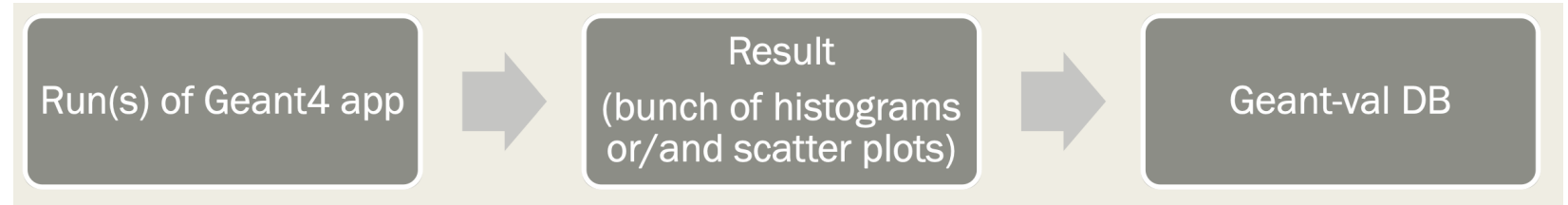# Geant-Val and Our Validation: Time for a Change?

Geant4 Collaboration meeting

Dmitri Konstantinov
University of Virginia/CERN

Reminder

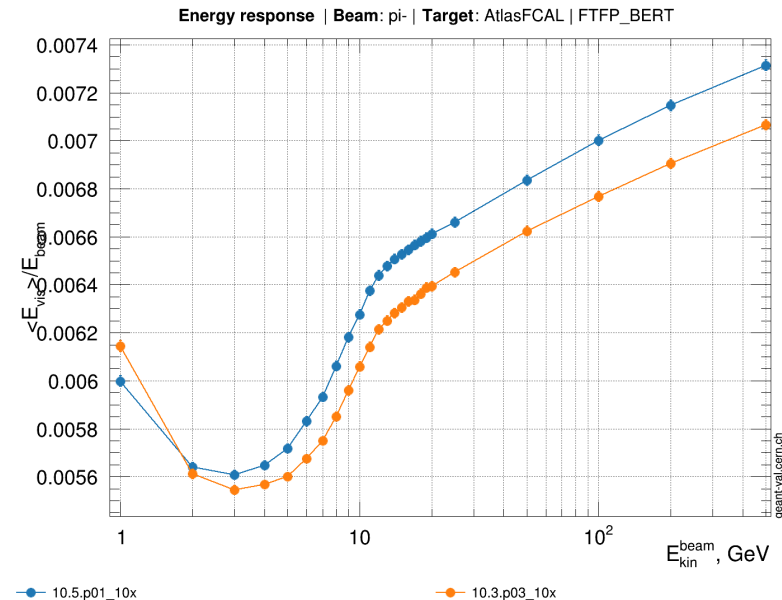| Run(s) of Geant4 app	⇒	Result (bunch of histograms or/and scatter plots)	⇒	Geant-val DB

Geant-Val is a validation web application for Geant4.

It provides consistent storage for histograms or scatter plots of test results.

Users can compare results produced by different Geant4 versions, physics lists (or models).
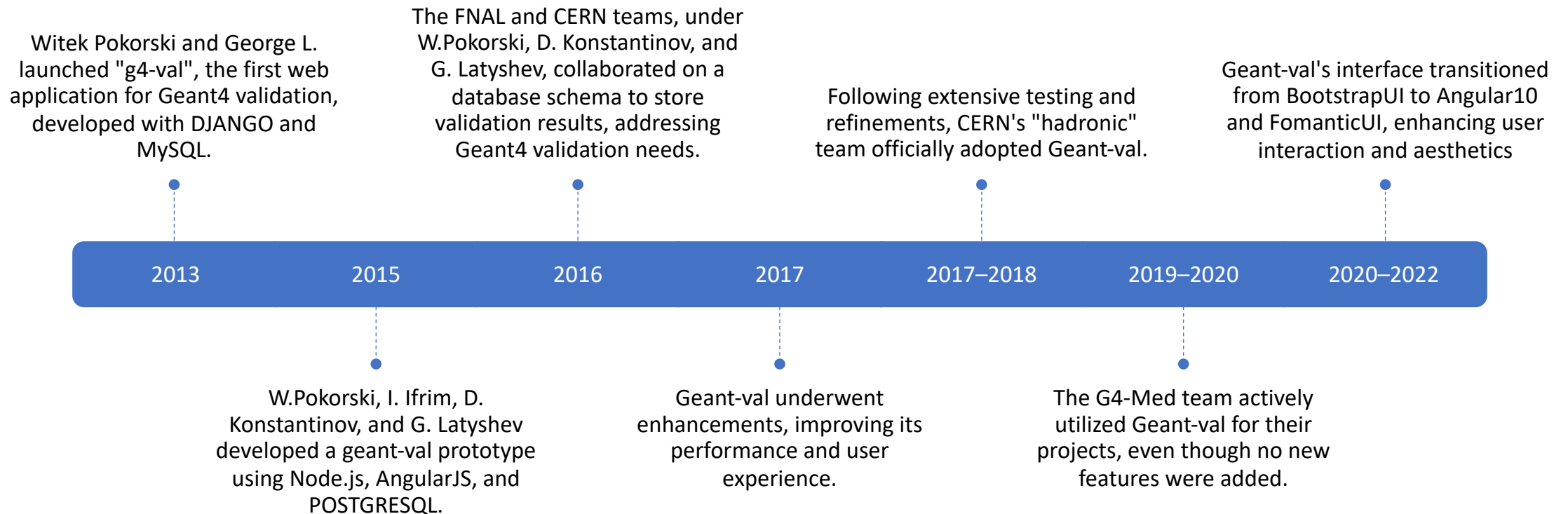
Try it - Geant-Val website

