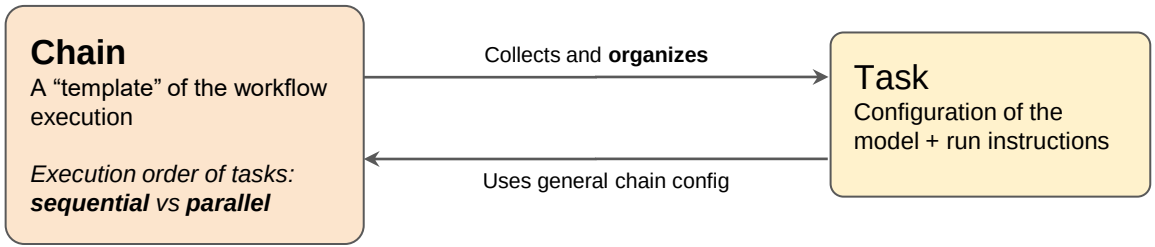
Stefano Cilloni
stefano.cilloni@suez.com

Agenda

- Configurability and modularity by design
- Execution efficiency
- Integrations
- Architecture

TASKS: composable by the building blocks

- Now available c.a. **50 tasks**
  - New tasks are being added…
- **Configurable** by design to the finest level
  - Choose models' configurations



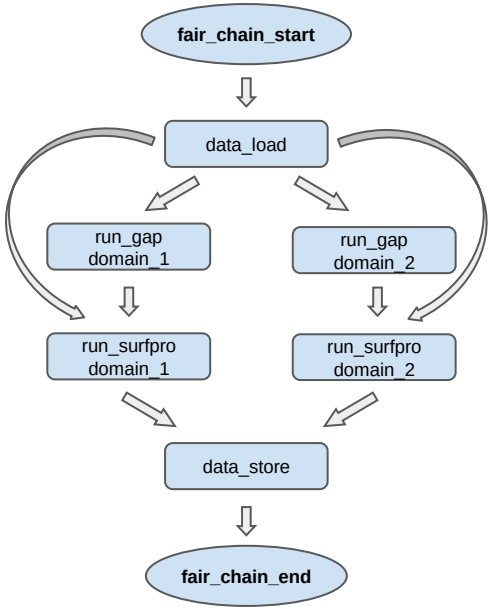
A **slice** of the available tasks

- FARM
- Orsa
- Gap
- Surfpro
- EMMA
- Modeval
- Kalman
- PSwift
- PSpray
- Surfpro
- Altemi
- Postbin
- Arpmeas
- Ar2min
- Spradsg
- Tim2par
- Orogex
- Landex

+ 30 more..

Flexible: create and run any chain

- The chain ***workflow***




```
….
chain_config:
    METEO_DOMAIN: …
    TIME: start: 2024-04-11T09:00
    ...
workflow:
  - fair_chain_start --> data_load

  - data_load --> run_gap_domain_1
  - data_load --> run_gap_domain_2
  - data_load --> run_surfpro_domain_1
  - data_load --> run_surfpro_domain_2

  - run_gap_domain_1 --> run_surfpro_domain_1
  - run_gap_domain_2 --> run_surfpro_domain_2
  - run_surfpro_domain_1 --> data_store
  - run_surfpro_domain_2 --> data_store

  - data_store --> fair_chain_end
```

This is the **actual definition** of a workflow!

