

AGORA Software Integration into FCCSW

Repository strategy, collaborative development
and a practical migration roadmap

01 REUSE

Build on Key4hep / FCCSW

02 INTEGRATE

Maintain a coordinated IHEP workspace

03 CONTRIBUTE

Send validated changes upstream

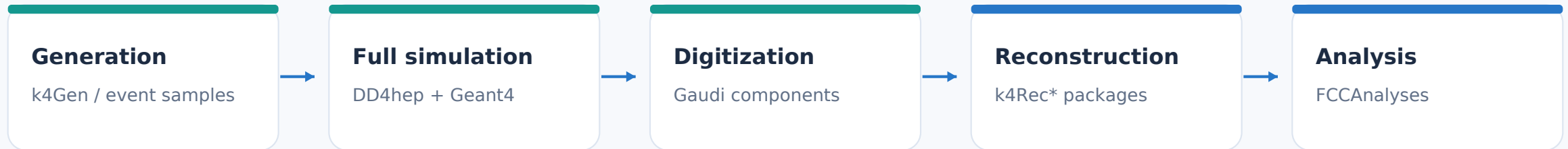
DISCUSSION PROPOSAL

12 October 2026

IHEP / AGORA Software

Why align AGORA with FCCSW?

FCC Software is a collection of Key4hep packages, detector models and processing workflows.



Existing foundations

- Core: Gaudi, EDM4hep / podio, DD4hep, Geant4
- Packages: shared geometry, digitization and reconstruction
- Workflows: detector-specific configurations and validation

AGORA contributions

- Detector description and experiment-specific details
- Digitization, tracking and calorimeter algorithms
- Reproducible end-to-end processing and physics checks

DESIGN RULE

Avoid building a parallel framework. Keep AGORA-specific elements modular and upstream-compatible.

Three initial repositories

Start with three upstream repositories; expand only when a clear functional boundary emerges.

WORKFLOW & SCRIPTS

FCC-config

Configuration files and job options for the event processing chain.

- AGORA detector/run configurations
- Simulation and reconstruction job options
- Reproducible end-to-end examples

[View upstream repository](#) ↗

DETECTOR GEOMETRY

k4geo

DD4hep geometry models, compact XML, plugins and detector tests.

- Port AGORA geometry in stages
- Validate geometry, materials and IDs
- Run detector-level simulation checks

[View upstream repository](#) ↗

TRACKING / DIGITIZATION

k4RecTracker

Gaudi tracker and vertex reconstruction components.

- Migrate relevant tracker algorithms
- Adapt EDM4hep input/output collections
- Test digitization and tracking quality

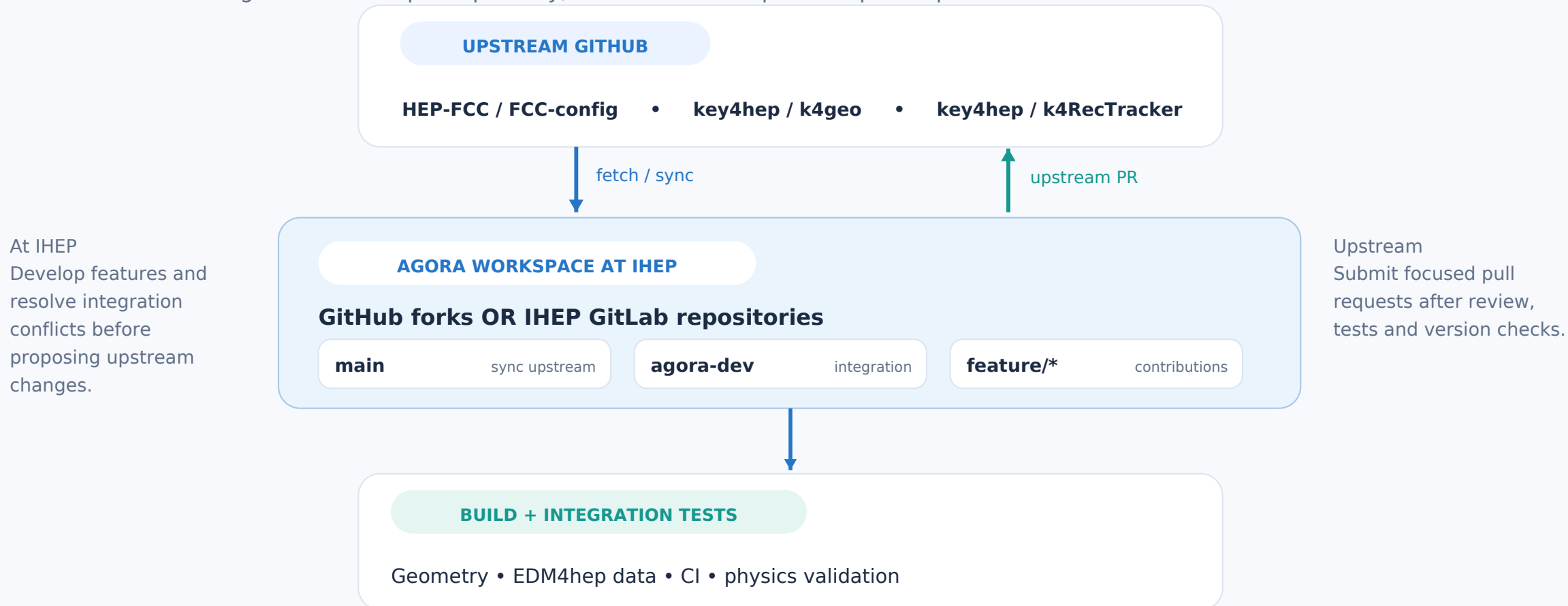
[View upstream repository](#) ↗

LATER, IF NEEDED

k4RecCalorimeter for calorimetry; a small AGORA-specific repository for code not suitable for an upstream package.

Develop locally. Contribute upstream.

One coordinated integration branch per repository; small validated upstream pull requests.



Hosting note: a GitLab copy is not a GitHub fork; retain a GitHub remote to submit upstream PRs.

A staged technical migration

Validate the existing FCC workflow before replacing components with AGORA implementations.

P0

Reproduce an FCC detector chain

Choose an existing detector example; pin Key4hep environment; record commands, inputs and outputs.

Working baseline**P1**

Integrate AGORA geometry

Port DD4hep models to k4geo; test materials, detector IDs, geometry overlaps and ddsim.

AGORA simulation**P2**

Connect digitization and reconstruction

Adapt EDM4hep collections; integrate tracking; extend calorimetry in the appropriate package.

Reco prototype**P3**

Validate and contribute

Automate build/regression tests, compare physics observables and prepare upstream pull requests.

Tested contributions

Decisions and immediate next steps

Agree the minimum collaboration model, then start with one verifiable full-chain example.

DECISION	PROPOSED DIRECTION	OUTCOME
Hosting	One canonical writable home; prefer a GitHub AGORA org. IHEP GitLab may provide CI/mirroring.	Choose host
Repository scope	Start from the three agreed repositories; expand only when justified.	Confirm forks
Branch & review	Keep main upstream-aligned; use agora-dev + feature branches and integration checks.	Agree policy
Pilot workflow	Pick an FCC-ee reference workflow and a pinned Key4hep environment.	Name example
Responsibilities	Appoint a maintainer per repository and identify first upstream PR candidates.	Assign owners

FIRST ACTIONS

Create repositories → assign maintainers → reproduce one baseline chain → open migration issues → prepare the first small PR.