Geant-Val: Steady Performance

- **Revisiting Previous Content**: Many parts of today's presentation will echo the slides shared during our last collaboration meeting. This repetition is intentional, given the unchanged status and importance of Geant-Val's aspects.

- **Key Departure**: Grigory Latyshev, one of our primary developers, has moved on, creating a significant gap in the development process.

- **Solo**: As of now, I am only one who can handle web development for Geant-Val.

- **Status Update**: Over the past year, there have been no new developments in the Geant-Val web application and its auxiliary tools.

- **Reliability Affirmed**: Despite lack of updates, Geant-Val remains stable and continues to serve its purpose.

- **The Path Forward**: Recognizing the importance of Geant-Val, it's crucial to plan its future…

History of Geant-Val

Witek Pokorski and George L. launched "g4-val", the first web application for Geant4 validation, developed with DJANGO and MySQL.

The FNAL and CERN teams, under W.Pokorski, D. Konstantinov, and G. Latyshev, collaborated on a database schema to store validation results, addressing Geant4 validation needs.

Following extensive testing and refinements, CERN's "hadronic" team officially adopted Geant-val.

Geant-val's interface transitioned from BootstrapUI to Angular10 and FomanticUI, enhancing user interaction and aesthetics

| 2013 | 2015 | 2016 | 2017 | 2017–2018 | 2019–2020 | 2020–2022 |

W.Pokorski, I. Ifrim, D. Konstantinov, and G. Latyshev developed a geant-val prototype using Node.js, AngularJS, and POSTGRESQL.

Geant-val underwent enhancements, improving its performance and user experience.

The G4-Med team actively utilized Geant-val for their projects, even though no new features were added.

Geant-Val Users & Their Use-Cases

- **Hadronic Validation**: Alberto Ribon uses Geant-Val to validate monthly reference releases in the "hadronic" domain.

- **Full Standalone Simulations of TestBeam of Hadronic Calorimeters:** Lorenzo Pezzotti has integrated simulations of hadronic calorimeters to compare with real test beam results. He's preparing to work on a full standalone simulation of the test beam for the HGCAL of the CMS experiment. (Report from Lorenzo - 26 Sept 2023, 11:40)

- **Benchmarking in Medical Physics**: The G4Med group, led by Susanna, uses Geant-Val in this domain, covering both stable and beta releases. (Report from Susanna - 27 Sept 2023, 16:30)

- **Electromagnetic Validation**: Majority of EM tests are integrated into Geant-Val, but they're not used consistently. They're run occasionally upon requests from Mihaly.

- **"Thin Target" Tests**: Julia Yarba (FNAL) is working on adding "thin target" tests.

- **FastSim Performance Validation**: Marc and Igor are adding tests to validate GFlash's performance. See Igor's presentation about his experience with geant-val today - 25 Sept 2023, 14:30

The Geant-config-generator Workflow

Geant4 applications come with diverse configurations, output formats, and naming conventions.

Instead of altering these applications, we wrote the Geant-config-generator which bridges the gap.

The Geant-config-generator is a Python utility designed to simplify and manage Geant4 physics tests.

Key Functionalities:

- **Test Configuration**: Transparently creates configurations using templates and steering files.

- **Job Submission**: Directly submits tasks to the HTCondor batch system (extendable to work different batch systems).

- **Result Processing**: Parses and merges job results.

- **Metadata Enhancement**: Appends missing metadata to ensure comprehensive data (missing in Geant4 application's output).

- **Parameter Communication**: Relays Geant4 application parameters to geant-val.

- **JSON File Creation**: Generates input files tailored for geant-val.

Outcome:

- Running the Geant4 application using **Geant-config-generator** yields histograms and scatter plots, which are systematically stored in the Geant-val database.

