

ePIC, sPHENIX, AND STAR COLLABORATIVE TOOLS: Phase I progress report

Alexander Undrus

September 23, 2026

CRISP and Invenio: two important collaborative tools

CRISP

Collaboration directory and workflow service

Maintains people, institutions, groups and roles.
Supports documents, tasks, meetings and workflows.
Serves the ePIC, sPHENIX and STAR phonebooks.

CRISP is a BNL-developed system that provides the management backend for experimental phonebooks and shift tracking tools.

Invenio Framework is free, open-source software for building large-scale digital repositories, digital libraries and research data-management systems. It is designed for 100+ million records and petabytes of files.

CERN developed Invenio and uses it in core repository services, including the CERN Document Server.

InvenioRDM

Research repository and data management service

Stores, organizes, shares and preserves research outputs and metadata.
BNL operates instances for sPHENIX, ePIC and RHIC Data Preservation.

Package inventory and CI/CD readiness

Both products have the packaging foundations needed for Phase I CI/CD demonstrations.

<u>HSF requirement</u>	<u>CRISP</u>	<u>InvenioRDM and packages</u>
Source, versions and maintainers	BNL repo includes the generic branch (branches for 3 collaborations are private). June overview credits nine contributors.	BNL deployment repository plus public upstream packages. BNL instances run version 13; upstream community contributes.
License and documentation	MIT license with installation, build, deployment and rollback procedures.	Reusable components have a MIT license, standard Python installation, guides, and published online documentation.
Tests and CI	Build, GraphQL, database-backed health and restoration checks are demonstrated.	Records package has 117 test files; OAuth has 19. Both have upstream CI workflows.
Possible improvements	Prominent quick start, contribution guidance and broader coverage.	Application repository license, quick start, tests and support ownership.

HSF - [HEP Software Foundation](#) - facilitates common software and computing efforts

Later slides show the CI/CD demo

Phase I assessment and a possible Phase II CI/CD track

Phase I: assess and demonstrate

(follow link to access)

Possible Phase II CI/CD items — scope TBD

Install experimental CRISP and Invenio instances

Develop Gitea triggers and runners

Exercise changes, tests and rollback locally

Package releases, extend integration tests,
Invenio transition to v14

Assess modularity and experiment dependencies

Explore CI across three CRISP branches

Develop the implementation and resource plan

Establish production operations and support

PHASE I: NEARLY COMPLETE
On track to finish by September 30

Why CI/CD is useful for these services

CI checks changes automatically. CD provides a repeatable way to deliver tested versions.

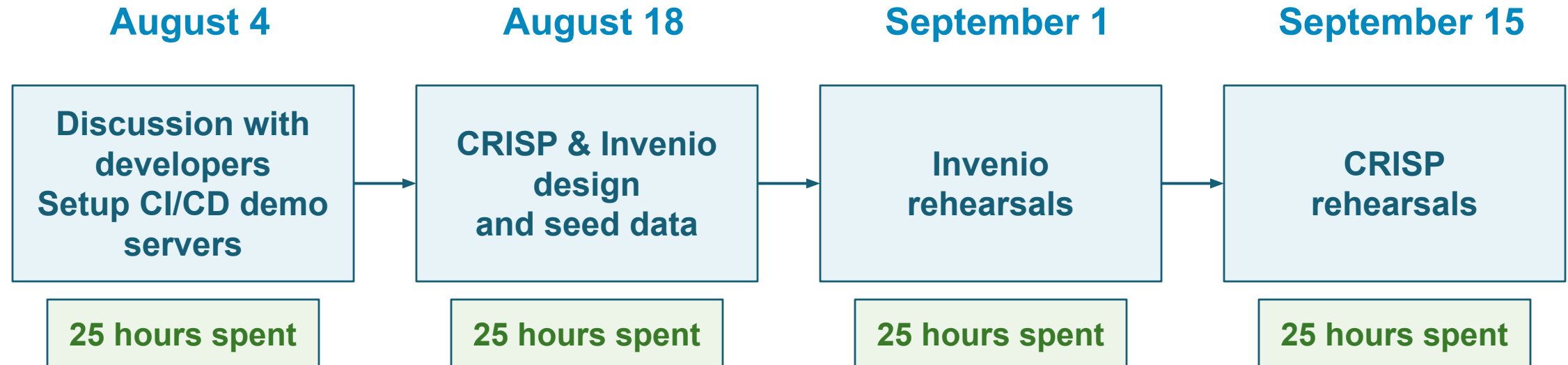
Service challenge	What CI/CD adds
Small changes can break a working service	Checks before release, including database access
Three CRISP branches and Invenio customizations	Consistent checks for each supported variant
Several people maintain the services	Recorded procedures, evidence and restoration

Manual operation remains possible. Repeatable checks reduce reliance on individual knowledge.

