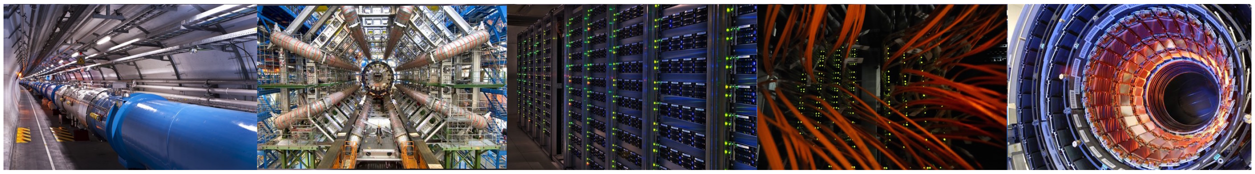

Strategic Plan for a Scientific Software Innovation Institute (S^2I^2) for High Energy Physics DRAFT

Peter Elmer (Princeton University)
Mike Sokoloff (University of Cincinnati)
Mark Neubauer (University of Illinois at Urbana-Champaign)

October 12, 2017

This report has been produced by the S2I2-HEP project (http://s2i2-hep.org) and supported by National Science Foundation grants ACI-1558216, ACI-1558219, and ACI-1558233. Any opinions, findings, conclusions or recommendations expressed in this material are those of the project participants and do not necessarily reflect the views of the National Science Foundation.

Executive Summary

Contributors

To add: names of individual contributors to both the text of this document and to the formulation of the ideas therein, through the workshops, meetings and discussions that took place during the conceptualization process.

Contents

1 Introduction

The High-Luminosity Large Hadron Collider (HL-LHC), is scheduled to start producing data in 2027 and extend the LHC physics program through the 2030's. Its primary science goal is to search for Beyond the Standard Model (BSM) physics, or study its details if there is an intervening discovery. Although the basic constituents of ordinary matter and their interactions are extraordinarily well described by the Standard Model (SM), a quantum field theory built on top of simple but powerful symmetry principles, it is incomplete. For example, most of the gravitationally interacting matter in the universe does not interact via electromagnetic or strong nuclear interactions. As it produces no directly visible signals, it is called dark matter. Its existence and its quantum nature lie outside the SM. Equally importantly, the SM does not address fundamental questions related to the detailed properties of its *own* constituent particles or the specific symmetries governing their interactions. The HL-LHC will record data from 100 times as many proton-proton collisions as did Run 1 of the LHC, and at twice the energy. Realizing the full potential of the HL-LHC will require large investments in upgraded hardware. Similarly, it will require commensurate investment in the research and development necessary to deploy software to acquire, manage, process, and analyze the data.

The magnitudes of the problems to be solved require approaches beyond simply scaling current solutions from today's technologies assuming Moore's Law and more or less constant operational budgets. The nature of computing hardware (processors, storage, networks) is evolving, the quantity of data to be processed is increasing dramatically, its complexity is increasing, and more sophisticated analyses will be required to maximize the HL-LHC physics yield. Developing and deploying sustainable software for the HL-LHC experiments, given these constraints, is both a technical and a social challenge. An NSF-funded, U.S. university-based Scientific Software Innovation Institute (S2I2) can play a leadership role in building the "software upgrade" needed to run in parallel with the hardware upgrades planned for the HL-LHC.

The two general purpose detectors at the LHC, ATLAS and CMS, are operated by collaborations of more than 3000 scientists each. U.S. personnel constitute about 30% of the collaborators on these experiments. Within the U.S., funding for the construction and operation of ATLAS and CMS is jointly provided by the Department of Energy (DOE) and the National Science Foundation (NSF). The NSF is also planning a major participation in the hardware upgrade of ATLAS and CMS for the HL-LHC. This will use the "Major Research Equipment and Facilities Construction" (MREFC) mechanism with a possible start in 2020. Funding for the LHCb experiment is provided only by the NSF. Computing resources (hardware and operations personnel) are generally provided in proportion to the number of collaborators from each country or region. In planning for the HL-LHC, it is critical that all parties agree (roughly) on the software goals and priorities, and that the efforts tend to complement each other. In this spirit, the HEP Software Foundation (HSF) began a planning exercise in late 2016 to prepare a Community White Paper (CWP). Its goal was to provide a roadmap for software R&D in preparation for the HL-LHC era which would identify and prioritize the software research and development investments required:

1. to achieve improvements in software efficiency, scalability and performance and to make use of the advances in CPU, storage and and network technologies;
2. to enable new approaches to computing and software that can radically extend the physics reach of the detectors;
3. to ensure the long term sustainability of the software through the lifetime of the HL- LHC.

The S2I2 conceptualization process ran in parallel with the CWP process and focussed on three additional goals:

1. to identify specific focus areas for S2I2 R&D efforts;

2. to build a consensus within the U.S. HEP software community for a common effort;
3. to engage with experts from related fields of scientific software development to identify areas of common interest and develop teams for collaborative work.

2 Science Drivers

Should focus on HL-LHC, but make reference to broader HEP at the end

In the past 200 years, physicists have discovered the basic constituents of ordinary matter and they have developed a very successful theory to describe the interactions (forces) between them. All atoms, and the molecules from which they are built, can be described in terms of these constituents. The nuclei of atoms are bound together by strong nuclear interactions. Their decays result from strong and weak nuclear interactions. Electromagnetic forces bind atoms together, and bind atoms into molecules. The electromagnetic, weak nuclear, and strong nuclear forces are described in terms of quantum field theories. The predictions of these theories can be very, very precise, and they have been validated with equally precise experimental measurements. The electromagnetic and weak nuclear interactions are intimately related to each other, but with a fundamental difference: the particle responsible for the exchange of energy and momentum in electromagnetic interactions (the photon) is massless while the corresponding particles responsible for the exchange of energy and momentum in weak interactions (the W and Z bosons) are about 100 times more massive than the proton. A critical element of the Standard Model (SM) is the prediction (made more than 50 years ago) that a qualitatively new type of particle, called the Higgs boson, would give mass to the W and Z bosons. Its discovery [1,2] at CERN's Large Hadron Collider (LHC) in 2012 was the last critical element of the SM to be confirmed experimentally.

The SM describes essentially all known physics very well, but its mathematical structure and some important empirical evidence tell us that it is incomplete. These observations motivate a large number of SM extensions, generally using the formalism of quantum field theory, to describe physics Beyond the Standared Model (BSM). For example, "ordinary" matter accounts for only 5% of the mass-energy budget of the universe while dark matter, which interacts with ordinary matter gravitationally, accounts for 27%. While we know something about dark matter at macroscopic scales, we know nothing about its microscopic, quantum nature, *except* that its particles are not found in the SM and they lack electromagnetic and SM nuclear interactions. BSM physics also addresses a key feature of the observed universe: the apparent dominance of matter over anti-matter. The fundamental processes of leptogenesis and baryongenesis (how electrons and protons, and their heavier cousins, were created in the early universe) are not explained by the SM, nor is the required level of CP violation (the asymmetry between matter and anti-matter under charge and parity conjugation). Constraints on BSM physics come from "conventional" HEP experiments plus others searching for dark matter particles either directly or indirectly.

The LHC was designed to search for the Higgs boson and for BSM physics – goals in the realm of discovery science. The ATLAS and CMS detectors are optimized to observe and measure the direct production and decay of massive particles. They have now begun to measure the properties of the Higgs boson more precisely to test how well they accord with SM predictions.

Where ATLAS and CMS were designed to study high mass particles directly, LHCb was designed to study heavy flavor physics where quantum influences of very high mass particles are manifest in lower energy phenomena. Its primary goal is to look for BSM physics in CP violation (asymmetries in the decays of particles and their corresponding antiparticles) and rare decays of beauty and charm hadrons. As an example of how one can relate flavor physics to extensions of the SM, Isidori, Nir, and Perez [3] have considered model-independent BSM constraints from measurements of mixing and CP violation. They assume the new fields are heavier than SM fields and construct an effective theory. Then, they "analyze all realistic extensions of the SM in terms of a limited number of parameters (the coefficients of higher dimensional operators)." They determine bounds on an effective coupling strength couplings of their results is that kaon, B_d, B_s, and D^0 mixing and CPV measurements provide powerful constraints that are complementary to each other and often constrain BSM physics more powerfully than direct searches for high mass particles.

The Particle Physics Project Prioritization Panel (P5) issued their *Strategic Plan for U.S. Particle Physics* [4] in late May, 2014. It was very quickly endorsed by the High Energy Physics Advisory Panel and submitted to the DOE and the NSF. The report says, *we have identified five compelling line of inquiry that show great promise for discovery over the next 10 to 20 years. These are the Science Drivers:*

- *Use the Higgs boson as a new tool for discovery*
- *Pursue the physics associated with neutrino mass*
- *Identify the new physics of dark matter*
- *Understand cosmic acceleration: dark matter and inflation*
- *Explore the unknown: new particles, interactions, and physical principles.*

The HL-LHC will address the first, third, and fifth of these using data acquired at twice the energy of Run1 and with 100 times the luminosity. As the P5 report says,

The recently discovered Higgs boson is a form of matter never before observed, and it is mysterious. What principles determine its effects on other particles? How does it interact with neutrinos or with dark matter? Is there one Higgs particle or many? Is the new particle really fundamental, or is it composed of others? The Higgs boson offers a unique portal into the laws of nature, and it connects several areas of particle physics. Any small deviation in its expected properties would be a major breakthrough.

The full discovery potential of the Higgs will be unleashed by percent-level precision studies of the Higgs properties. The measurement of these properties is a top priority in the physics program of high-energy colliders. The Large Hadron Collider (LHC) will be the first laboratory to use the Higgs boson as a tool for discovery, initially with substantial higher energy running at 14 TeV, and then with ten times more data at the High- Luminosity LHC (HL-LHC). The HL-LHC has a compelling and comprehensive program that includes essential measurements of the Higgs properties.

In addition to HEP experiments, the LHC hosts the one of world's foremost nuclear physics experiments. "The ALICE Collaboration has built a dedicated heavy-ion detector to exploit the unique physics potential of nucleus-nucleus interactions at LHC energies. [Their] aim is to study the physics of strongly interacting matter at extreme energy densities, where the formation of a new phase of matter, the quark-gluon plasma, is expected. The existence of such a phase and its properties are key issues in QCD for the understanding of confinement and of chiral-symmetry restoration." [5] In particular, these collisions reproduce the temperatures and pressures of hadronic matter in the very early universe, so provide a unique window into the physics of that era.

Summary of Physics Motivation: The ATLAS and CMS collaborations published letters of intent to do experiments at the LHC in October 1992, about 25 years ago. At the time, the top quark had not yet be discovered; no one knew if the experiments would discover the Higgs boson, supersymmetry, technicolor, or something completely different. Looking forward, no one can say what will be discovered in the HL-LHC era. However, with data from 100 times the number of collisions recorded in Run 1, and the collisions at twice the energy, the next 20 years are likely to bring even more exciting discoveries.

3 Computing Challenges

During the HL-LHC era (Run 4, starting circa 2027), the ATLAS and CMS experiments will record about 10 times as much data from 100 times as many collisions as they did in in Run 1. Already in Run 3 (starting circa 2021), the LHCb experiment will process data from 100 times as many collisions as it did in Run 1. The software and computing budgets for these experiments are projected to remain flat, and Moore's Law will not provide the required increase in computing power to enable fully processing all the data. Even assuming the experiments significantly reduce the amount of data stored per event, the total size of the datasets will be well into the exabyte scale; they will be constrained primarily by costs and funding levels, not by scientific interest. The overarching goal of a HEP S^2I^2 will be maximizing the return-on-investment [to upgrading the accelerator and detectors and to supporting the research program] to enable break-through scientific discoveries. The Community White Paper and S2I2 conceptualization processs identified many specific software challenges, but they may be broadly classified into three categories:

(i) developing new software implementations of existing algorithms to use resources more effectively;

(ii) developing qualitatively new algorithms, and their software implementations, to extend the physics reach of the detectors and/or to use resources more effectively;

(iii) developing the software required to take advantage of a heterogenous computing envinronment.

These categories can be illustrated by use cases.

(i) Over the past 15 years, more than 2000 CMS and ATLAS collaborators have written in excess of 10M source lines of code (SLOC) to support their activities. Almost all the algorithms were designed for single-threaded execution on "traditional" x86 CPUs. To be used moving forward, these must be adapted to run on multi-threaded, multi-core systems and heavily exploit vectorized operations. They must be re-designed and optimized to avoid cache misses.

(ii) Amongst possible approaches:

   - Machine learning algorithms may be able to replace computationally expensive parts of event simulation, pattern recognition, and calibration.
   - Highly parallel algorithms designed for GPUs and other massively parallel architectures may be able to replace (essentially) serial algorithms.

(iii) Most of HEP's computationally intensive work is now done on architecturally similar platforms explicitly designed and operated for our use. This is likely to change. We must *also* be able to use heterogenous commercial clouds, High Performance Computing (HPC) resources at national laboratories, etc., effectively.

The human and social challenges run in parallel with the technical challenges. All algorithms and software implementations are developed and maintained by flesh and blood individuals, many with unique expertise. *Working together as a coherent community, and engaging productively with the larger scientific software community, will be critical to the success of the enterprise.*

Software is ubiquitous in the process by which data is acquired, managed, processed and analyzed to produce physics results from experiments such as those at the LHC. This includes application-level (algorithmic) software to support for number of stages in the typical data flow:

- data acquisition (DAQ)
- high level trigger (HLT) - a software process, runs in (quasi-) real time to select a small subset of data to be processed offline
- calibration and alignment of the detectors

- reconstruction - pattern recognition and related data reduction algorithms, typically in common for an entire experiment
- data analysis - activities initiated by individuals or groups within the experiment to select data subsets and extract physics results

Also, and perhaps less obviously, simulations of detectors and physics are equally critical in preparing experiments and analyzing data. The number of nominal stages is similar. The CPU resources used to generate simulated data often match, or even exceed, those used to process real data offline. In addition, the scale of the (geographically distributed) computing infrastructure requires significant software development to support data, workflow and resource management.