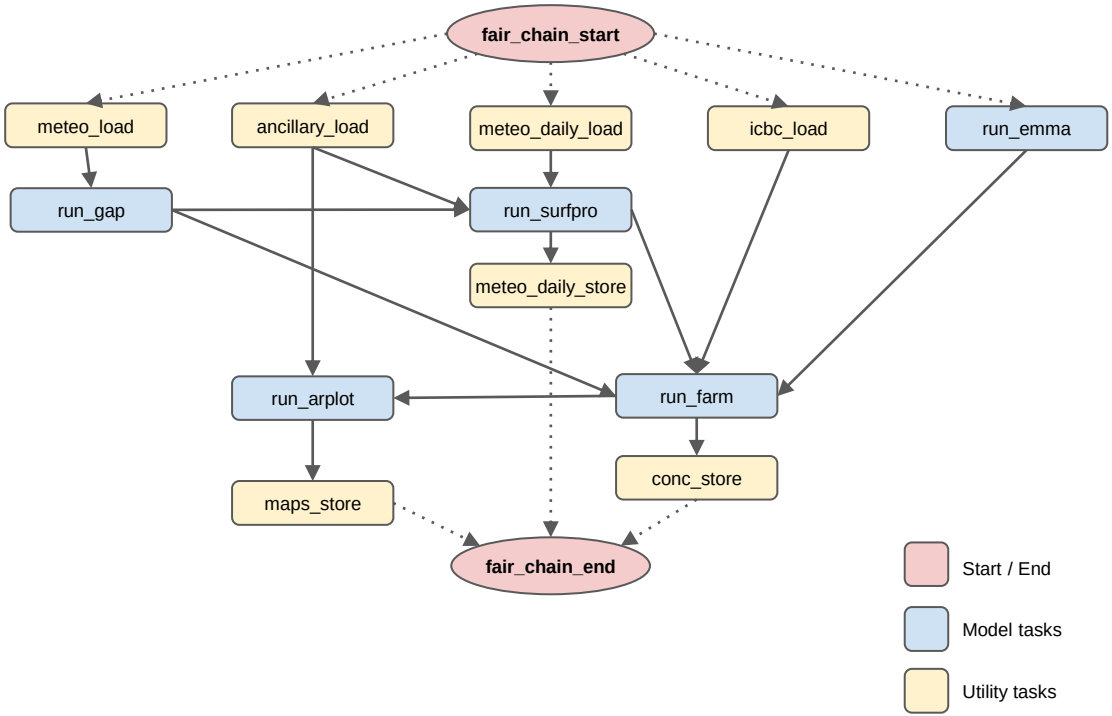
Not just an example!

Example 1 - A workflow

```
workflow:
- fair_chain_start -> meteo_load
- fair_chain_start -> meteo_daily_load
- fair_chain_start -> icbc_load
- fair_chain_start -> ancillary_load
- fair_chain_start -> run_emma

- meteo_load -> run_gap
- meteo_daily_load -> run_surfpro
- icbc_load -> run_farm
- ancillary_load -> run_surfpro
- ancillary_load -> run_arplot
- run_gap -> run_surfpro
- run_gap -> run_farm
- run_surfpro -> run_farm
- run_emma -> run_farm
- run_farm -> run_arplot
- run_surfpro -> meteo_daily_store
- run_farm -> conc_store
- run_arplot -> maps_store

- conc_store -> fair_chain_end
- maps_store -> fair_chain_end
- meteo_daily_store -> fair_chain_end
```