CI/CD helps preserve BNL's software investment by capturing how its packages should behave, how to build and deploy them, and how to verify future changes.

It complements documentation, maintained repositories, and clear ownership.

Progress from August to September

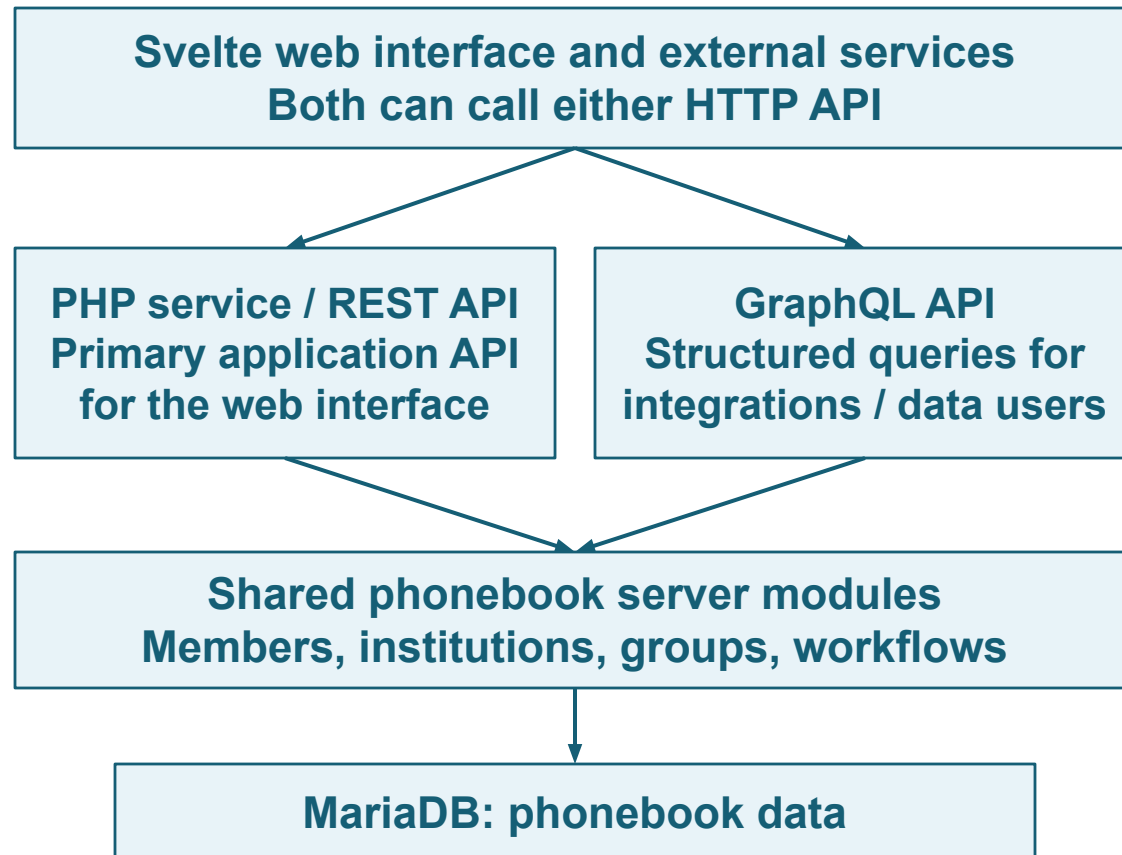


Mock-demo work: 100 hours spent / 100 hours planned