Several major software sustainability challenges exist given the physics goals and 15-20 year time scale of HL-LHC: (1) the evolution of computing hardware (processors, storage, networks) and the need to find cost-effective solutions, (2) increased complexity of the data and/or detectors, and (3) increased sophistication of analyses required from larger datasets.

The challenges for processor technologies are well known [6]. While the number of transistors on integrated circuits doubles every two years (Moore's Law), power density limitations and aggregate power limitations lead to a situation where "conventional" sequential processors will be replaced by parallel architectures, with consequent implications for changes to the algorithms implemented in our software. Understanding how emerging architectures (from low power processors to parallel architectures like GPUs to specialized technologies like FPGAs) will allow HEP computing to realize the exponential growth in computing power required to achieve our science goals will be a critical element of any R&D effort. Similar challenges exist with storage and network at the scale of HL-LHC [7], with implications for the persistency and computing models and the software supporting them.

The software landscape itself is quite varied. The preparations for LHC have yielded important community software tools for data analysis like ROOT [8] and detector simulation GEANT4 [9, 10], both of which have been critical not only for LHC but in most other areas of HEP (and beyond). Other tools have been shared between some, but not all, experiments. Examples include the GAUDI [11] event processing framework, RooFit [12] for data modeling and fitting and the TMVA [13] toolkit for multivariate data analysis. Tools developed in collaboration with computer scientists include FRONTIER [14] for cached access to databases, XROOTD [15] and dCache [16] for distributed access to bulk file data, CERNVM-FS [17] for distributed and cached access to software, IgProf [18] for profiling very large C++ applications like those used in HEP, and the Parrot [19] virtual file system. In addition many software tools written by theorists are used heavily by the experiments, including event generators such as SHERPA [20] and ALPGEN [21] and jet finding tools like FastJet [22, 23]. This is not meant to be an exhaustive list. The Community White Paper provides a more complete software development roadmap, along with specific ideas for the sustainability challenges for each software package and relevance to the long-run HL-LHC and broader HEP programs.

A particularly interesting example of software development that is ripe for sustainable evolution is the 10M lines of code already developed within individual experiments, often as a by-product of the physics research program, and typically without with the explicit aim of producing sustainable software. While much of the code is experiment-specific due to actual differences in the detectors and appropriate techniques, some of this is simply redundant development of different implementations of the same functionalities. Common development of algorithms and software implementations for the HL-LHC era should allow better use of developer resources *and* provide more sustainable software. An example example is the need to develop parallel algorithms and implementations for the increasingly computationally intensive charged particle track reconstruction.

One of the major social challenges identified as part of the CWP and S2I2 process is how to effectively engage with the Computer Science, Data Science, Software Engineering, and other domain science communities. They have been very clear that they want to know what can HEP offer them to help advance their science. They are less interested in telling us what known paradigms and technologies we should use to solve *our* problems and more interested in working with us to identify and study intellectually interesting problems in *their* domains. As an example, while the multi-exabyte scale of the anticipated datasets will not be as large as some in the private sector, it *will* be large and we should be able work with them to provide access to the meta-data associated with its organization, management, and access. This will support computer science research, and what they learn in studying our patterns will inform our software designs for managing the datasets, creating a virtuous cycle. Identifying areas where HEP can work with other communities to everyone's mutual benefit will require carefully identifying appropriate projects and partners.

In developing a broad vision for HL-LHC computing, and understanding where marginal investments in software can enable the most science and the most transformative science, we need to engage more fully with outside experts. Areas where they may be able help us address include:

- developing "enterprise computing models" to balance investments in software engineering compared to hardware and to balance computing power versus storage requirements, also accounting for operating costs (personnel, wall power, cooling, real estate, etc.);
- determining the life-cycle cost of purchasing and maintaining hardware compared to buying resources from commercial cloud services;
- understanding the trade-offs between software optimized for individual experiments and software which is commonly developed and used, but optimized for no one.
- understanding how to develop and maintain sustainable software, including design, documentation, and training.
- understanding what tools already exist in other scientific disciplines or are being planned.

During the conceptualization process we began to build relationships with some relevant software experts, but we did not make similar progress building bridges to project management experts from outside HEP. Assuring ourselves that an Institute is investing well in developing and advancing software critical to the scientific goals of the HL-LHC community will be a continuing challenge. The governance and management structure of the Institute will need to address these issues directly.

Several scales define the computing needs of the experiments. One of the simplest metrics is the size of the datasets. At CERN's Resource Review Board meeting in April, 2015 the Computing Resources Scrutiny Group (CRSG) reported their findings. The 2014 tape and disk storage requirements for each of the experiments is summarized in Table 1. In the HL-LHC era, the number of events produced in the detectors will increase by a factor of 100. The experiments are all financially constrained – while all want to record data from (at least) 100 times as many collisions, it will be necessary to record less data per event to fit into the anticipated budget envelopes. In addition to the data stored on tape and disk, the experiments will need to process proportionately more data in their high level triggers, and they will need to generate and analyze proportionately more simulated data. The CSRG made several comments and recommendations. The first two are:

Table 1: Mass storage used by the LHC experiments in 2014, corresponding to the full Run 1 statistics. Numbers extracted from the CRSG report, CERN-RBB-2015-014 [24]. During the HL-LHC era, the experiments plan to record data from 100 times as many collisions, although it will be necessary to store less data per event.

Experiment	Disk Usage (PB)	Tape Usage (PB)	Total (PB)
ALICE	20	15	35
ATLAS	86	73	159
CMS	59	69	128
LHCb	15	16	31
Total	180	173	**353**

1. As reported previously, the experiments' requests for Run 2 are made keeping in mind a flat funding profile (not adjusted for inflation). However, we see a tendency as Run 2 progresses for the experiments' requests to outstrip the growth that can be accommodated by this profile.

2. The CRSG strongly supports software engineering development and recommends that sufficient effort be funded to support this. Improving the efficiency of software, including making optimal use of new hardware designs is essential to mitigate the growth in resource use. There have been substantial improvements made for Run 2. *In the longer term, with orders-of-magnitude increases in the expected computing needs for Run 3, this work is even more essential* (emphasis added).

Run 2 started in the summer of 2015, with the bulk of the luminosity being delivered in 2016 - 2018. The April 2016 CSRG report [?] estimated the ALICE, ATLAS, and CMS usages for the full period 2016 - 2018. A summary is shown in Table 2, along with corresponding numbers for LHCb taken from their 2017 estimate [?]. Altogerther, the LHC experiments will be saving more than an exabyte of data in mass storage by the end of Run 2. In their April 2017 report [?], the CSRG says that "growth equivalent to 20%/year $\cdots$ towards HL-LHC $\cdots$ should be assumed". While no one expects this projection to be accurate over 10 years, simple exponentiation projects a factor of 6 growth in data to be stored offline. Investing in sustainable and innovative software infrastructure will allow the experiments to fully process the multi-exabyte data sets produced by the upgraded accelerator and detectors, and enable them to reap the benefits and achieve the promised scientific reach of these instruments.

Table 2: Estimated mass storage to be used by the LHC experiments in 2018, at the end of Run 2 data-taking. Numbers extracted from the CRSG report, CERN-RBB-2016-049 [?] for ALICE, ATLAS, & CMS and taken from LHCb-PUB-2017-019 [?] for LHCb.

Experiment	Disk Usage (PB)	Tape Usage (PB)	Total (PB)
ALICE	98	86	184
ATLAS	164	324	488
CMS	141	144	285
LHCb	41	79	120
Total	444	633	**1077**

4 Summary of S2I2-HEP Conceptualization Process

The proposal "Conceptualization of an S2I2 Institute for High Energy Physics (S2I2-HEP)" was submitted to the NSF in August, 2015. Awards ACI-1558216, ACI-1558219, and ACI-1558233 were made in July, 2016 and the S^2I^2 conceptualization project began in fall, 2016. Two major deliverables were foreseen from the conceptualization process in the original S2I2-HEP proposal:

(1) A **Community White Paper (CWP)** describing a global vision for software and computing for the HL-LHC era; this includes discussions of elements that are common to the LHC community as a whole and those that are specific to the individual experiments. It also discusses the relationship of the common elements to the broader HEP and scientific computing communities. Many of the topics discussed here address issues required for a HEP S^2I^2 implementation proposal, but which are also generic to the larger community. The CWP document has been organized and written as an initiative of the HEP Software Foundation, in collaboration with our DOE Laboratory and foreign colleagues. In addition to providing material for this S2I2 Strategic Plan, the CWP provides a roadmap for members of the international HL-LHC community to approach their funding agencies.

(2) A separate **Strategic Plan** identifying areas where the U.S. university community can provide leadership and discussing those issues required for an S^2I^2 which are not (necessarily) relevant to the large community. This is the document you are currently reading. Topics in the Strategic Plan include:

- where does the U.S. university community already have expertise and important leadership roles;
- which software elements and frameworks would provide the best educational and training opportunities for students and postdoctoral fellows;
- what types of programs (short courses, short-term fellowships, long-term fellowships, etc.) might enhance the educational reach of an S^2I^2;
- possible organizational, personnel and management structures and operational processes;
- how the investment in an S^2I^2 can be judged and how the investment can be sustained to assure the scientific goals of the HL-LHC.

The Strategic Plan has been prepared in collaboration with members of the U.S. university community. Although it is not a project deliverable, an additional goal of the conceptualization exercise has been to build a team for submitting an S^2I^2 implementation proposal, should there be an appropriate solicitation.

The process to produce these two documents has been built around a series of dedicated workshops, meetings and special outreach sessions in preexisting workshops. Many of these were organized under the umbrella of the HSF and involved the full international community. A smaller, dedicated set of workshops focused on S2I2- or U.S.- specific topics, including interaction with the Computer Science community. S2I2-HEP project participant support funds were used to support the participation of relevant individuals in both types of workshop. A complete list of the workshops held as part of the CWP or to support the S2I2-specific efforts is included in Appendix B.

In order to anchor it to the larger LHC planning process, the CWP process carried out by the HSF was initiated by a charge [25] from the Worldwide LHC Computing Grid (WLCG) to the HSF and the LHC experiments. The HSF took the charge and, per its mission, broadened it to invite other relevant HEP experiments and labs to participate. The CWP charge asked for the identification and prioritization of the software research and development investments required:

1. to achieve improvements in software efficiency, scalability and performance and to make use of the advances in CPU, storage and network technologies

2. to enable new approaches to computing and software that could radically extend the physics reach of the detectors

3. to ensure the long term sustainability of the software through the lifetime of the HL-LHC

The community at large was engaged in the CWP and S2I2 processes by building on existing communication mechanisms. The involvement of the LHC experiments (including in particular the software and computing coordinators) in the CWP process allowed for communication using the pre-existing experiment channels. To reach more widely than the LHC experiments, specific contacts were made with individuals with software and computing responsibilities in the FNAL muon and neutrino experiments, Belle-II, the Linear Collider community as well as various national computing organizations. The HSF had in fact been building up mailing lists and contact people beyond LHC for about 2 years before the CWP process began, and the CWP process was able to build on that.

Early in the process a number of working groups were established on various topics which were expected to be important parts of the HL-LHC roadmap: *Careers, Staffing and Training; Computing Models, Facilities, and Distributed Computing; Conditions Database; Data Organization, Management and Access; Data Analysis and Interpretation; Data and Software Preservation; Detector Simulation; Event Processing Frameworks; Machine Learning; Physics Generators; Software Development, Deployment and Validation/Verification; Software Trigger and Event Reconstruction; and Visualization.*

In addition there was a small set of working groups envisioned at the beginning of the CWP process that failed to gather significant community interest or which were integrated into the active working groups listed above. These inactive working groups were: *Math Libraries; Data Acquisition Software; Various Aspects of Technical Evolution (Software Tools, Hardware, Networking); Monitoring; Security and Access Control; and Workflow and Resource Management.*

The CWP process began with a kick-off workshop at UCSD/SDSC in January, 2017 and concluded with a final workshop in June, 2017 in Annecy, France, with a large number of intermediate topical workshops and meetings. The entire CWP process involved an estimated 250 participants. The working groups continued working with the aim of producing a white paper per working group by the end of September, 2017, and the full Community White Paper document by the end of October, 2017.

At the CWP kick of workshop (in Jan. 2017 at UCSD/SDSC), each of the (active) working groups defined a charge for itself, as well as a plan for meetings, google group for communication, etc. The precise path for each working group in terms of teleconference meetings and actual in-person sessions or workshops varied from group to group. Each of the active working groups has produced a working group report, which is available from the HSF CWP webpage [26].

XXXX Additional description about the CWP endgame. (Once we have finished it!) XXXXX

The CWP process was intended to assemble the global roadmap for software and computing for the HL-LHC. In addition S2I2-specific activities were organized to explore which subset of the global roadmap would be appropriate for a U.S. university-based Software Institute and what role it would play together with other U.S. efforts (including both DOE efforts, the US-Atlas and US-CMS Ops programs and the Open Science Grid) and with international efforts. In addition the S2I2-HEP conceptualization project investigated how the U.S. HEP community could better collaborate with and leverage the intellectual capacity of the U.S. Computer Science and SI2 communities. Two dedicated S2I2 HEP/CS workshops were held as well as a dedicated S2I2 workshop co-located with the ACAT conference. In addition numerous outreach activities and discussions took place with the U.S. HEP community and specifically with PIs interested in software and computing R&D.