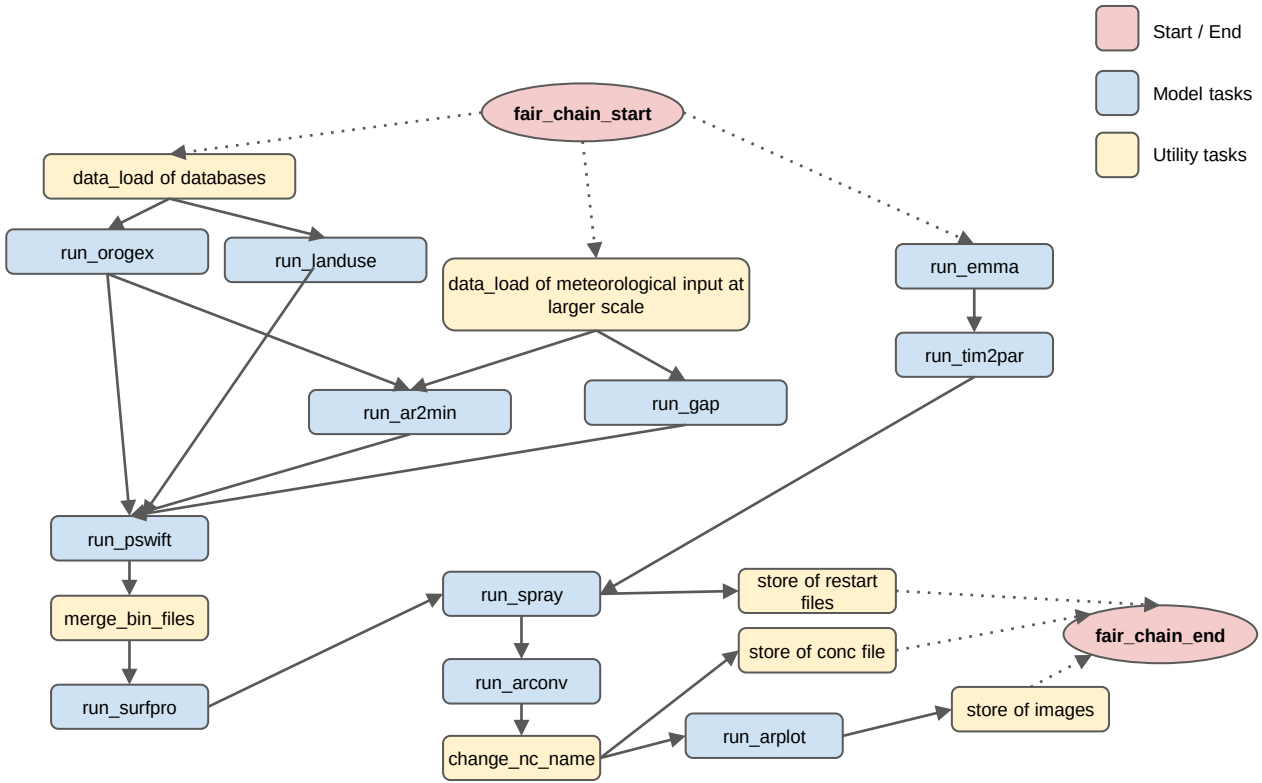


Example 2 - A more complete workflow

```
workflow:
  - fair_chain_start -> data_load_wrf
  - fair_chain_start -> data_load_for_orogex

  - data_load_for_orogex -> orogex_task
  - fair_chain_start -> data_load_for_landex
  - data_load_for_landex -> landex_task
  - data_load_wrf -> ar2min_task
  - orogex_task -> ar2min_task
  - data_load_wrf -> gap_task
  - fair_chain_start -> data_load_for_pswift
  - data_load_for_pswift -> pswift_task
  - ar2min_task -> pswift_task
  - landex_task -> pswift_task
  - orogex_task -> pswift_task
  - pswift_task -> merge_binary_files_task
  - fair_chain_start -> data_load_for_surfpro
  - data_load_for_surfpro -> surfpro_task
  - merge_binary_files_task -> surfpro_task
  - gap_task -> surfpro_task
  - landex_task -> surfpro_task
  - fair_chain_start -> emma_task
  - emma_task -> tim2par_task
  - surfpro_task -> spray_task
  - emma_task -> spray_task
  - tim2par_task -> spray_task
  - spray_task -> arconv_task
  - spray_task -> change_netcdf_variable_names
  - arconv_task -> change_netcdf_variable_names
  - fair_chain_start -> data_load_for_arplot
  - data_load_for_arplot -> arplot_task
  - change_netcdf_variable_names -> arplot_task
  - change_netcdf_variable_names -> data_store
  - spray_task -> data_store
  - arplot_task -> data_store

  - data_store -> fair_chain_end
```



In practice - fairctl

spray_chain.yaml

```yaml
api_version: v1
kind: Chain
metadata:
  name: spray_chain
spec:
  chain_config:
    domains_groups:
      SPRAY_DOMAINS:
        [...]
    time:
      start_datetime: "2024-04-11T12:00:00+01:00"
      end_datetime: "2024-04-11T13:00:00+01:00"
    [...]

  tasks:
    - api_version: v1
      kind: Task
      metadata:
        name: data_load_from_surfpro
      spec:
        [...]
    - api_version: v1
      kind: Task
      metadata:
        name: data_load_from_emma
      spec:
        [...]
    - api_version: v1
      kind: Task
      metadata:
        name: spray_task
      spec:
          target_domain:
            domain_group: SPRAY_DOMAINS
            domain_id: 1
          epoch_index: 1
          input_files:
            meteo_3d_binary_file_id:
SURFPRO_OUTFILE_PSWIFT_DOMAINS_1_BIN
            emission_pemtim_file_id: PEMTIM_FILE
            emission_pemspe_file_id: PEMSPE_FILE
            emission_pempar_file_id: PEMPAR_FILE
          output_files:
            - CONC
          restart_parameters:
            use_restart: 1
            save_restart: 1
            restart_file_id: RESTART_SPRAY_DOMAINS_1
            SIN_RESTA: 3600
            SFREQ_RESTA: 3600
          [...]
        [...]

  workflow:
    - fair_chain_start -> data_load_from_surfpro
    - fair_chain_start -> data_load_from_emma
    - fair_chain_start -> data_load_from_tim2par
    - fair_chain_start -> data_load_for_spray

    - data_load_from_surfpro -> spray_task
    - data_load_from_emma -> spray_task
    - data_load_from_tim2par -> spray_task
    - data_load_for_spray -> spray_task

    - spray_task -> data_store

    - data_store -> fair_chain_end
```

```
$ fairctl create chain -f ./spray_chain.yaml
chain/example_chain_spray created


$ fairctl get chain
NAME                          TASKS   PERFORMED RUNS
spray_chain                    5       0
other_chain_example            7       2


$ fairctl get chain spray_chain  -o yaml
kind: Chain
metadata:
  name: spray_chain
spec:
  chain_config:
    domains_groups:
      SPRAY_DOMAINS:
...
```

```
$ fairctl get example chains
$ fairctl get example chain spray
```