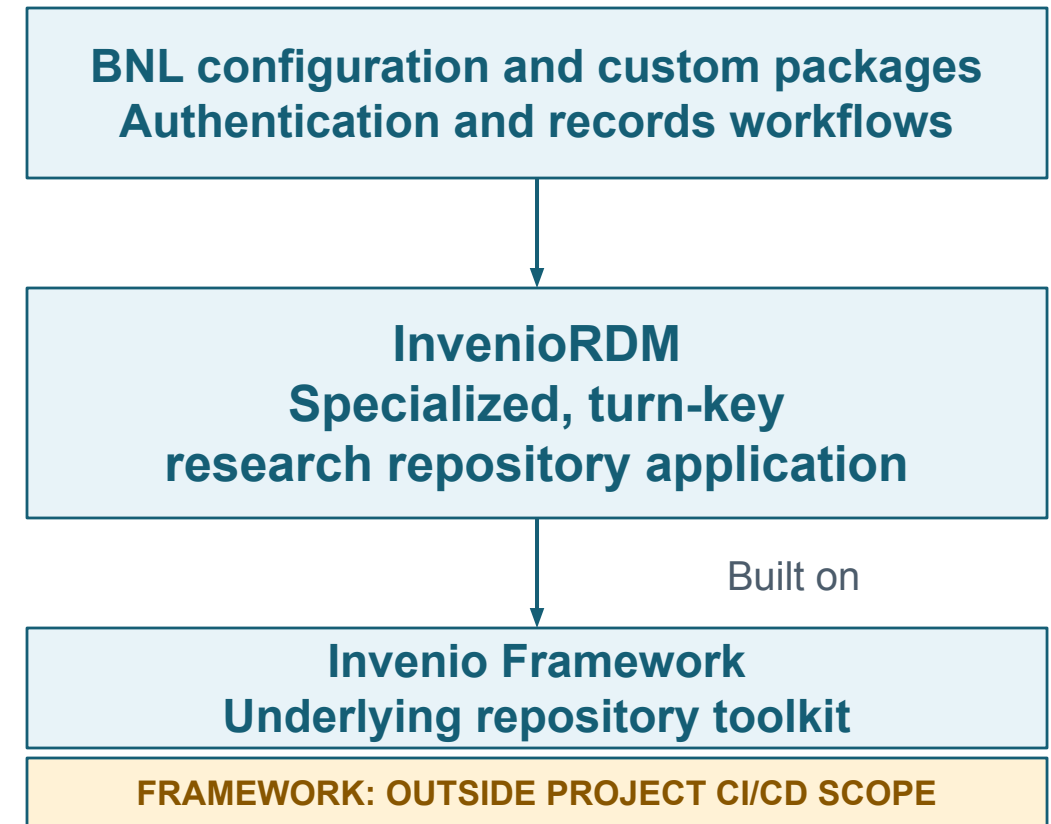
Production implementation will require substantially more effort.
A detailed implementation and support estimate is in progress.

Software layers and CI/CD project scope

CRISP



Invenio



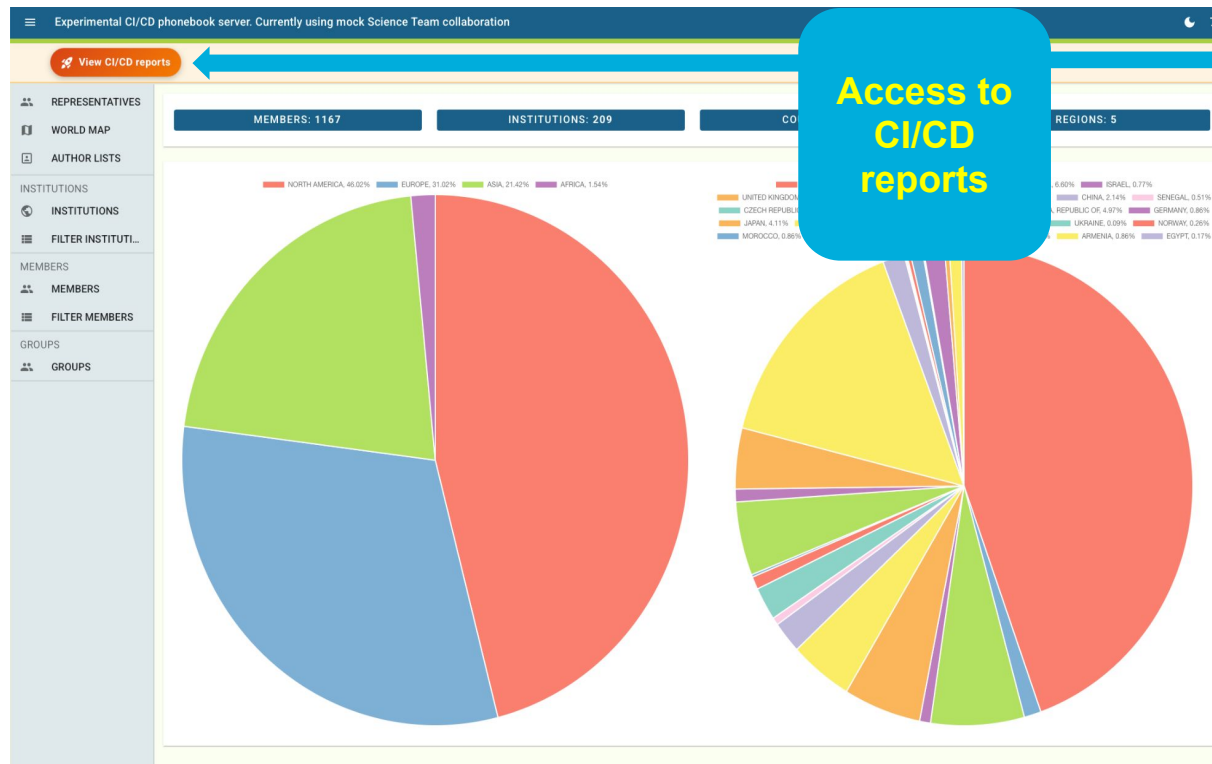
[June CRISP presentation: Collaborative Research and Information Sharing Platform](#)