5 The HEP Community and Its Software

5.1 Software Development and Infrastructure in the HEP Community

The HEP community has by necessity developed significant experience in creating software infrastructure and processes which integrate software contributions from large, distributed communities of physics researchers. Each HEP experiment requires at a minimum application software for data acquisition, data handling, data processing, simulation and analysis as well as "infrastructure" software for application frameworks, data persistence, etc. Over the past 20 years HEP experiments have also became large enough to require resources larger than possible at the host laboratory (i.e. distributed computing), they began to develop also more sophisticated software for data management, data access and workload/workflow management.

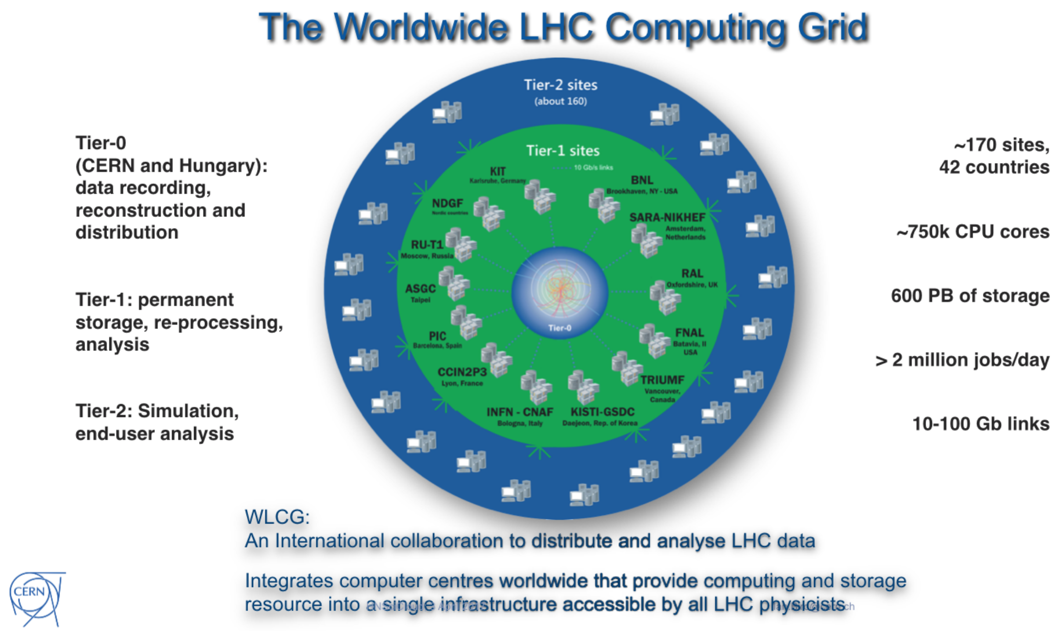

Figure 1: The Worldwide LHC Computing Grid (WLCG), which provides the computing resources needed by the four LHC experiments (ALICE, ATLAS, CMS, LHCb).

These software elements are used 24 hours a day, 7 days a week, over the entire year. For example, they are used by the LHC experiments in the ~ 170 computing centers participating in the Worldwide LHC Computing Grid (shown in Figure 1). (**Reference also OSG here?**) The computing use ranges from "production" activities which are organized centrally by the experiment (e.g. basic processing of RAW data) to "analysis" activities initiated by individuals or small groups of researchers for their specific research investigations.

To build this software each of the major HEP experiments provides a set of "software architectures and lifecycle processes, development, testing and deployment methodologies, validation and verification processes, end usability and interface considerations, and required infrastructure and technologies" (to quote the NSF S2I2 solicitation). Computing hardware to support the development process itself (such as continuous integration and test machines) is typically provided by the host laboratory for the experiments, e.g. CERN for the LHC experiments. Each experiment manages software release cycles for its software stack in order to meet goals ranging from physics needs to bug and performance fixes. On slightly longer time scales the software development efforts within the experiments must respond to various challenges including evolving physics goals and discoveries, general infrastructure and technology evolution as well as the evolution of the experiments themselves (detector upgrades, accelerator energy and luminosity increases, etc.). For the most part, the HEP community has not formally adopted any explicit development methodology or model, however the de-facto method adopted is very similar to agile software development [**?**, AGILE]

HEP experiments have also maintained these software infrastructures over time scales ranging from years to decades and in projects involving hundreds to thousands of developers. Figure 2 shows an example from the CMS experiment at the LHC. Over a ten year period, up to 300 people were involved in making changes to the software each month. The figure also shows an example of the evolution of the technical infrastructure, in which the version control system was changed from CVS to github.

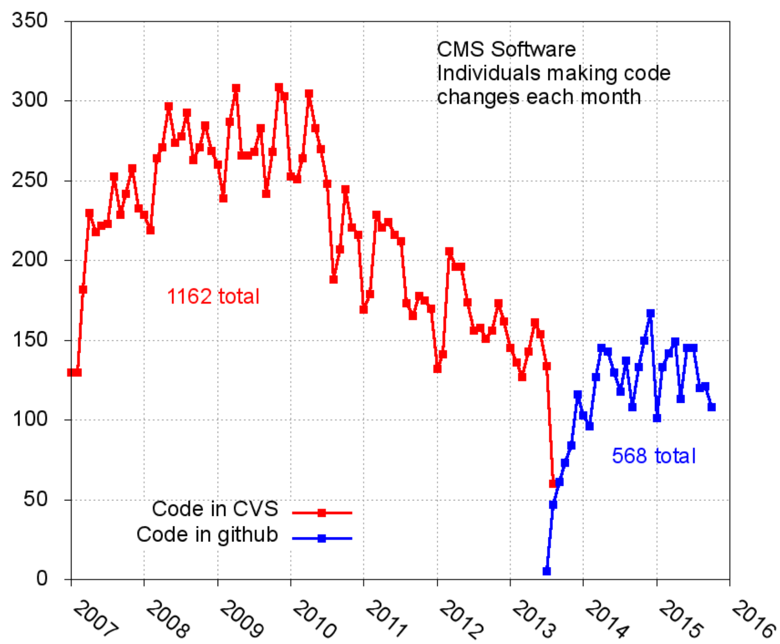

Figure 2: Evolution of the number of invididuals making contributions to the CMS software each month over the period from 2007 to 2016. Also shown is how the developer community was maintained through large changes to the technical infrastructure, in this case the evolution of the version control system from CVS to github. This plot shows only the application software managed in the experiment-wide software release (CMSSW) and not “infrastructure” software (e.g. for data and workflow management) or “analysis” software developed by individuals or small groups.

5.2 The HEP Software Ecosystem

In practice much of the actual software and infrastructure is implemented *independently* by each experiment. This includes managing the software development and deployment process and the resulting software stack. Some of this is a natural result of the intrinsic differences in the actual detectors (scientific instruments) used by one experiment or another. Independent software stacks are also the healthy result of different experiments and groups making different algorithmic and implementation choices. And last, but not least, each experiment must have control over its own schedule to insure that it can deliver physics results in a competitive environment. This implies sufficient control over the software development process and the software itself which the experiment uses.

The independence of the software processes in each experiment of course has some downsides. At times similar functionalities are implemented redundantly in multiple experiments. Issues of long term software sustainability can arise in these cases when the particular functionality is not actually mission-critical or specific to the experiment. Obtaining human resources (both in terms of effort and in terms of intellectual input) can be difficult if the result only impacts one particular HEP experiment. Trivial technical and/or communication issues can prevent even high quality tools developed in one experiment from being adopted by another.

The HEP community has nonetheless a developed an ecosystem of common software tools which are shared more widely in the community. Ideas and experience with software and computing in the HEP community are shared at dedicated HEP software/computing conferences such as CHEP and ACAT (**references!**).

Overall the software landscape itself is quite varied. The preparations for LHC have yielded important community software tools for data analysis like ROOT [8] and detector simulation GEANT4 [9, 10], both of which have been critical not only for LHC but in most other areas of HEP and beyond. Other tools have been shared between some, but not all, experiments. Examples include the GAUDI [11] event processing framework, RooFit [12] for data modeling and fitting and the TMVA [13] toolkit for multivariate data analysis. Tools developed in collaboration with computer scientists include FRONTIER [14] for cached access to databases, XROOTD [15] and dCache [16] for distributed access to bulk file data, CERNVM-FS [17] for distributed and cached access to software, IgProf [18] for profiling very large C++ applications like those used in HEP, and the Parrot [19] virtual file system. In addition many software tools written by theorists are used heavily by the experiments, including event generators such as SHERPA [20] and ALPGEN [21] and jet finding tools like FastJet [22, 23]. This is not meant to be an exhaustive list. The conceptualization process will need to produce a full software map, along with specific ideas of the sustainability challenges for each software package and relevance to the long-run HL-LHC and HEP programs.

More interesting is the 10M lines of code mentioned earlier that has been developed within individual experiments, often as a by-product of the physics research program and typically without with the explicit aim of producing sustainable software. While much of the code is experiment specific due to actual differences in the detectors and appropriate techniques, some of this is simply redundant development of different implementations of the same functionalities. Long term sustainability issues exist in many places in such code. One obvious example is the need to develop parallel algorithms and implementations for the increasingly computationally intensive charged particle track reconstruction. A challenge for the Institute will be to forge effective collaborative efforts between the experiments to solve the software issues.

The community also a long history with "common software" projects. Despite the best of intentions many of these projects are seen as having failed to deliver XXXX

6 The Institute Role

6.1 Institute Role within the HEP Community

The mission of a Scientific Software Innovation Institute (S2I2) for HL-LHC Physics should be to serve as both an active software research and development center and as an intellectual hub for the larger R&D effort required to ensure the success of the HL-LHC scientific program. The timeline for the LHC and HL-LHC is shown in Figure 3. A Software Institute operating roughly in the 5 year period from 2019 to 2023 (inclusive) will coincide with two important steps in the ramp up to the HL-LHC: the delivery of the Computing Technical Design Reports (CTDRs) of ATLAS and CMS in ∼2020 and LHC Run 3 in 2021-2023. The CTDRs will describe the experiments' technical blueprints for building software and computing to maximize the HL-LHC physics reach, given the financial constraints defined by the funding agencies. For ATLAS and CMS, the increased sizes of the Run 3 data sets relative to Run 2 will not be that great, and changes to the detectors will be modest (compared to the upgrades anticipated for Run 4). As a result, ATLAS and CMS will have an opportunity to deploy prototype elements of the HL-LHC computing model during Run 3 as real road tests, even if not at full scale. In contrast, LHCb is making its major transition in terms of how much data will be processed at the onset of Run 3. Some Institute deliverables could be deployed at full scale to directly maximize LHCb physics and provide valuable experience the larger experiments can use to prepare for the HL-LHC.

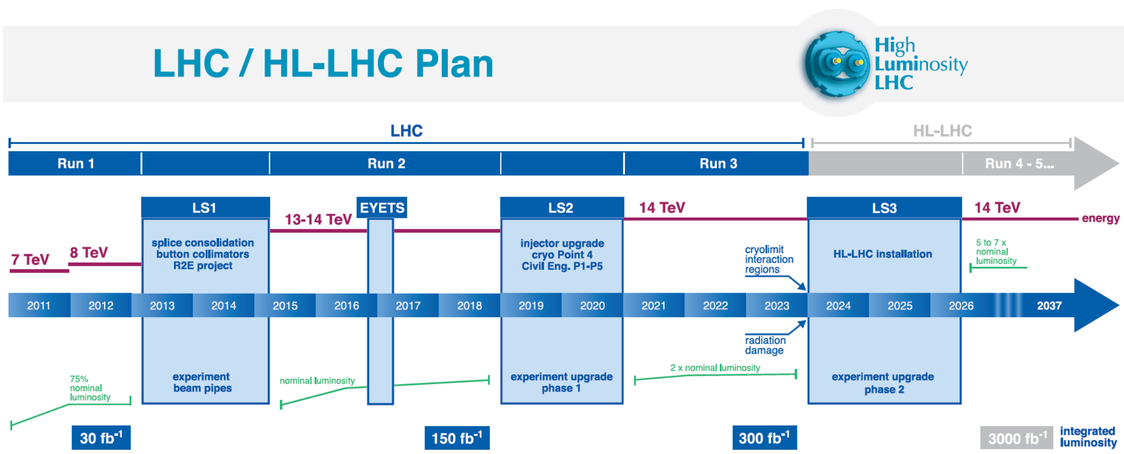

Figure 3: Timeline for the LHC and HL-LHC, indicating both data-taking periods and "shutdown" periods which are used for upgrades of the accelerator and detectors. Data-taking periods are indicated by green lines showing the relative luminosity and red lines showing the center of mass energy. Shutdowns with no data-taking are indicated by blue boxes (LS = Long Shutdown, EYETS = Extended Year End Technical Stop).

To accomplish its mission as an active R&D center and intellectual hub in the 2019-2023 time period, the specific goals of an S2I2 for HL-LHC Physics will be:

1. To build a new community process by which the software for the HL-LHC era can be developed, including both the collaborations and workforce required to execute the process.

2. To drive the software R&D process in specific focus areas and serve as an intellectual hub for the larger community effort

3. To demonstrate concrete results from the community process and institute collaborations in (*a*) the development of the CTDRs for ATLAS and CMS and (*b*) during LHC Run 3

A community *process* for software development which spans multiple experiments will be a somewhat new entity within the HEP community. Numerous projects exist, or have existed, which aim to deliver specific software packages to multiple experiments.

The direct resources funded by the institute will not be enough by itself to design, build and sustain the cyberinfrastructure required for the challenges of the HL-LHC. What the institute can do is serve as the primary engine for innovation in collaboration with the wider HEP and CS research communities. It should leverage and amplify the impact of effort from those communities.