Geant-Val: Recognizing Imperfections

- **Q**: Am I satisfied with the current state of geant-val?
  A: No!

Technical Improvements Needed:

- **Update to Angular**: Migrate to the latest Angular2 version (current LTS).

- **System Upgrade**: Move away from the centos7 VM.

- **Hosting Exploration**: Investigate the use of various CERN web services for hosting geant-val.

- **Web Interface Enhancements**: Improve and develop new web interfaces to streamline the validation process.

- **Upload results**: Improve upload security.

- **Database Concerns**:
  The PostgreSQL database, as part of our stack, has shown limitations. Adapting it to new user requests has been challenging, often requiring extensive changes. Could try NoSQL DB - MongoDB.

And this is not full list....

Steps for Improvement

Here's what we need to do better:

- **Put Tests Together**.

Right now, our validation applications are spread out in different places. Some are even private. You can find them at these links:

https://gitlab.cern.ch/geant4/geant4-dev/-/tree/master/examples
https://gitlab.cern.ch/geant4/geant4-dev/-/tree/master/tests
https://gitlab.cern.ch/geant4/g4tests-verification
https://github.com/lopezzot
https://gitlab.cern.ch/GeantValidation/geant-validation-tests

Steps for Improvement

Here's what we need to do better:

- **Put Tests Together**.

Why?

- Centralizing tests ensures that they remain accessible for future developers and teams, **preserving the collective knowledge and expertise**.

- Centralizing tests allows for **better version control.**

- With all tests in one place, it's easier to **maintain comprehensive documentation**.

- Having tests in one place promotes **standardization.**

- With tests in one place, it is **easier to conduct reviews**.

- Centralized tests can be more easily integrated into **continuous integration (CI)** pipelines.

Steps for Improvement

Here's what we need to do better:

- **Put Tests Together.**

How?

GitHub might be the right platform for this.

- Create an organization, perhaps named 'geant4-validation-apps'.
- It's currently free.
- Could have a distinct repository for each test (if needed)
- Some repositories can be open for external contributors.
- Some can be kept private until the associated paper is published.

Steps for Improvement

Here's what we need to do better:

Introduce and Follow Standards for Tests:

- **Detailed README**: Every application should have a comprehensive README that guides users.

- **Documentation**: There should be a publication (better), or a PDF document associated with each application.

- **Verified Code**: Each application must undergo a thorough check to ensure it is properly written, and functions as intended.

- **Complete Toolset**: The application should include all the necessary scripts to produce the final results.

Steps for Improvement

Here's what we need to do better:

Improving Our Validation Process:

- **Addressing Post-release Bugs**: There have been cases where bugs were reported immediately after a release. Interestingly, we often had tests that could detect these bugs, but we didn't run them.

- **Regular Testing**: With various Geant4 WG define and execute a selected set of tests for monthly validation and before each release.

- **Structured Test Schedule**: It's essential to specify which tests we'll run monthly, which ones before a beta release, and which ones before an annual release.

- **Learning from Mistakes**: Each bug in Geant4 is an opportunity for improvement. For every error identified, we should create a test to catch it, ensuring its non-recurrence and enhancing Geant4's reliability.

Steps for Improvement

Here's what we need to do better:

- **Work Together.**

  We need to agree on how to test and do it as a team.

Steps for Improvement

Here's what we need to do better:

- **Work Together.**

How???

Steps for Improvement

Here's what we need to do better:

- **Work Together.**

The **G4-Med team** is a regular user of geant-val for their simulations.

Can we consider relocating G4-med tests to individual GitHub repositories?

Along with the move, it is a good opportunity to **review and refine** each application for better performance and clarity.

As we undertake this transition, we could prioritize **setting clear standards** for our tests, ensuring uniformity and quality.

Is it a feasible plan?

Steps for Improvement

Here's what we need to do better:

Work Together.

- Consider updating the geant-config-generator to be compatible with various batch systems, not just the CERN one.

- By making these adjustments, we can encourage labs with significant computing resources, like KISTI and FNAL, to participate in geant4 validation using their computing farms.

More thoughts.

Manpower Challenges in Geant4:

It's evident that Geant4 faces a manpower shortage in several areas, including documentation, examples, website, and now validation.

A potential solution could be to introduce a minimum requirement for the Geant4 annual census, perhaps setting it to 1-2 months.

And, of course, transparency: when tasks, progress, and challenges are visible to all, it encourages responsibility and care.

Hope to see relevant discussions after [Anna's report from Evolution Task Force .](#)

Closing Thoughts

These were just my suggestions on enhancing the Geant4 validation process, aiming for greater reliability, coverage, and robustness.

I'm confident that many of you also have valuable ideas to share.

Let's use our time together this week to share ideas and formulate a solid plan for a more effective Geant4 validation.

Thank you!