[June InvenioRDM presentation: Collaborative Tools for EIC and RHIC: InvenioRDM](#)

Two demo CI/CD servers created (on internal BNL network)

CRISP demo server demonstrates the generic version with the fictional ScienceTeam collaboration (use Guest access to enter)

Invenio demo server runs InvenioRDM v13 with synthetic metadata records and direct access to the run reports.



ePIC Communities My dashboard undrus@...

InvenioRDM CI/CD - V13

Search

CI/CD demonstration deployment:

[View CI/CD reports](#)

Recent uploads

January 1, 2024 (v1) Presentation (Slides) Restricted

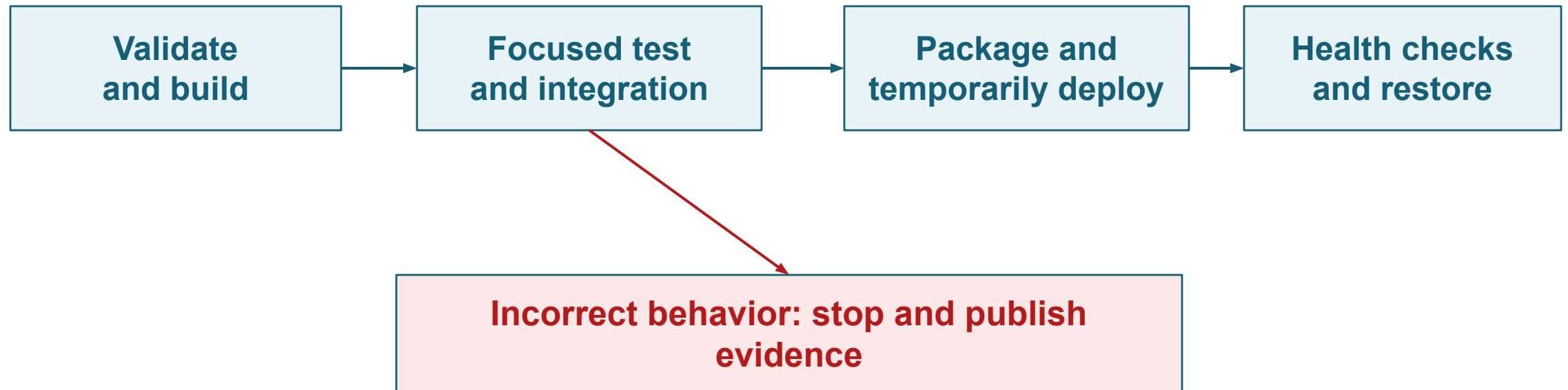
[CI/CD demonstration record 1](#)
Researcher, Demo

Synthetic metadata-only record generated for the InvenioRDM CI/CD demonstration environment.
Part of ePIC Public CI/CD Demonstration

Uploaded on August 28, 2026

CRISP demonstration pipeline

compiles Svelte components and bundles JavaScript into browser assets with npm / Webpack before testing and packaging the generated runtime.



Nine stages. A pipeline stops at the failed stage.

Phase II: develop Gitea triggers and runners for these procedures.

CRISP: example of a web-accessible CI report

CRISP CI/CD verification runs

Each row is one verification run. Stage icons: ✓ passed, ✗ failed, ○ bypassed. Updated 2026-09-14 20:27:09 UTC.