The resulting process must be impactful, integrated, innovative and informed. (to expand)

6.2 Institute Role in the Software Lifecycle

The Institute will focus its resources on developing innovative ideas through the prototype stage and along the path to become software products used by the wider community. It will also provide technical support to the experiments and others to develop sustainability and support models for the software products developed. In its role as an intellectual hub for HEP software innovation, it provide advice and guidance broadly on software development within the HEP ecosystem. For example, a new idea or direction under consideration by an experiment could be critically evaluated by the Institute in terms of its essence, novelty, sustainability and impact which would then provide written recommendations for the proposed activity. This will be achieved through a critical mass of experts in scientific software development inside and outside of HEP and the computer science community who partner with the Institute.

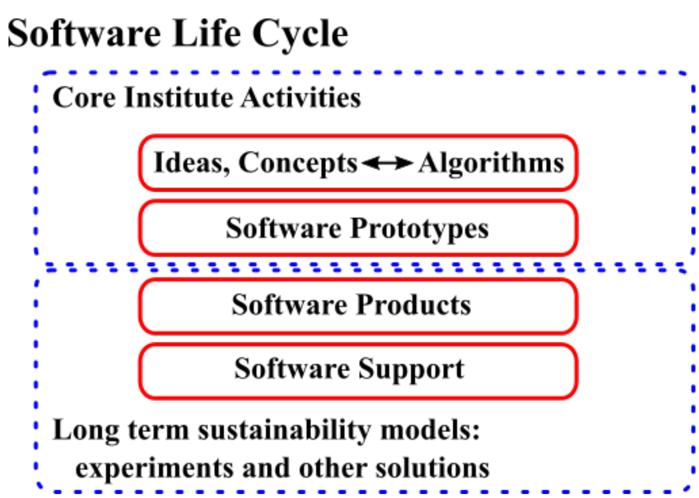

Figure 4: Roles of the Institute

- The institute can have different levels of ownership.
- Community software to be developed (and owned) by the institute needs a sustainability model

6.3 Institute Elements

The Institute will have a number of internal functional elements, as shown in Figure 5. (External interactions of the institute will be described in Section 9.)

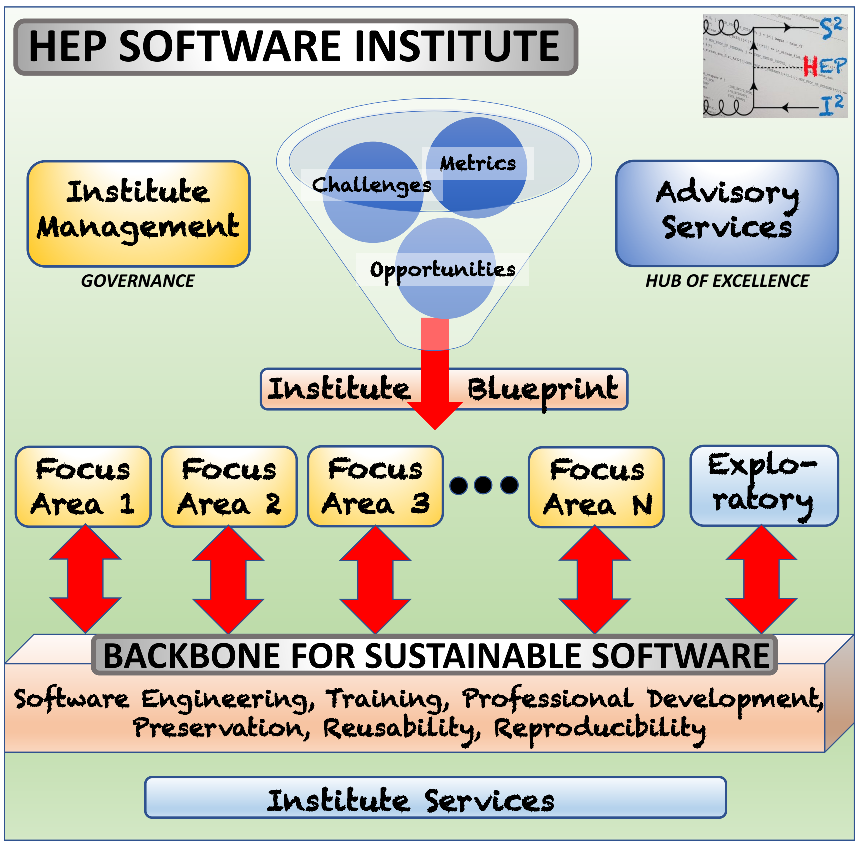

Figure 5: Internal elements of the Institute.

Institute Management: In order to accomplish its mission, the institute will have a well-defined internal management structure, as well as external governance and advisory structures. Further information on this aspect is provided in Section 8.

Focus Areas: The Institute will have N focus areas, which will pursue the main R&D goals being pursued by the Institute. High priority candidates for these focus areas are described in Section 7. How many of these will be implemented in an Institute implementation will depend on available funding. Each focus area will have its own specific plan of work and metrics for evaluation.

Institute Blueprint: The Institute Blueprint activity will maintain the software vision for the Institute and, several times per year, bring together expertise to answer specific key questions within the scope of the Institute vision or within the wider scope of HEP software/computing activities. This will be a key element to inform the evolution of the Institute and the wider community in the medium and long term.

Exploratory: From time to time the Institute may deploy modest resources for short term exploratory R&D projects of relevance to inform the planning and overall mission of the Institute.

Backbone for Sustainable Software: In addition to the specific technical advances which will be enabled by the Institute, a dedicated "backbone" activity will focus on how these activities are communicated to students and researchers, identifying best practices and possible incentives, developing and providing training and making data and tools available to the public. Further information on this activity is included in Section 7.6.

Advisory Services: The Institute will play a role in the larger research software community (in HEP and beyond) by being available to provide technical and planning advice to other projects and by participating in reviews. The Institute will execute this functionality both with individuals directly employed by the Institute and by involving others through its network of partnerships.

Institute Services: As required the Institute may provide other services in support of its software R&D activities. These may include: basic services such as access to build platforms and continuous integration systems; software stack build and packaging services; technology evaluation services; performance benchmarking services; access to computing resources and related services required for testing of prototypes at scale in the distributed computing environment. In many cases, the actual services will not be owned by the Institute itself, but instead to one its many partners. The role of the Institute in this case will be to organize the access in support of its mission.

7 Strategic Areas for Initial Investment

A university-based S2I2 focussed on software needed to ensure the scientific success of the HL-LHC will be part of a larger Research, Development, and Deployment community. The process leading to the Community White Paper, discussed in Section 4, identified three *impact criteria* for judging the value of additional investments:

- **Impact - Physics:** Will efforts in this area enable new approaches to computing and software that maximize, and could potentially radically extend, the physics reach of the detectors?
- **Impact - Resources:** Will efforts in this area achieve required improvements in software efficiency, scalability and performance and make use of the advances in CPU, storage and network technologies?
- **Impact - Sustainability:** Will efforts in this area guarantee the long term sustainability of the software through the lifetime of the HL-LHC?

These are key questions for projects funded by any mechanism, especially an S2I2. During the process, Working Groups (WGs) formed to consider potential activities in a variety of areas:

- Data Analysis and Interpretation
- Machine Learning
- Software Trigger and Event Reconstruction
- Data Access, Organization and Management
- Workflow and Resource Management
- Data and Software Preservation
- Careers, Staffing and Training
- Visualization
- Detector Simulation
- Various Aspects of Technical Evolution (Software Tools, Hardware, Networking)
- Data Acquisition Software
- Conditions Database
- Physics Generators
- Computing Models, Facilities and Distributed Computing
- Software Development, Deployment and Validation/Verification
- Event Processing Frameworks

In preparing the individual CWP "chapters", each WG was asked to evaluate their proposed R&D activities in terms of these criteria. In assembling the shorter CWP which summarizes the material produced by each WG, the editors identified high, medium, and lower impact areas for investment.

7.1 Rationale for choices and prioritization of a university-based S2I2

The S2I2 should focus it efforts on a subset of high impact areas for R&D. It will not have the resources to solve all the interesting software problems for the HL-LHC, and it cannot take responsibility for deploying and sustaining experiment-specific software. It needs to align its activities with the rest of the community. In addition to identifying areas in which it will lead efforts, the Institute should clearly indentify areas in which it will not. These will include some where it will have no significant role at all, and others where it might participate with lower priority.

The S2I2 process was largely community-driven. In preparing for the final workshop, held in conjunction with the ACAT workshop in August, 2017, *additional* S2I2-specific criteria were developed for identifying Focus Areas for the Institute and specific initial R&D topics within each:

- **Interest/Expertise:** Does the U.S. university community have a strong interest and expertise in the area?
- **Leadership:** Are the proposed focus areas complementary to efforts funded by the US-LHC Operations programs, the DOE, or international partners?
- **Value:** Is there potential to provide value to more than one LHC experiment and to the wider HEP community?
- **Research/Innovation:** Are there opportunities for combining research and innovation as part of partnerships between the HEP and Computer Science/Software Engineering/Data Science communities?

Opportunities for advanced training and education of students and post-docs were also considered. At the end of the workshop, there was a general consensus that high priority Focus Areas where an S2I2 can play a leading role include:

- Scalable Analysis Systems
  - plus Resource and Preservable Workflow Management for Analysis
  - plus Visualization for Data Analytics
- Machine Learning Applications
  - plus ML links to Simulation (fast sim, tuning, efficient use of)
  - plus Visualization for ML Analytics
- Data Organization, Management and Access (DOMA)
  - plus Interactions with Networking Resources
- Reconstruction Algorithms and Software Triggering
  - plus Anomaly Detection

Two more potential Focus Areas were identified as medium priority for an S2I2:

- Production Workflow, Workload and Resource management
- Event Visualization
  - primarily collaborative and immersive event displays

Production workflow as well as workload and resource management are absolutely critical software elements for the success of the HL-LHC. And they will require sustained investment to keep up with the increasing demands. However, the existing operations programs plus other DOE-funded projects are leading the efforts here. One topic in this area where an S2I2 might lead or collaborate extensively is workflows for compute-intensive analysis. Within the S2I2, this can be addressed as part of Scalable Analysis Systems. Similarly, visualization for data analytics can be addressed there and visualization for ML analytic can be addressed as part of ML Applications.

Although software R&D efforts in each of the following areas will be critical for the success of the HL-LHC, there was a general consensus that other entities are leading the efforts, and they should have low priority for S2I2 efforts and resources:

- Conditions Database
- Event Processing Frameworks
- Data Acquisition Software
- General Detector Simulation
- Physics Generators

- Network Technology

As is evident from our decision to include elements of production workflow and visualization into higher priority focus areas, the definitions of focus areas are intentionally fluid. In addition, some of the proposed activities intentionally cross nominal boundaries.

7.2 Data Analysis Systems

At the heart of experimental HEP is development of facilities (e.g. particle colliders, underground laboratories) and instrumentation (e.g. detectors) that provides sensitivity to new phenomena. It is through the analysis and intepretation of data from sophisticated detectors that enables HEP to understand the universe at its most fundamental level, including the constituents of matter and their interactions, and the nature of space and time itself. The span of questions that can be answered by a single collaboration range from a few flagship observations from a scientifically focused experiment to a very diverse and large set of questions for multi-purpose detector. In all cases, data is analyzed by groups of researchers of varying sizes, from individual researchers to very large groups of scientists.

7.2.1 Challenges and Opportunities

Over the past 20 years the HEP community has developed and gravitated around a single analysis ecosystem: ROOT. This software ecosystem currently both dominates HEP analysis and impacts the full event processing chain, providing foundation libraries, I/O services, etc. It is an advantage for the HEP community compared to other science disciplines. It provides an integrated and validated toolkit. This lowers the hurdle to start an analysis, enabling the community to talk a common analysis language, as well as making improvements and additions to the toolkit quickly available to the whole community allowing a large number of analyses to benefit. The open source analysis tools landscape used primarily in industry is however evolving very fast and surpasses the HEP efforts both in total investment in analysis software development and the size of communities that use these new tools.

The emergence and abundance of alternative and new analysis components and techniques coming from industry open source projects is a challenge for the HEP analysis software ecosystem. The community is very interested in using these new techniques and technologies and would like to use these together with established components of the ecosystem and also be able to interchange old components with new open source components. We propose in the first year to perform R&D on enabling new open source tools to be plugged in dynamically in the existing ecosystem and mechanisms to dynamically exchange parts of the ecosystem with new components. This could include investigating new ways of package management and distribution following open source approaches. For the 3 year time frame, we propose to research a comprehensive set of bridges and ferries between the HEP analysis ecosystem and the industry analysis tool landscape, where a bridge enables the ecosystem to use an open source analysis tool and a ferry allows to use data from the ecosystem in the tool and vice versa.

The maintenance and sustainability of the current analysis ecosystem is a challenge. The ecosystem supports a number of use cases and integrates and maintains a wide variety of components. Components have to be prioritized to fit into the available effort envelope, which is provided by a few institutions and less distributed across the community. Legacy and less used parts of the ecosystem are hard to retire and their continued support strain the available effort. In the first year, we propose R&D to evolve policies to minimize this effort by retiring less used components from the integration and validation efforts. We propose to enable individuals to continue to use retired components by taking over their maintenance and validation following the central efforts of the ecosystem, spending a little of their own effort. But not every component can just be retired if it is not used by most of the ecosystem users. Therefore for the 3 year time frame, we propose to evolve our policies how to replace components with new tools, maybe external, and solicit the community helps in bridging and integrating it. In general we need to streamline the adoption of new alternatives in the analysis community and the retirement of old components of the ecosystem.