Integrations

- Launch job in clusters
  - Connect with multiple resource managers
- Resource managers
  - Kubernetes
  - Htcondor (work in progress)
  - In the plans to support SLURM and more…

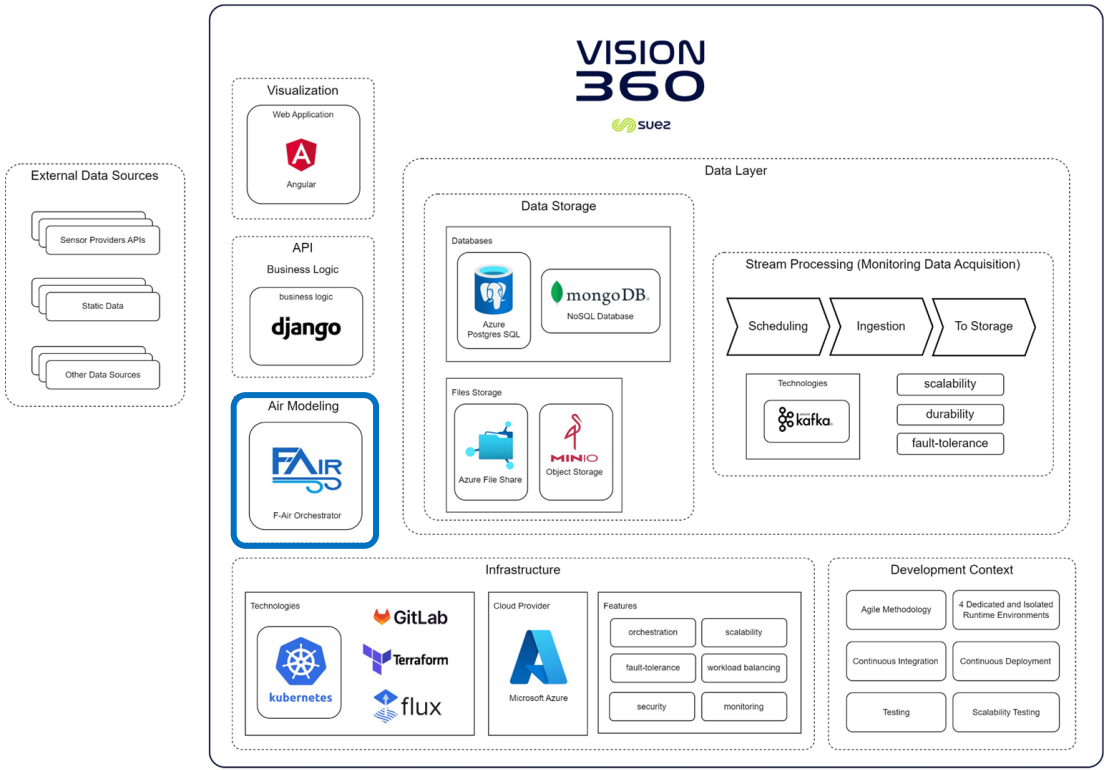
- Shipped as
  - Python package

    Bare-metal or experimental setups
  - Docker container

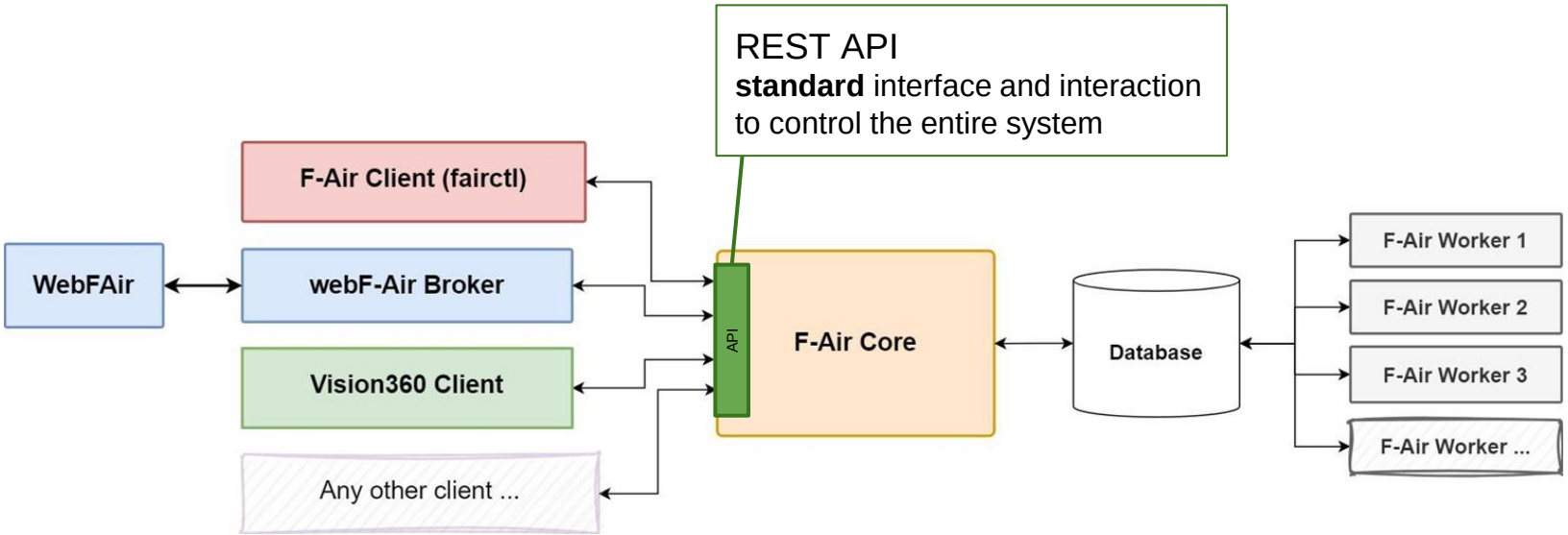
    Containerized environments

Powering the AQ modeling in Vision360

F-Air 4.0 is the **engine** → power AQ modeling in Vision360



The REST API allows multiple (possibly custom) clients to interact with F-Air



F-Air 4.0

Orchestration of *any* air quality model

Configuration-driven, modular, extensible model execution service

Q & A

Example - Spray task configuration

```yaml
- api_version: v1
  kind: Task
  metadata:
    name: spray_task
  spec:
    task_class: "f_air.tasks.models.spray.v3_4_0.Spray_v1"
    task_config:
      execution:
        required_cpus: 4
        MAXPTC: 50000000
        MXNSTK: 50000000
      target_domain:
        domain_group: SPRAY_DOMAINS
        domain_id: 1
      input_files:
        meteo_3d_binary_file_id: SURFPRO_OUTFILE_PSWIFT_DOMAINS_1_BIN
        emission_pemtim_file_id: PEMTIM_FILE
        emission_pemspe_file_id: PEMSPE_FILE
        emission_pempar_file_id: PEMPAR_FILE
      output_files:
        - CONC
      restart_parameters:
        use_restart: 1
        save_restart: 1