Run	Date (UTC)	Change	Release	Mode	Result	Stages	Evidence
20260914T202615Z-1237418	2026-09-14 20:27:09 UTC	Intentionally failing code change	demo-20260914T202615Z-1237418	build, deploy, health check, and rollback	FAILED	✓ ✓ ✓ ✗ ○ ○ ○ ○ ○ ○ 3/9 stages passed — failed: Focused change test	details
20260914T202512Z-1236430	2026-09-14 20:26:15 UTC	Code change	demo-20260914T202512Z-1236430	build, deploy, health check, and rollback	PASSED	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ 9/9 stages passed	details
20260914T202322Z-1235426	2026-09-14 20:24:31 UTC	Configuration change	demo-20260914T202322Z-1235426	build, deploy, health check, and rollback	PASSED	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ 9/9 stages passed	details

September 14 runs: configuration and code changes pass. The intentional regression fails as expected.

CRISP: example test results

CRISP CI/CD verification evidence

Overall result: **PASSED**

Change under verification

Code diff or changeset description

```
--- a/src/utils/non-breaking-space.js
+++ b/src/utils/non-breaking-space.js
@@ -2,3 +2,3 @@
 export const non_breaking_space = ( str ) => {
-   return str.replace( /\./g, '~' ).replace( ' ', '~' );
+   return str.replace( /\./g, '~' ).replace( / /g, '~' );
 }
 \ No newline at end of file
```

Pipeline stages

Stage	Result	Summary	Seconds	Log
Source validation	PASSED	PHP syntax, scripts, and locked frontend metadata are valid.	3	log
Change staging	PASSED	The change is isolated in a clean build workspace.	0	log
Frontend build and static validation	PASSED	Install locked dependencies, compile the production frontend, lint generated PHP, and verify required runtime outputs.	51	log
Focused change test	PASSED	Scenario-specific behavior is verified before deployment.	0	log
GraphQL integration test	PASSED	The generated runtime answers a database-backed GraphQL query.	1	log
Versioned runtime artifact	PASSED	Generated public runtime files are archived and checksummed.	4	log
Reversible deployment	PASSED	The runtime is backed up, replaced, and services restarted.	1	log
Post-deployment health checks	PASSED	UI, asset, GraphQL, login, and database checks pass.	1	log
Rollback and service restore	PASSED	The original runtime is restored and services returned active.	1	log

Informative title

Changeset description

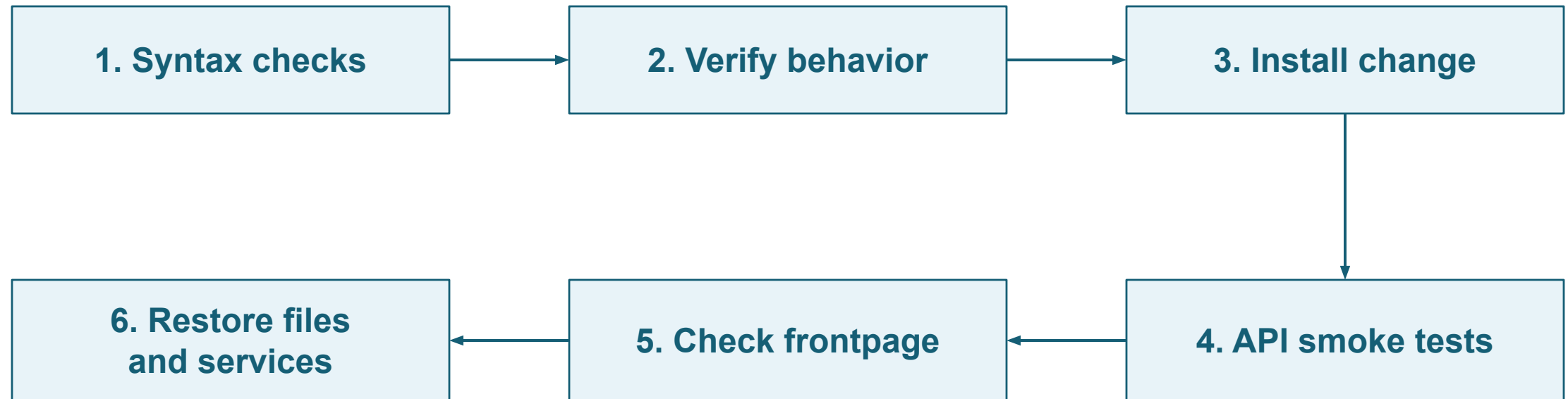
Color coded results:
9 / 9 stages passed

Detailed logfile

```
----- End inline test -----
Behavior under test: replace every ordinary space with "~" and append "~".
Input:   "A B C."
Expected: "A~B~C~."
Actual:  "A~B~C~."
PASS: the changed formatter produces the required output.
-----
Result: PASSED (command status 0)
Duration: 0 seconds
Finished: 2026-09-14 20:26:06 UTC
```

Invenio demonstration pipeline

CRISP and Invenio each need a custom pipeline: their software, checks and installation steps differ.