7.2.2 Current Approaches

The baseline analysis model utilizes successive stages of data reduction, finally analyzing a compact dataset with quick real time iteration. Experiments use a series of processing steps to reduce large input datasets down to sizes suitable for laptop scale analysis. The line between managed production-like analysis processing and individual analysis, as well as the balance between harmonized vs. individualized analysis data formats differs by experiment, based on their needs and optimization level and the maturity of an experiment in its life cycle. The current baseline model stems from the goal to exploit the maximum possible scientific potential of the data while minimizing the time to insight for a large number of different analyses performed in parallel. It is a complicated product of diverse criteria ranging from computing resources and related innovation to management styles of the experiment collaborations. An evolution of the baseline approach is the ability to produce physics-ready data right from the output of the high-level trigger of the experiment, whereas the baseline approach also depends on further processing of the data with updated or new software algorithms or detector conditions. This could be a key enabler of a simplified analysis model that allows simple stripping of data and very efficient data reduction.

Methods for analyzing the data at the LHC experiments have been developed over the years and successfully applied to LHC data to produce physics results during Run-1 and Run-2. Analysis at the LHC experiments typically starts with users running code over centrally-managed data that is of O(100 kB/event) and contains all of information required to perform a typical analysis leading to publication. In this section, we describe some proposed models of analysis for the future building on the experience of the past.

The most common approach to analyzing data is through a campaign of data reduction and refinement, ultimately producing flat ntuples and histograms used to make plots and tables from which physics inference can be made. The centrally-managed data are O(100 kB/event) and are typically too large (e.g. O(100 TBs) for 35 fb-1of Run-2 data) to be brought locally to the user. An often stated aim of the data reduction steps is to arrive at a dataset that can fit on ones laptop, presumably to facilitate low-latency, high-rate access to a manageable amount of data during the final stages of analysis. At its core, creating and retaining intermediate datasets from data reduction campaign, bringing and keeping them close (e.g. on laptop/desktop) to the analyzers, is designed to minimize latencies and risks related to resource contention.

7.2.3 Research and Development Roadmap and Goals

The goal for future analysis models is to reduce the time to insight while exploiting the maximum possible scientific potential of the data within the constraints of computing and human resources. Analysis models aim towards giving scientists access to the data in the most interactive way possible, to enable quick turn-around in iteratively learning new insights from the data.

Many analyses have common deadlines defined by conference schedules. The increased analysis activity before these deadlines require the analysis system to be sufficiently elastic to guarantee a rich physics harvest. Also heterogeneous computing hardware like GPUs and new memory architectures will emerge and can be exploited to reduce the time to insight further.

Diversification of the Analysis Ecosystem. Over the past 20 years the HEP community has developed and rallied around an analysis ecosystem centered on ROOT. ROOT and its ecosystem both dominate HEP analysis and impact the full event processing chain, providing foundation libraries, I/O services etc. that have prevalence in the field. The analysis tools landscape is however evolving in ways that can have a durable impact on the analysis ecosystem and a strong influence on the analysis and core software landscape a decade from now. Data intensive analysis is growing in importance in other science domains as well as the wider world. Powerful tools from Data Science and new development initiatives, both within our field and in the wider open source community, have emerged. These tools include software and platforms for visualizing large volumes of complex data and machine learning applications, Automation of workflows and the use of automated pipelines are increasingly important and prevalent, often leveraging open source software such as continuous integration tools. Notebook interfaces have already demonstrated their value for tutorials and exercises in training sessions and facilitating reproducibility. Remote services like notebook-based analysis-as-a-service should be explored. We should leverage data formats which are stardard within data science, which is critical for gaining access to non-HEP tools, technologies and expertise from Computer Scientists. We should investigate optimizing some of the more promising formats for late-stage HEP analysis workflows.

Connecting to Modern Cyberinfrastructure. Facilitating easy access and efficient use of modern cyberinfrastruture for analysis workflows will be very important during the HL-LHC due to the anticipated proliferation of such platforms and an increased demaind for analysis resources to achieve the physics goals. These include scalable platorms, campus clusters, clouds, HPCs, which employ modern and evolving architectures such as GPUs, TPUs FPGAs memory-intensive systems and web servies. Develop mechanisms to instantiate resources for analysis from shared infrastruture as demand arises and share them elastically to support easy, efficient use. An approach gaining a lot of interest for deployment of anlaysis job payload is containers on Grid, Cloud, HPC and local resources. The goal is to develop approaches to data analysis which make it easy to utilize heterogeneous resources for analysis workflows. The challenges include making heterogeneous look not so to the analyzers and adapting to changes on resources (both technically and financially) not controlled by a given experiment.

Functional, Declarative Programming. Define what you want to do with data, tell systems. Dont tell systems how to do it. Allows systems to optimize data access patterns, execution concurrency. Further optimization could be gained by switching to a functional or declarative programming model. This would allow scientists to express the intended data transformation as a query on data. Instead of having to define and control the how, the analyst would declare the what of his analysis, essentially removing the need to define the event loop in an analysis and leave it to underlying services and systems to optimally iterate over events. Analogously to how programming in C++ abstracts implementation features compared to programming in assembler, it appears that these high-level approaches will allow to abstract from the underlying implementations, allowing the computing systems more freedom in optimizing the utilization of diverse forms of computing resources. We propose on the 3 year time frame to conclude on the already ongoing R&D projects (for example TDataFrame in ROOT) and to follow up with additional R&D projects to develop a prototype functional or declarative programming language model.

Improved Non-event data handling. An important area that has not received sufficient development is the access to non-event data for analysis (cross section values, scale factors, tagging efficiencies,). The community feels that like the existing capabilities for event data, namely easy storage of event data of all sorts of different content, a similar way of saving and accessing non-event information during the analysis step is needed. There exist many ways of doing this now, but no commonly accepted and supported way has yet emerged. This could be expanded to think about event vs. non-event data in general to support use cases from small data volumes (for example cross sections) to large data volumes (BDTs and NNs). We propose R&D in the area of non-event information handling on the 3-year time scale, which would facilitate analysis at much higher scales than today.

High-throughput, Low-latency Analysis Systems. [Add some intro]

- Spark-like analysis systems. A new model of data analysis, developed outside of HEP, maintains the concept of sequential ntuple reduction but mixes interactivity with batch processing. Spark is one such system, but TensorFlow, Dask, Pachyderm, and Thrill are others. Distributed processing is either launched as a part of user interaction at a command prompt or wrapped up for batch submission. The key differences from the above are:

1. parallelization is implicit through map/filter/reduce functionals
2. data are abstracted as remote, distributed datasets, rather than files
3. computation and storage are mixed for data locality: a specialized cluster must be prepared, but can yield higher throughput.

A Spark-like analysis facility would be a shared resource for exploratory data analysis (e.g. making quick plots on data subsets through the spark-shell) and batch submission with the same interface (e.g. substantial jobs through spark-submit). The primary advantage that software products like Spark introduce is in simplifying the users access to data, lowering the cognitive overhead to setting up and running parallel jobs. Certain types of jobs may also be faster than batch processing, especially flat ntuple processing (which benefits from SQL-like optimization) and iterative procedures such as fits and machine learning (which benefit from cluster-wide cache).

Although Spark itself is the leading contender for this type of analysis, as it has a well developed ecosystem with many third-party tools developed by industry, it is the style of analysis workflow that we are distinguishing here rather than the specific technology present today. Spark itself is hard to interface with C++, but this might be alleviated by projects such as ROOTs TDataFrame, which presents a Spark-like interface in ROOT, and may allow for more streamlined interoperability.

- Query-based analysis systems. In one vision for a query-based analysis approach, a series of analysis cycles, each of which provides minimal input (queries of data and code to execute), generates the essential output (histograms, ntuples, etc.) which can be retrieved by the user. The analysis workflow should be accomplished without focus on persistence of data traditionally associated with data reduction, however transient data may could be generated in order to efficiently accomplish this workflow and optionally could be retained to a facilitate an analysis checkpoint for subsequent execution. In this approach, the focus is on obtaining the analysis end-products in a way that does not necessitate a data reduction campaign and associated provisioning of resources.

  Advantages of a query-based analysis include:

  1. *Minimalist Analysis.* A critical consideration of the Sequential Ntuple Reduction method might reasonably question why analyzers would bother to generate and store intermediate data to get to same the outcomes of interest (histograms, etc). A more economical approach is to provide only the minimal information – code providing instructions for selecting the dataset, events of interest, and items to plot.
  2. *Democratization of Analysis.* In the Sequential Ntuple Reduction method, as one gets further down the data reduction chain, the user (or small group of users) needs to figure out how to provision and manage the storage required to accommodate this intermediate data which in many cases is accessed with small (¡10-4) or zero duty cycle. For small groups, the resources required (both in personnel and hardware) to execute such a data reduction campaign might be prohibitive in the HL-LHC era, effectively pricing them out of contributing strongly to analyses – possibly a lost opportunity for innovation and discovery. Removing the requirements on storing intermediate data in the analysis chain would help to democratize data analysis and streamline the overall analysis workflow.
  3. Ease of Provenance. The query-based analysis provides an opportunity for autonomous storage of provenance information, as all processing in an analysis step from primary

analysis-level data to the histograms is contained to a given facility. This information can be queried as well, for example.

Key elements of the required infrastructure for a future query-based analysis system are expected to include:

1. *Sharing resources with traditional systems.* Unlike a traditional batch system, access to this query system is intermittent, so it would be hard to justify allocating exclusive resources to it. Even with a large number of users to smooth out the minute-by-minute load, a query system would have strong a day-night effect, weekday-weekend effect, and pre-conference effect. Therefore, the query system must share resources with a traditional batch system (performing event reconstruction, making new AODs, for instance). Then the query system could elastically scale in response to load, preempting the batch system.
2. *Columnar Partitioning of Analysis Data.* Organizing data to enable fast-access of hierarchical event information (columnar data) is both a challenge and an opportunity. Presenting column partitions to analysis system as the fundamental unit of data management as opposed to files containing collections of events would bring several advantages for HEP end-user analysis (not reconstruction). These column partitions would become first-class citizens in the same sense that files are today: either as single-column files or more likely as binary blobs in an object store. We note that columns are already a first-class citizen in the ROOT file system, however, appropriate data management and analysis software that leverages this capability is missing. Given a data store full of columns, datasets become loose associations among these columns, with metadata identifying a set of columns as mutually consistent and meaningful for analysis.
3. *Fast Columnar Data Caching.* Columnar cache is a key feature of the query system, retaining input data between queries, which are usually repeated with small modifications (intentionally as part of a systematics study or unplanned as part of normal data exploration). RAM cache would be a logical choice, given the speed of RAM memory, but the query system cant hold onto a large block of RAM if it is to share resources with a batch system. Furthermore, it cant even allocate large blocks of RAM temporarily, since this would trigger virtual memory swapping to a disk that is slower than the network it is getting the source data from. The query system must therefore stay within a tight RAM budget at all times. The query systems cache would therefore need to be implemented in SSD (or some future fast storage, such as X-Point). We can assume the query system would have exclusive access to an attached SSD disk, since caching is not required for the batch process.
4. *P*rovenance. The query system should also attach enough provenance to each dataset that it could be recreated from the original source data, which is considered immutable. User datasets, while they can't be modified in-place, can be deleted, so a dataset's paper trail must extend all the way back to source data. This paper trail would take the form of the original dataset name followed by queries for each step of derivation: code and closure data.

7.2.4 Impact and Relevance for S2I2

Physics Impact: The very fast turnaround of analysis results that could be possible with new approaches to data access and organization would lead to rapid turnaround for new science.

Resources Impact: Optimized data access will lead to more efficient use of resources, thus holding down the overall costs of computing.

Sustainability Impact: This effort would improve the reproducibility and provenance tracking for workflows (especially analysis workflows), making physics analyses more sustainable through the lifetime of the HL-LHC.

Interest/Expertise: University groups have already pioneered significant changes to the data access model for the LHC through the development of federated storage systems, and are prepared to take this further. Other groups are currently exploring the features of modern storage systems and their possible implementation in experiments.

Leadership:

Value: All LHC experiments will benefit from new methods of data access and organization, although the implementations may vary due to the different data formats and computing models of each experiment.

Research/Innovation: This effort would rely on partnerships with data storage and access experts in the CS community, some of whom are already providing consultation in this area.

7.3 Reconstruction and Trigger Algorithms

The reconstruction of raw detector data and simulated data and its processing in real time represent a major component of today's computing requirements in HEP. A recent projection [27] of the ATLAS 2016 computing model results in >85% of the HL-LHC CPU resources being spent on the reconstruction of data or simulated events. We have evaluated the most important components of next generation algorithms, data structures, and code development and management paradigms needed to cope with highly complex environments expected in HEP detector operations in the next decade. New approaches to data processing were also considered, including the use of novel, or at least, novel to HEP, algorithms, and the movement of data analysis into real-time environments.

Several types of software algorithms are essential to the interpretation of raw detector data into analysis-level objects. Specifically, these algorithms can be categorized as:

1. Online: Algorithms, or sequences of algorithms, executed on events read out from the detector in near-real-time as part of the software trigger, typically on a computing facility located close to the detector itself.

2. Offline: As distinguished from online, any algorithm or sequence of algorithms executed on the subset of events preselected by the trigger system, or generated by a Monte Carlo simulation application, typically in a distributed computing system.

3. Reconstruction : The transformation of raw detector information into higher level objects used in physics analysis. A defining characteristic of reconstruction which separates it from analysis is that the quality criteria used in the reconstruction to, for example, minimize the number of fake tracks, are independent of how those tracks will be used later on. Reconstruction algorithms are also typically run as part of the processing carried out by centralized computing facilities.

4. Trigger: the online classification of events which reduces either the number of events which are kept for further offline analysis, the size of such events, or both. In this working group we were only concerned with software triggers, whose defining characteristic is that they process data without a fixed latency. Software triggers are part of the real-time processing path and must make decisions quickly enough to keep up with the incoming data, possibly using substantial disk buffers.