        restart_file_id: RESTART_SPRAY_DOMAINS_1
        SIN_RESTA: 3600
        SFREQ_RESTA: 3600
      run_parameters:
        DTMIN: 1800
        DTSYNC: 30
        DTEMIS_DELAY: 10
        DTMIN_FINAL: 10
        RSEED: 1234567
        meteorological_parameters:
          CREA_METEO: 2
        emission_parameters:
          EMIPAR: 1
        particles_simulation_parameters:
          ICOARE: 1
        concentration_parameters:
          SIN_CALCON: 30
          SFREQ_CALCON: 3600
          MED_CALCON: 3600
          CAMP_CALCON: 30
          NUMMAT: 10
          SPECIE: [1, 2, 3, 4, 5, 6, 7, 8, 1, 2]
          NUMSORG: [-1, -1, -1, -1, -1, -1, -1, -1, 5, 5]
          VETSOU: [[-1], [-1], [-1], [-1], [-1], [-1], [-1], [-1], [911-915], [911-915]]
          output_matrixes_names:
            ["PM10", "PM25", "NOX", "SO2", "BAP", "NMVOC", "CO", "C6H6", "PM10_parchi", "PM25_parchi"]
        deposition_parameters:
          IDEPO: 0
          IWETD: 0
  workflow:
    - fair_chain_start -> data_load_from_surfpro
    - fair_chain_start -> data_load_from_emma
    - fair_chain_start -> data_load_from_tim2par
    - fair_chain_start -> data_load_for_spray

    - data_load_from_surfpro -> spray_task
    - data_load_from_emma -> spray_task
    - data_load_from_tim2par -> spray_task
    - data_load_for_spray -> spray_task

    - spray_task -> data_store

    - data_store -> fair_chain_end
```