The Invenio demo does not run compilations. It checks and replaces selected files in an existing installation.

Invenio: example of a web-accessible CI report

InvenioRDM CI/CD verification runs

Each row is one verification run of a change (a future Gitea pull request). Updated 2026-09-16 20:34:32 UTC.

Run	Date (UTC)	Change	Release	Mode	Result	Stages	Evidence
20260916T203403Z	2026-09-16 20:34:03 UTC	CI/CD demonstration release marker	demo-20260916T203403Z	install and rollback	PASSED	 (6/6 stages passed)	details
rdm-records-intentional-failure-20260916T194443Z	2026-09-16 19:44:52 UTC	invenio-rdm-records package intentional-failure rehearsal	rdm-records-intentional-failure-20260916T194443Z	validation only	FAILED	 (1/6 stages passed — failed: Package code verification)	details
rdm-records-20260916T194357Z	2026-09-16 19:44:42 UTC	invenio-rdm-records package rehearsal	rdm-records-20260916T194357Z	install and rollback	PASSED	 (6/6 stages passed)	details

Configuration and package rehearsals pass. The negative control stops before installation.

Possible Phase II CI/CD track — scope TBD

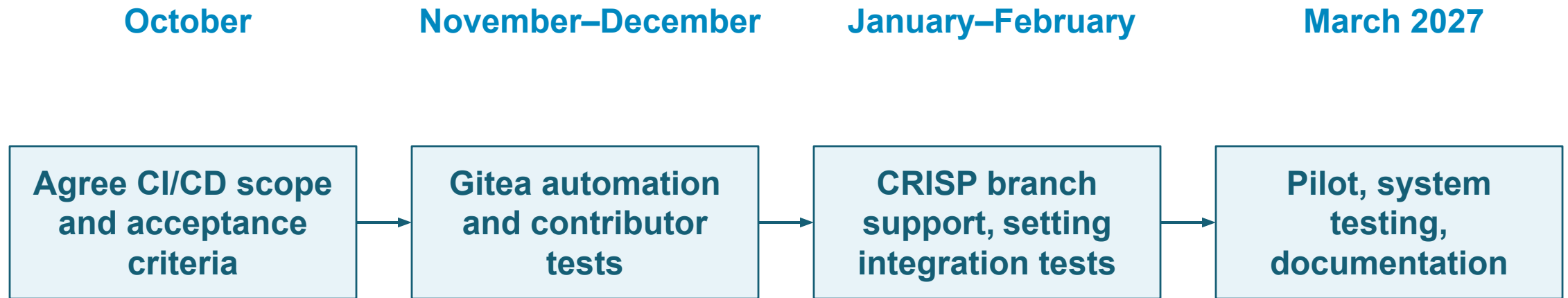
CI/CD is one possible track within the broader Phase II supported-services plan.

Possible item	People needed
Gitea triggers, runners and releases	DevOps engineers
CRISP branch design and test fixtures	CRISP developers and experiment maintainers
Invenio configuration and integration tests, facilitate transition to v14	Invenio developers and DevOps engineers
Monitoring, recovery procedures and Operations Handbook	DevOps engineers with application developers

Detailed CI/CD implementation and ongoing-support effort estimates are in progress.

Broader Phase II priorities, tool integrations and ownership remain TBD with the experiments.

Possible Phase II CI/CD milestones



Possible March 2027 delivery: a CI/CD pilot for CRISP and Invenio production instances, with automated checks and recovery rehearsal.

Schedule and delivery are subject to the upcoming review, agreed scope and resources.

Conclusion: Phase I results and Phase II planning

Phase I is nearly complete and on track for September 30.

Demonstrated

Both demo CI/CD instances can test changes, reject known errors, temporarily deploy updates and restore the original files.

Phase II CI/CD track

A possible CI/CD track can turn the demonstrations into maintained pipelines and preserve BNL software knowledge.

Benefits for BNL physics experiments

Custom CI/CD pipelines can make updates safer and easier to repeat. They can support reliable collaboration services across ePIC, sPHENIX and STAR.