5. Real-time analysis: Data processing that goes beyond object reconstruction, and is performed online within the trigger system. The typical goal of real-time analysis is to combine the products of the reconstruction algorithms (tracks, clusters, jets...) into complex objects (hadrons, gauge bosons, new physics candidates...) which can then be used directly in analysis without an intermediate reconstruction step.

7.3.1 Challenges

Software trigger and event reconstruction techniques in HEP face a number of new challenges in the next decade. These are broadly categorized into 1) those from new and upgraded accelerator facilities, 2) from detector upgrades and new detector technologies, 3) increases in anticipated event rates to be processed by algorithms (both online and offline), and 4) from evolutions in software development practices.

Advancements in facilities and future experiments bring a dramatic increase in physics reach, as well as increased event complexity and rates. At the HL-LHC, the central challenge for object reconstruction is thus to maintain excellent efficiency and resolution in the face of high pileup values, especially at low object pT. Detector upgrades such as increases in channel density, high precision timing and improved detector geometric layouts are essential to overcome these problems. For software, particularly for triggering and event reconstruction algorithms, there is a critical need not to dramatically increase the processing time per event.

A number of new detector concepts are proposed on the 5-10 year timescale in order to help in overcoming the challenges identified above. In many cases, these new technologies bring novel requirements to software trigger and event reconstruction algorithms or require new algorithms to be developed. Ones of particular importance at the HL-LHC include high-granularity calorimetry, precision timing detectors, and hardware triggers based on tracking information which may seed later software trigger and reconstruction algorithms.

Trigger systems for next-generation experiments are evolving to be more capable, both in their ability to select a wider range of events of interest for the physics program of their experiment, and their ability to stream a larger rate of events for further processing. ATLAS and CMS both target systems where the output of the hardware trigger system is increased by 10x over the current capability, up to 1 MHz [28, 29]. In other cases, such as LHCb [30] and ALICE [31], the full collision rate (between 30 to 40 MHz for typical LHC operations) will be streamed to real-time or quasi-realtime software trigger systems. The increase in event complexity also brings a problem of overabundance of signal to the experiments, and specifically the software trigger algorithms. The evolution towards a genuine real-time analysis of data has been driven by the need to analyze more signal than can be written out for traditional processing, and technological developments which make it possible to do this without reducing the analysis sensitivity or introducing biases.

Evolutions in computing technologies are both opportunities to move beyond commodity x86 technologies, which HEP has used very effectively over the past 20 years, and significant challenges to derive sufficient event processing throughput per cost to reasonably enable our physics programs [32]. Specific items identified included 1) the increase of SIMD capabilities (processors capable of running a single instruction set simultaneously over multiple data), 2) the evolution towards multi- or many-core architectures, 3) the slow increase in memory bandwidth relative to CPU capabilities, 4) the rise of heterogeneous hardware, and 5) the possible evolution in facilities available to HEP production systems.

The move towards open source software development and continuous integration systems brings opportunities to assist developers of software trigger and event reconstruction algorithms. Continuous integration systems have already allowed automated code quality and performance checks, both for algorithm developers and code integration teams. Scaling these up to allow for sufficiently high statistics checks is among the still outstanding challenges. As the timescale for experimental data taking and analysis increases, the issues of legacy code support increase. Code quality demands increase as traditional offline analysis components migrate into trigger systems, or more generically into algorithms that can only be run once.

7.3.2 Current Approaches

Substantial computing facilities are in use for both online and offline event processing across all experiments surveyed. Online facilities are dedicated to the operation of the software trigger, while offline facilities are shared for operational needs including event reconstruction, simulation (often the dominant component) and analysis. CPU in use by experiments is typically at the scale of tens or hundreds of thousands of x86 processing cores. Projections to future needs, such as for the HL-LHC, show the need for a substantial increase in scale of facilities without significant changes in approach or algorithms.

The CPU needed for event reconstruction tends to be dominated by charged particle reconstruction (tracking), especially as the need for efficiently reconstructing low pT particles is considered. Calorimetric reconstruction, particle flow reconstruction, particle identification algorithms also make up significant parts of the CPU budget in some experiments.

Disk storage is typically 10s to 100s of PB per experiment. It is dominantly used to make the output of the event reconstruction, both for real data and simulation, available for analysis.

Current generation experiments have moved towards smaller, but still flexible, data tiers for analysis. These tiers are typically based on the ROOT [33] file format and constructed to facilitate both skimming of interesting events and the selection of interesting pieces of events by individual analysis groups or through centralized analysis processing systems. Initial implementations of real-time analysis systems are in use within several experiments. These approaches remove the detector data that typically makes up the raw data tier kept for offline reconstruction, and to keep only final analysis objects [34–36].

Detector calibration and alignment requirements were surveyed. Generally a high level of automation is in place across experiments, both for very frequently updated measurements and more rarely updated measurements. Often automated procedures are integrated as part of the data taking and data reconstruction processing chain. Some longer term measurements, requiring significant data samples to be analyzed together remain as critical pieces of calibration and alignment work. These techniques are often most critical for a subset of precision measurements rather than for the entire physics program of an experiment.

7.3.3 Research and Development Roadmap and Goals

The CWP identified seven broad areas which will be critical for software trigger and event reconstruction work over the next decade. These are:

Roadmap area 1: Enhanced vectorization programming techniques - HEP developed toolkits and algorithms typically make poor use of vector units on commodity computing systems. Improving this will bring speedups to applications running on both current computing systems and most future architectures. The goal for work in this area is to evolve current toolkit and algorithm implementations, and best programming techniques to better use SIMD capabilities of current and future computing architectures.

Roadmap area 2: Algorithms and data structures to efficiently exploit many-core architectures - Computing platforms are generally evolving towards having more cores in order to increase processing capability. This evolution has resulted in multi-threaded frameworks in use, or in development, across HEP. Algorithm developers can improve throughput by being thread safe and enabling the use of fine-grained parallelism. The goal is to evolve current event models, toolkits and algorithm implementations, and best programming techniques to improve the throughput of multithreaded software trigger and event reconstruction applications.

Roadmap area 3: Algorithms and data structures for non-x86 computing architectures (e.g. GPUs, FPGAs) - Computing architectures using technologies beyond CPUs offer an interesting alternative for increasing throughput of the most time consuming trigger or reconstruction algorithms. Such architectures (e.g. GPUs, FPGAs) could be easily integrated into dedicated trigger or specialized reconstruction processing facilities (e.g. online computing farms). The goal is to demonstrate how the throughput of toolkits or algorithms can be improved through the use of new computing architectures in a production environment.

Roadmap area 4: Enhanced QA/QC for reconstruction techniques - HEP experiments have extensive continuous integration systems, including varying code regression checks that have enhanced the quality assurance (QA) and quality control (QC) procedures for software development in recent years. These are typically maintained by individual experiments and have not yet reached the scale where statistical regression, technical, and physics performance checks can be performed for each proposed software change. The goal is to enable the development, automation, and deployment of extended QA and QC tools and facilities for software trigger and event reconstruction algorithms.

Roadmap area 5: Real-time analysis - Real-time analysis techniques are being adopted to enable a wider range of physics signals to be saved by the trigger for final analysis. As rates increase, these techniques can become more important and widespread by enabling only the parts of an event associated with the signal candidates to be saved, reducing the required disk space. The goal is to evaluate and demonstrate the tools needed to facilitate real-time analysis techniques. Research topics include compression and custom data formats; toolkits for real-time detector calibration and validation which will enable full offline analysis chains to be ported into real-time; and frameworks which will enable non-expert offline analysts to design and deploy real-time analyses without compromising data taking quality.

Roadmap area 6: Precision physics-object reconstruction, identification and measurement techniques - The central challenge for object reconstruction at HL-LHC is thus to maintain excellent efficiency and resolution in the face of high pileup values, especially at low object pT. Both trigger and reconstruction approaches need to exploit new techniques and higher granularity detectors to maintain or even improve physics measurements in the future. It is also becoming increasingly clear that reconstruction in very high pileup environments, such as the HL-LHC or FCC hh, will not be possible without adding some timing information to our detectors, in order to exploit the finite time during which the beams cross and the interactions are produced. The goal is to develop and demonstrate efficient techniques for physics object reconstruction and identification in complex environments.

Roadmap area 7: Fast software trigger and reconstruction algorithms for high-density environments - Future experimental facilities will bring a large increase in event complexity. The scaling of current-generation algorithms with this complexity must be improved to avoid a large increase in resource needs. In addition, it may be desirable or indeed necessary to deploy new algorithms, including advanced machine learning techniques developed in other fields, in order to solve these problems. The goal is to evolve or rewrite existing toolkits and algorithms focused on their physics and technical performance at high event complexity (e.g. high pileup at HL-LHC). Most important targets are those which limit expected throughput performance at future facilities (e.g. charged-particle tracking). A number of such efforts are already in progress across the community.

Add "Anomaly Detection", per ACAT discussion

US-CMS suggestions for "projects" (overlapping with above):

- Addressing the pileup/multiplicity induced exponential scaling issue of conventional HEP reconstruction algorithms
- Vectorization and advanced architectures (KNL, GPU, FPGA, but also future versions of XEON with ever increasing width of the vector units) of existing and new algorithms.
- How do we guarantee that the experiments makes as much use as possible of all the silicon we buy ?
- Example problem: CMS has a fully functional reconstruction for KNL that is x5 slower than on XEON, despite the fact that KNL has x10 more flops. So there is a relative factor of 50 or so in effectiveness of use of the silicon we buy for KNL vs XEON. If Intel were to merge features of the KNL into future XEON chips, we are likely getting worse use of the silicon we buy unless we do some serious R&D on our algorithms, and their implementations.
- Class of reconstruction algorithms that are re-written and optimized for vectorized architectures

7.3.4 Impact and Relevance for S2I2

Reconstruction algorithms are projected to be the biggest CPU consumer at HL-LHC. Code modernization or new approaches are needed given large increases in pileup (4x) and trigger output rate

(5-10x) and drive the estimates of resource needs the HL-LHC beyond what would be achievable with a flat budget. Trigger/Reco algorithm enhancements (and new approaches) enable extended physics reach even in more challenging detection environments (e.g., pileup). Moreover, Trigger/Reco algorithm development is needed to take full advantage of enhanced detector capabilities (e.g., timing detectors, high-granularity calorimeters). Real time analysis ideas hope to effectively increase achievable trigger rates (for fixed budget) through making reduced size, analysis-ready output from online trigger(-less) system.

Physics Impact: Pileup mitigation will be the fundamental technical issue of HL-LHC physics, and improvements to the reconstruction algorithms designed for modern architectures will be important for realizing the physics potential of the detectors.

Resources Impact: There are significant computing resources at HPC centers that could be made available to HL-LHC experiments at little cost, but many optimizations of existing code will be required to fully take advantage of them.

Sustainability Impact: University groups are already making progress in the use of chipsets such as GPUs for specific HEP applications, such as track pattern recognition and fitting. New detector elements that are expected for HL-LHC upgrade could especially benefit from pattern recognition on new architectures, and groups that are building these detectors will likely get involved.

Interest/Expertise: University groups are already making progress in the use of chipsets such as GPUs for specific HEP applications, such as track pattern recognition and fitting. New detector elements that are expected for HL-LHC upgrade could especially benefit from pattern recognition on new architectures, and groups that are building these detectors will likely get involved.

Leadership: It is likely that there will be some overlap with work done at DOE HPC centers, but NSF HPC centers might require independent efforts. (???)

Value: All LHC experiments will benefit from these techniques, although many implementations will likely be experiment-specific given differing detector configurations.

Research/Innovation: Much assistance will be required from the computing and software engineering communities to help prepare algorithms for new architectures.

7.4 Applications of Machine Learning

Machine Learning (ML) is a rapidly evolving approach to characterizing and describing data with the potential to radically change how data is reduced and analyzed. Some applications will qualitatively improve the physics reach of data sets. Others will allow much more efficient use of processing and storage resources, effectively extending the physics reach of the HL-LHC experiments. Many of the activities in this focus area will explicitly overlap with those in the other focus areas. Some will be more generic. As a first approximation, the HEP community will build domain specific applications on top of existing toolkits and ML algorithms developed by computer scientists, data scientists, and scientific software developers from outside the HEP world. We will also work with them to understand where some of our problems do not map onto existing paradigms well, and how these problems can be re-cast into abstract formulations of more general interest.

7.4.1 Opportunities

The world of data science has developed a variety of very powerful ML approaches for classification (using pre-defined categories), clustering (where categories are discovered), regression (to produce continuous outputs), density estimation, dimensionality reduction, etc. Some have been used productively in HEP for more than 20 years; others have been introduced relatively recently. More are on their way. A key feature of these algorithms is that most have open software implementations that are reasonably well documented. HEP has been using ML algorithms to improve software performance in many types of software for more than 20 years, and ML has already become ubiquitous in some types of applications. For example, particle identification algorithms that require combining information from multiple detectors to provide a single figure of merit use a variety of BDTs and neural nets. With the advent of more powerful hardware and more performant ML algorithms, we want to use these tools to develop application software that will:

- replace the most computationally expensive parts of pattern recognition algorithms and algorithms that extract parameters characterizing reconstructed objects;
- compress data significantly with negligible loss of fidelity in terms of physics utility;
- extend the physics reach of experiments by qualitatively changing the types of analyses that can be done.

The abundance of ML algorithms and implementations presents both opportunities and challenges for HEP. Which are most appropriate for our use? What are the trade-offs of one compared to another? What are the trade-offs of using ML algorithms compared to using more traditional software? These issues are not necessarily factorizable, and a key goal of an Institute will be making sure that the lessons learned by one any research team are usefully disseminated to the greater HEP world. In general, the Institute will serve as a repository of expertise. Beyond the R&D projects it sponsors directly, the Institute will help teams develop and deploy experiment-specifc ML-based algorithms in their software stacks. It will provide training to those developing new ML-based algorithms as well as those planning to use established ML tools.