**CI/CD is one possible part of the broader Phase II plan.
Detailed scope, ownership and effort estimates remain to be agreed.**

Backup slides

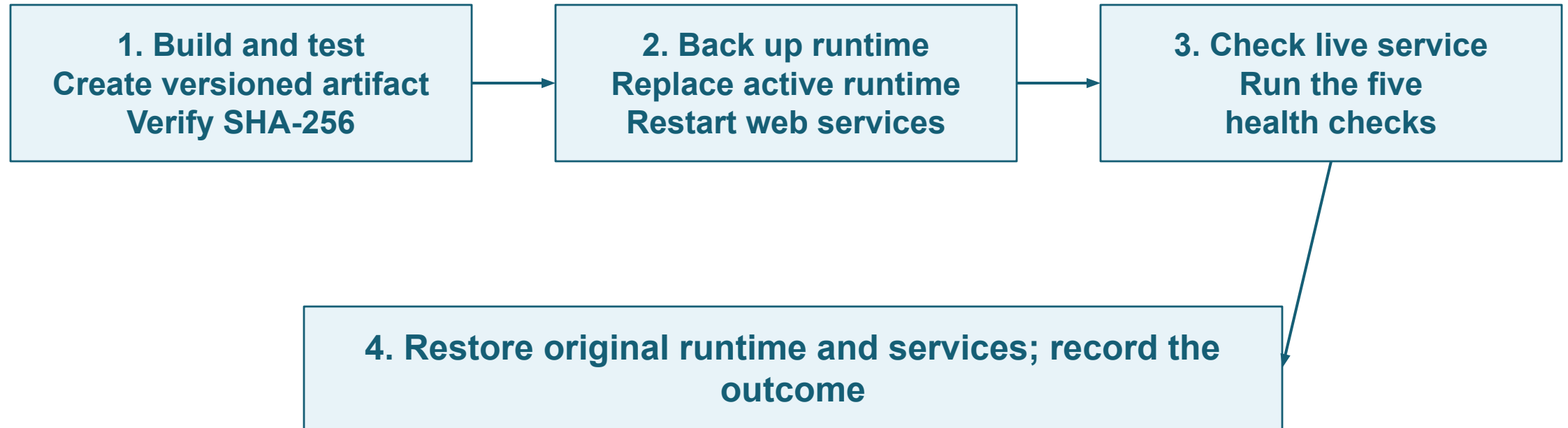
Backup: what the CRISP tests check

Check	What is exercised and expected
Source and generated build	Syntax checks, then compile Svelte and bundle JavaScript with npm / Webpack
Configuration change	Generated service configuration is valid and reports version 3.1
Focused formatter test	Input “A B C.” must produce “A~B~C.~”; the bad formatter fails this check
GraphQL integration	Isolated PHP server returns Query for { __typename }, using the database context
After temporary deployment	Page, localization, GraphQL, guest token and authenticated institution request

The negative control is deliberately wrong: rejection before deployment is the expected result.

Backup: CRISP temporary deployment and rollback

A passing candidate temporarily replaces the demo runtime, then the original is restored.



The checkout, database and external secret file remain unchanged.

A regression rejected before deployment has nothing to restore; rollback is marked skipped.

Backup: why CRISP has 9 stages and Invenio has 6

The count reflects the demonstrated change scope and how each report groups its work.

CRISP: build and replace a runtime

1. Source validation
2. Change staging
3. Frontend build and static validation
4. Focused change test
5. GraphQL integration test
6. Versioned runtime artifact
7. Reversible deployment
8. Post-deployment health checks
9. Rollback and service restore

Invenio: install selected files

1. Syntax checks
2. Configuration/template or package verification
3. Change install
4. API smoke tests
5. UI verification
6. Rollback and service restore

CRISP compiles Svelte and bundles JavaScript, then packages the generated runtime. Invenio uses an existing installation. It reports API and UI checks as separate stages.

Both reject bad changes before installation and restore originals after successful deployment rehearsals.

Backup: Invenio repositories and demonstration clones

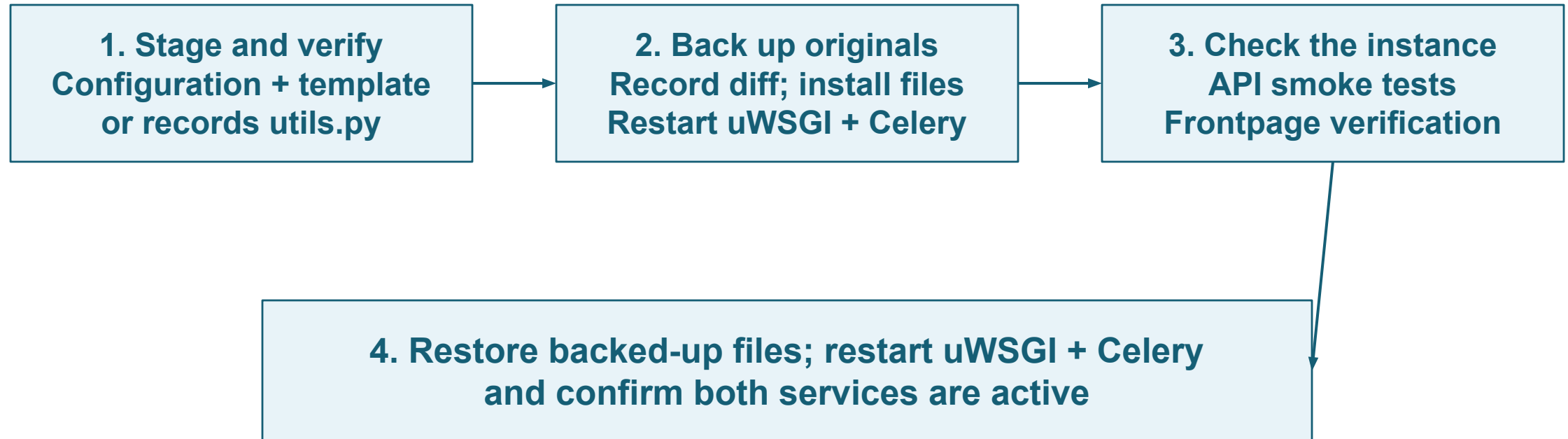
Demo clones retain minor instance changes, including HTML templates.

Component	Existing repository	Demo clone
Records	<u>invenio-rdm-records</u>	<u>invenio-rdm-records_CICD</u>
Authentication	<u>invenio-oauthclient</u>	<u>invenio-oauthclient_CICD</u>
Application	<u>inveniordm</u>	<u>Invenio_CDCCD</u>

Gitea triggers and runner development follow in Phase II.

Backup: Invenio temporary deployment and rollback

Install only the validated files for the rehearsal; restore their saved originals afterward.



The intentional failure is validation-only: no package file is installed.

File-level deployment and restoration are demonstrated; database schema recovery remains separate.

Backup: what the Invenio tests check

Check	What is exercised and expected
Configuration and template	Validate required settings and frontpage structure for a release-marker change
Records helper: normal change	Move the “no value found” marker outside the helper, keeping the same answers
Known-input test	Supply off, zero and empty text. Check that the helper returns these values
Deliberately broken change	Replace “value exists” with “value is true”. The test must catch the wrong answer
After temporary installation	Check community / record APIs and the frontpage CI/CD banner or marker

These are targeted checks of configuration, a package utility and a running instance.

Backup: Invenio: example test results

Pipeline stages

Stage	Result	Summary	Seconds	Log
Syntax checks	PASSED	Staged utils.py compiles successfully.	0	syntax-checks.log
Package code verification	PASSED	Focused ChainObject package behavior and staged candidate verified.	10	configuration-verification.log
Change install	PASSED	Staged utils.py installed; uwsgi.service and celery.service restarted.	20	change-install.log
API smoke tests	PASSED	Community and record endpoints passed.	2	api-smoke-tests.log
UI verification	PASSED	Frontpage responds and displays the CI/CD banner.	0	ui-verification.log
Rollback and service restore	PASSED	Original package file restored; uwsgi.service and celery.service restarted.	12	rollback.log

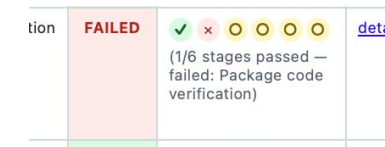
The installation is restored to its original configuration and the uWSGI and Celery services are restarted before the run completes.

Records-package rehearsal, September 16. All six stages passed, including service restore.

Backup: Invenio test: “off” must not become “on”

A function in a Python module checks several sources in order and returns the first available value, even if that value means “off”.

The test supplies known values and checks the answer



Test step	Simple example
Set up two sources	First source says “off”. The next source says “on”.
Check the correct answer	The helper must return “off”, because a value was provided.
Try the deliberately broken change	It ignores “off”, continues searching and returns “on”.

The changed rule confuses “a value exists” with “the value is true”.

**The test detects the wrong answer and blocks installation.
It also checks zero and empty text, which are valid answers too.**