7.4.2 Current Approaches

The use of ML in HEP analyses has become commonplace over the past two decades. Many analyses use the HEP-specific software package TMVA [13] included in the CERN ROOT [8] project. Recently, many HEP analysts have begun migrating to non-HEP ML packages like SciKit-Learn [37] and Keras [38]. Data scientists at Yandex created a Python package that provides a consistent API to most ML packages used in HEP [39], and another that provides some HEP-specific ML algorithms [40]. Packages like Spearmint [41] perform Bayesian optimization and can can improve HEP Monte Carlo [42, 43]. The keys to successfully using ML for any problem are:

- creating/indentifying the optimal training, validation, and testing data samples;
- designing and selecting feature sets; and
- defining appropriate problem-specific loss functions.

While each experiment is likely to have different specific use cases, we expect that many of these will be sufficiently similar to each other that research and development can be done commonly. We also expect that experience with one type of problem will provide insights into how to approach other types of problems.

7.4.3 Research and Development Roadmap and Goals

The following specific examples illustrate possible first-year activities.

- Charged track and vertex reconstruction is one of the most CPU intensive elements of the software stack. The algorithms are typically iterative, alternating between selecting hits associated with tracks and characterizing the trajectory of a track (a collection of hits). Similarly, vertices are built from collections of tracks, and then characterized quantitatively. ML algorithms have been used extensively outside HEP to recognize, classify, and quantitatively describe objects. We will investigate how to replace the most computationally expensive parts of the pattern recognition algorithms and the fitting algorithms that extract parameters characterizing the reconstructed objects. As existing algorithms already produce high-quality physics, the primary goal of this activity will be developing replacement algorithms that execute much more quickly while maintaining sufficient fidelity.
- ML algorithms can often discover patterns and correlations more powerfully than human analysts alone. This allows qualitatively better analysis of recorded data sets. For example, ML algorithms can be used to characterize the substructure of "jets" observed in terms of underlying physics processes. ATLAS, CMS, and LHCb already use ML algorithms to separate jets into those associated with b-quark, c-quarks, or lighter quarks. ATLAS and CMS have begun to investigate whether sub-jets can be reliably associated with quarks or gluons. If this can be done with both good efficiency and accurate understanding of efficiency, the physics reach of the experiments will be radically extended.
- The ATLAS, CMS, and LHCb detectors all produce much more data than can be moved to permanent storage. The process of reducing the size of the data sets is referred to as the trigger. Electronics sparsify the data stream using zero suppression and they do some basic data compression. While this will reduce the data rate by a factor of 100 (or more, depending on the experiment) to about 1 terabyte per second, another factor of order 1 500 is required before the data can be written to tape (or other long-term storage). ML algorithms have already been used very successfully to rapidly characterize which events should be selected for additional consideration and eventually persisted to long-term storage. The challenge will increase both quantitatively and qualitatively as the number of proton-proton collions per bunch crossing increases.
- All HEP experiments rely on simulated data sets to accurately compare observed detector response data with expectations based on the hypotheses of the Standard Model or models of new physics. While the processes of subatomic particle interactions with matter are known with very good precision, computing detector response analytically is intractable. Instead, Monte Carlo simulation tools, such as GEANT [ref], have been developed to simulate the propagation of particles in detectors. They accurately model trajectories of charged particles in magnetic fields, interactions and decays of particles as they traverse the fiducial volume, etc. Unfortunately, simulating the detector response of a single LHC proton-proton collision takes on the order of several minutes. *Fast simulation* replaces the slowest components of

the simulation chain with computationally efficient approximations. Often, this is done using simplified parameterizations or look-up tables which don't reproduce detector response with the required level of precision. A variety of ML tools, such as Generative Adversarial Networks and Variational Auto-encoders, promise better fidelity and comparable executions speeds (after training). For some of the experiments (ATLAS and LHCb), the CPU time necessary to generate simulated data will surpass the CPU time necessary to reconstruct the real data. The primary goal of this activity will be developing fast simulation algorithms that execute much more quickly than full simulation while maintaining sufficient fidelity.

7.4.4 Impact and Relevance for S2I2

Physics Impact: Software built on top of machine learning will provide the greatest gains in physics reach by providing new types of reconstructed object classification and by allowing triggers to more quickly and efficienctly select events to be persisted.

Resources Impact: Replacing the most computationally expensive parts of reconstruction will allow the experiments to use computing resources more efficiently. Optimizing data compression will allow the experiments to use data storage and networking resources more efficiently.

Sustainability Impact: Building our domain-specific software on top of ML tools from the larger scientific software community should reduce the need to maintain equivalent tools we built (or build) ourselves, but it will require that we help maintain the toolkits we use.

Interest/Expertise: U.S. university personnel are already leading significant efforts in using ML, from reconstruction and trigger software to tagging jet flavors to identifying jet substructures.

Leadership: There is a natural area for Intitute leadership: in addition to the existing interest and expertise in the university HEP community, this is an area where engaging academics from other disciplines will be a critical element in making the greatest possible progress.

Value: All LHC experiments will benefit from using ML to write more performant software. Although specific software implementations of algorithms will differ, much of the R&D program can be common. Sharing insights and software elements will also be valuable.

Research/Innovation: ML is evolving very rapidly, so there are many opportunities for basic and applied research as well as innovation. As most of the work developing ML algorithms and implementing them in software (as distinct from the applications software built using them) is done by experts in the computer science and data science communities, HEP needs to learn how to effectively use toolkits provided by the open scienctific software community. At the same time, some of the HL-LHC problems may be of special interest to these other communities, either because the sizes of our data sets are large (multi-exabyte) or because they have unique features.

7.5 Data Organization, Management and Access (DOMA)

Experimental HEP has long been a data intensive science and this will continue through the HL-LHC era. The success of HEP experiments is built on their ability to reduce the tremendous amounts of data produced by HEP detectors to physics measurements.

The reach of these data-intensive experiments is limited by how fast data can be accessed and digested by the computational resources; both technology and large increases in data volume require new computational models [7]. HL-LHC and the HEP experiments of the 2020s are no exception.

Extending the current data handling methods and methodologies is expected to be intractable in the HL-LHC era. The development and adoption of new data analysis paradigms gives the field, as a whole, a window in which to adapt our data access and data management schemes to ones which are more suited and optimally matched to a wide range of advanced computing models and analysis applications. This type of shift has the potential for enabling new analysis methods and allowing for an increase in scientific output.

7.5.1 Challenges and Opportunities

The LHC experiments currently provision and manage about an Exabyte of storage, approximately half of which is archival, and half is traditional disk storage. The storage requirements per year are expected to jump by a factor of 10 for the HL-LHC. This itself is faster than projected Moore's Law gains and will present major challenges. Storage will remain one of the visible cost drivers for HEP computing, however the projected growth and cost of the computational resources needed to analyze the data is also expected to grow even faster than the base storage costs. The combination of storage and analysis computing costs may restrict scientific output and potential physics reach of the experiments, thus new techniques and algorithms are likely to be required.

These three main challenges for data in the HL-LHC era can thus be summarized:

1. **Big Data:** the HL-LHC era will significantly increase both the date rate and the data volume. The computing systems will need to handle this without significant cost increases and within evolving storage technology limitations.

2. **Dynamic Distributed Computing:** In addition, the significantly increased computational requirements for the HL-LHC era will also place new requirements on data. Specifically the use of new types of compute resources (cloud, HPC) with different dynamic availability and characteristics are used will require more dynamic DOMA systems.

3. **New Applications:** New applications such as machine learning training or high rate data query systems for analysis will likely be employed to meet the computational constraints and to extend the physics reach of the HL-LHC. These new applications will place new requirements on how and where data is accessed and produced. For example, specific applications (e.g. training for machine learning) may require use of specialized processor resources such as GPUs, placing further requirements on data.

The rapid increase in recent years of data-intensive problems in both the commercial world and in the rest of the research world also provides a number of opportunities. These include:

- one
- two
- three

There must be R&D efforts in data management on how to minimize the impact of the data access and storage model on the overall cost of doing scientific analysis. This R&D should include an optimization of both the capital costs of storage, as well as the potential impacts the storage systems can have on the CPU requirements for the experiments and their costs. (Bah, silly statement from CWP WG doc.)

The projected event complexity of data from future LHC runs and from high resolution liquid argon detectors will require advanced reconstruction algorithms and analysis tools to understand. The precursors of these tools, in the form of new machine learning paradigms and pattern recognition algorithms, already are proving to be drivers for the CPU needs of the HEP community. As these techniques continue to grow and blossom, they will place new requirements on the computational resources that need to be leveraged by all of HEP. The storage systems that are developed, and the data management techniques that are employed will need to directly support this wide range of computational facilities, and will need to be matched to the changes in the computational work, so as not to impede the improvements that they are bringing.

As with CPU, the landscape of storage protocols accessible to us is trending towards heterogeneity. Thus, the ability to leverage new storage technologies as they become available into existing data delivery models becomes a challenge that we must be prepared for. In part, this also means HEP experiments should be prepared to leverage tactical storage. Storage that becomes most cost-effective as it becomes available (e.g., from a cloud provider) and have a data management and provisioning system that can exploit such resources on short notice. As discussed in the preceding sections, much of this change can be aided by active R&D into our own IO patterns; an approach which has not yet been adopted widely by the field.

On the hardware side, R&D is needed in alternative approaches to data archiving to determine the possible cost/performance tradeoffs. Currently, tape is extensively used to hold data that cannot be economically made available online. While the data is still accessible, it comes with a high latency penalty; limiting possible analysis. We suggest investigating either separate direct access-based archives (e.g. disk or optical) or new models that overlay online direct access volumes with archive space. This is especially relevant when access latency is proportional to storage density. Either approach would need to also evaluate reliability risks and the effort needed to provide data stability.

In the end, the results have to be weighed against the storage deployment models that, currently, differ between the various experiments. This makes evaluation of the effectiveness of a particular solution relatively complex. Unless experiments converge on a particular deployment model, we dont see how one can maximize the benefits of any particular storage ecosystem. The current patchwork of funding models may make that impractical to achieve but we do want to emphasize that unless convergence happens it is unlikely that the most cost-effective approach can be implemented..While our focus is convergence within the LHC community we do not want to imply that efforts to broaden that convergence to include non-LHC experiments should not be pursued. Indeed, as the applicable community increases, costs are typically driven lower. and sustainability of the devised solutions increases. This needs to be explored as it is not clear to what extent LHC-focused solutions can be used in other communities that ostensibly have different cultures, processing needs, and even funding models. We should caution that making any system cover an ever wider range of requirements inevitably leads to more complex solutions that are difficult to maintain and while they perform well on average they rarely perform well for any specific use.

Finally, any and all changes undertaken must not make the ease of access to data any worse than it is under current computing models. We must also be prepared to accept the fact that the best possible solution may require significant changes in the way data is handled and analyzed. What is clear is that what is being done today will not scale to the needs of HL LHC.

7.5.2 Current Approaches

The original LHC computing models (circa 2005) were built up from the simpler models used before distributed computing was a central part of HEP computing. This allowed for a reasonably clean separation between three different aspects of interacting with data: organization, management and access.

Data Organization: This is essentially how data is structured as it is written. Most data is written in flat files, in ROOT format, typically with a column-wise organization of the data. The records corresponding to these columns are compressed. The internal details of this organization are typically visible only to individual software applications.

Data Management: The key challenge here was the transition to the use of distributed computing in the form of the grid. The experiments developed dedicated data transfer and placement systems, along with catalogs, to move data between computing centers. To first order the computing models were rather static: data was placed at sites and the relevant compute jobs were sent to the right locations. Applications might interact with catalogs or, at times, the workflow management systems does this on behalf of the applications.

Data Access: Various protocols are used for direct reads (rfio, dcap, xrootd, etc.) with a given computer center and/or explicit local stagein and caching for read by jobs. Application access may use different protocols than those used by the data transfers between site.

Before the LHC turn-on and in the first years of the LHC, these three areas were to first order optimized independently. Many of the challenges were in the area of "Data Management (DM)" as the Worldwide LHC Computing Grid was commissioned. As the LHC computing matured through Run 1 and Run2, the interest has turned to optimizations spanning these three areas. For example, the recent use of "Data Federations" mixes up Data Management and Access. As we will see below, some of the foreseen opportunities towards HL-LHC may require global optimizations.

The vast majority of HEP data (in particular all event data) is written once and read many times. It is treated as read-only and never updated or modified. The data management systems in HEP have traditionally exploited this simplification to (for example) create replicas of the same data in various centers to facilitate access and in the way that workflows are structured. HEP has usually referred to this area as "Data Management (DM)", with a focus in particular on transfers, data placement at distributed computer centers and (perhaps) the storage and storage management within a given center. This narrower view mapped to how projects were actually organized and to an initial focus on particular tools which had been lacking or not robust in the pre-LHC days, e.g. data transfer tools or catalogs, A narrower 'DM' view also mapped well to the largest problems initially faced by the LHC and other scientific experiments to build a worldwide distributed computing system. Simply transferring a set of files between two computer centers reliably and efficiently was a major challenge in 2005.

Thus in this document we take a broader view than traditional "DM", and consider the combination of "Data Organization, Management and Access (DOMA)") together. We believe that this full picture of data needs in HEP will provide important opportunities for efficiency and scalability as we enter the many-Exabyte era.

7.5.3 Research and Development Roadmap and Goals

Atomic Size of Data:

Data Organization Paradigms:

Data Distribution and Caching:

Support for Query-based analysis techniques:

Rethinking Data Persistence:

Example projects:

Event-level data storage and access

- Evaluate and prototype optimal interfaces for different access patterns (simulation, reconstruction, analysis)
- Assess the impact of different access patterns on catalogs and data distribution
- Evaluate the optimal use of event stores for event-level storage and access

File-level data access

- Evaluate row-based vs. column-based access: impact of storage organization on the performance of each kind of access, potential storage format providing good performance for both
- Evaluation of declarative interfaces and in-situ processing
- Evaluate just in time decompressions schemes and mappings onto hardware architectures considering the flow of data from spinning disk to memory and application
- Investigate the long term replacement of gridftp as the primary data transfer protocol. Define metrics (performance, etc.) for evaluation.
- Benchmark end-end data delivery for the main use cases (reco, MC, various analysis workloads, etc.), what are the impediments to efficient data delivery to the CPU to and from (remote) storage? What are the necessary storage hierarchies, and how does that map into technologies foreseen?

Data caching:

- Benefit of caching for main use cases (reconstruction, analysis, simulation)
- Benefit of caching for Machine Learning-based applications, in particular for the learning phase
- Potential benefit of a CDN-like approach
- Potential benefit of a NDN-like approach (medium/long-term)

Federated Data Centers (a prototype "Data-Lake")

- Understanding the needed functionalities, including policies for managing data and replications, availability, quality of service, service levels, etc.;
- Understand how to interface a data-lake federation with heterogeneous storage systems in different sites
- Investigate how to define and manage the interconnects, network performance and bandwidth, monitoring, service quality etc. Integration of networking information and testing of advanced networking infrastructure.
- Investigate policies for managing and serving derived data sets, lifetimes, re-creation (on-demand?), caching of data, etc.

Workflow and workload management

- What does a common layer look like. Can a prototype be implemented based on well-understood functionality?
- Specify and execute workflow rather than jobs?

- Data format optimization
- Completely different thinking
  - Data access model
  - Data persistence model (How do you store your data to optimize access for analysis and processing)
  - Data distribution model (How do you provide access to data in a computing model that
  - Problem: Analysis facility needs optimized data formats and data distribution to provide reproducibility and provenance for analysis workflows
  - Problem: Distributed analysis teams with own resources, how do provide democratic access to all data
  - Problem: Fast turnaround processing with near-infinite elasticity: how to provide access and store output

7.5.4 Impact and Relevance for S2I2

Physics Impact: The very fast turnaround of analysis results that could be possible with new approaches to data access and organization would lead to rapid turnaround for new science.

Resources Impact: Optimized data access will lead to more efficient use of resources, thus holding down the overall costs of computing.

Sustainability Impact: This effort would improve the reproducibility and provenance tracking for workflows (especially analysis workflows), making physics analyses more sustainable through the lifetime of the HL-LHC.

Interest/Expertise: University groups have already pioneered significant changes to the data access model for the LHC through the development of federated storage systems, and are prepared to take this further. Other groups are currently exploring the features of modern storage systems and their possible implementation in experiments.

Leadership:

Value: All LHC experiments will benefit from new methods of data access and organization, although the implementations may vary due to the different data formats and computing models of each experiment.

Research/Innovation: This effort would rely on partnerships with data storage and access experts in the CS community, some of whom are already providing consultation in this area.

7.6 Backbone: Software Engineering, Data/Software Preservation and Reproducibility

Description: In addition to the technical advances that the institute will enable, it is essential to focus on how these advances are communicated and taken up by students, software developers (both within the experiments and outside), analysts, and researchers in other fields and members of the general public with scientific interests in HEP and big data. To do this, we focus on finding and improving best practices; determining and applying incentives; developing, coordinating and providing training; and making data and tools accessible by and useful to the public.

Best Practices:

- Document best practices within HEP
- Collaborate with empirical software engineers
  - Bring them in as unfunded partners? Or provide some institute funding? Or work with them on separate proposals for the collaboration?
  - Help them study HEP software work
  - Apply lessons in general outreach to development community and integrate into training activities
- Improve best practices: modernization of software development process for scientists
  - Individual experts can improve the software by orders of magnitude by just understanding the algorithms and intended optimizations and applying the appropriate optimizations. How can we improve the overall process that the quality of software and its optimization is better out of the box?
- Improve best practices: support for testbeds for validation and scaling
  - Ops program has the hardware. S2I2 has people and R&D for capabilities. So scaling and performance verification vs scale should be a collaboration between S2I2 and Ops program.
  - Also bring in experts from HPC centers, industry, either as official institute partners (but not funded by the institute) or on an as-needed basis
- Improve best practices: tool support
  - Packaging, distribution (as practices that the HEP community needs to create, maybe through the experiments or OSG, not general services that the S2I2 will operate for the HEP community)

Incentives:

- Work with larger software community to develop/customize incentives for sharing software, for having your software used (in discoveries, by others building off it)
  - Define metrics
- Publicize this to HEP community
  - Use blogs, webinars, talks at conferences, workshops
- Use in hiring/promotion/tenure decisions at laboratories
  - Create sample language, circulate to departments

7.7 OSG-HEP

8 Institute Organizational Structure and Evolutionary Process

The management and governance structures chosen for the Institute should answer to the following questions:

1. **Goals:** What are the goals of the Institute?
2. **Interactions:** Who are the primary clients/beneficiaries of the Institute? How are their interests represented? How can the Institute align its priorities with those of the LHC experiments?
3. **Operations:** How does the Institute execute its plan with the resources it directly controls? How does the Institute leverage and collaborate with other organizations? How does the Institute maintain transparency?
4. **Metrics:** How is the impact of the Institute evaluated? And by whom?
5. **Evolution:** What are the processes by which the Institutes areas of focus and activities evolve?

During the S2I2 conceptualization process the U.S. community had a number of discussions regarding possible structures and converged on the strawman model described show in Figure 6 as a baseline. The specific choices may evolve in an eventual implementation phase depending on funding levels, project participants, etc.

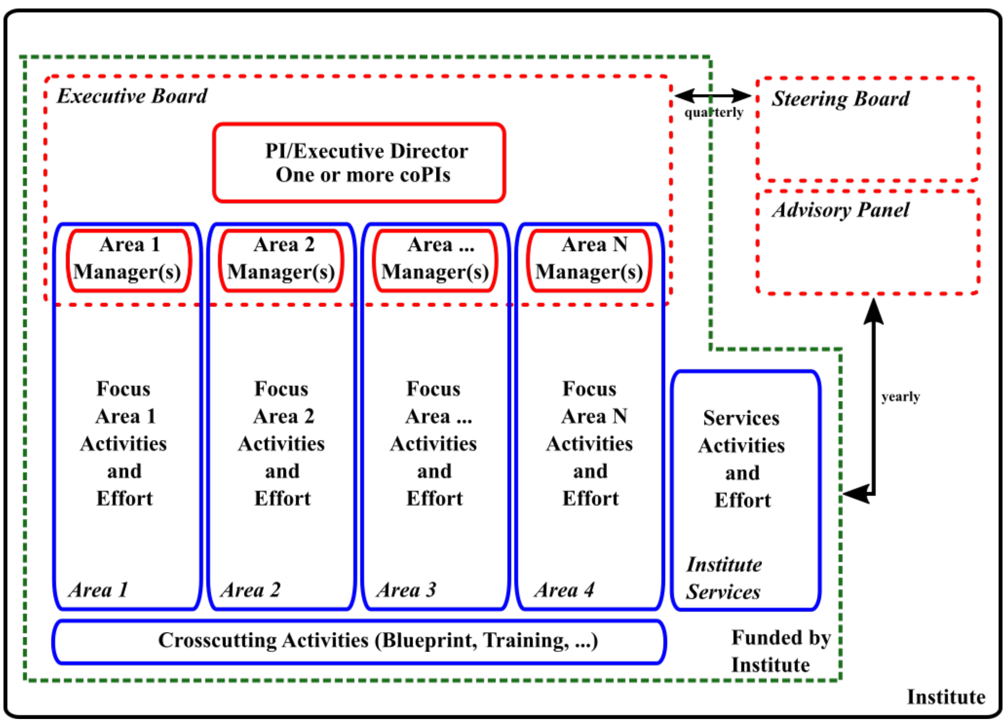

Figure 6: Strawman Model for Institute Management and Governance. **(Figure to be remade!)**

The main elements in this organizational structure and their roles within the Institute are:

PI/co-PIs: as on the eventual Institute implementation proposal, with project responsibilities as defined by NSF.

Focus Areas: A number of Focus Areas will be defined for the institute at any given point in time. These areas will represent the main priorities of the institute in terms of activities aimed at developing the software infrastructure to achieve the mission of the Institute. The S2I2-HEP conceptualization process has identified a initial set of high impact focus areas. These are described in Section 7 of this document. The number and size of focus areas which will be included in an Institute implementation will depend on funding available and resources needed to achieve the goals. The areas could also evolve over the course of the institute, however it is expected that a typical number will be 3-5. Each focus area within an Institute will have a written set of goals for the year and corresponding institute resources. The active focus areas will be reviewed together with the Advisory Panel once/year and decisions will be taken on updating the list of areas an their yearly goals, with input from the Steering Board.

Executive Board: the Executive Board will manage the day to day activities of the Institute. It will consist of the PI, coPIs and the managers of the areas of focus. A weekly meeting will be used to manage the general activities of the institute and make shorter term plans. In many cases, a liaison from other organizations (e.g. the US LHC Ops programs) would be invited as an "observer" to weekly Executive Board meetings in order to facilitate transparency and collaboration (e.g. on shared services or resources).

Area Manager(s): each Area Manager will manage the day to day activities within a focus area. It is for the moment undefined whether there will be an Area Manager plus a deputy, co-managers or a single manager. An appropriate mix of HEP, Computer Science and representation from different experiments will be a goal.

Steering Board: a Steering Board will be defined to meet with the executive team approximately quarterly to review the large scale priorities and strategy of the institute. (Areas of focus will also be reviewed, but less frequently.) The steering board will consist of two representatives for each participating experiment, plus representatives of CERN, FNAL, etc. Members of the Steering Board will be proposed by their respective organizations and accepted by the Executive Director in consultation with the Executive Board.

Executive Director: an Executive Director will manage the overall activities of the institute and its interactions with external entities. In general day-to-day decisions will be taken by achieving consensus in the Executive Board and strategy and priority decisions based on advice and recommendations by the Steering and Executive Boards. In cases where consensus cannot be reached, the Executive Director will take a final decision. It would also be prudent for the Institute to have a Deputy Director who is able to assume the duties during periods of unavailabilty of the Exeutive Director.

Advisory Panel: an Advisory Panel will be convened to conduct an internal review of the project once per year. The members of the panel will be selected by the PI/co-PIs with input from the Steering Board. The panel will include experts not otherwise involved with the institute in the areas of physics, computational physics, sustainable software development and computer science.

9 Building Partnerships

The role envisioned for the Institute in Section 6 will require collaborations and partnerships with a number of external entities.

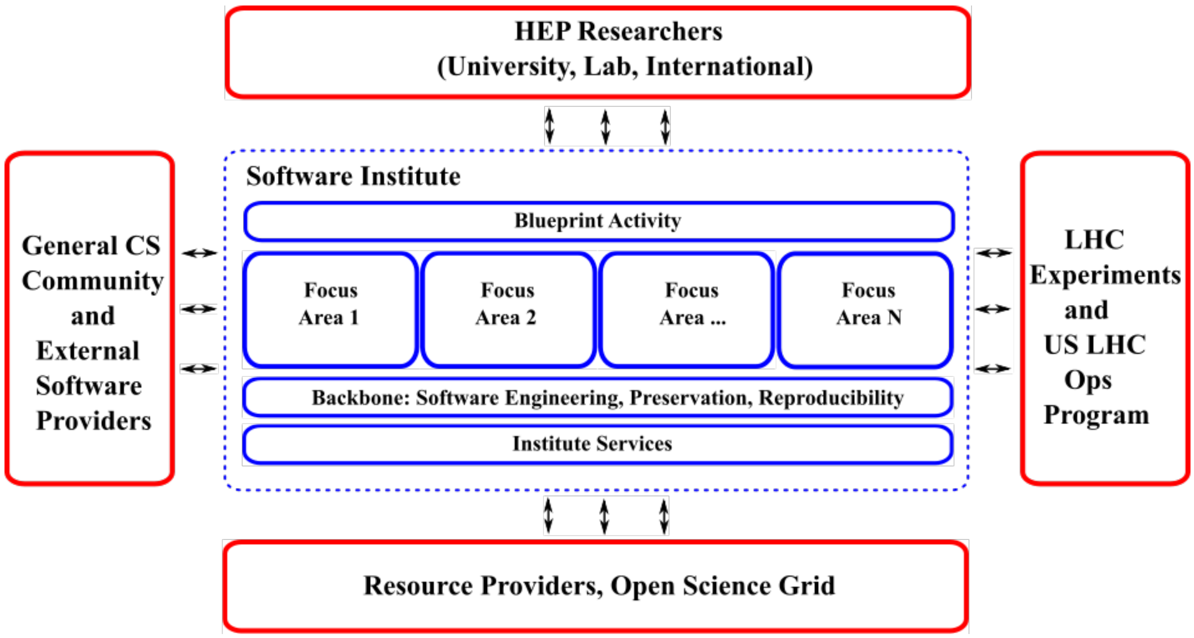

Figure 7: Relationship of the Institute to other entities

The Institute will partner with a number of other entities, as shown in Figure 8.

HEP Researchers (University, Lab, International):

LHC Experiments:

LHC Ops Programs:

Computer Science Community:

External Software Providers:

OSG-LHC:

DOE and the National Labs:

Industry: XXX projects with Intel like the Big Data Reduction Facility [44], through an Intel Parallel Computing Center [45], with Google [46, 47], with AWS [46–48] or in some cases more informally (e.g. with NVIDIA).

CERN and the HEP Software Foundation

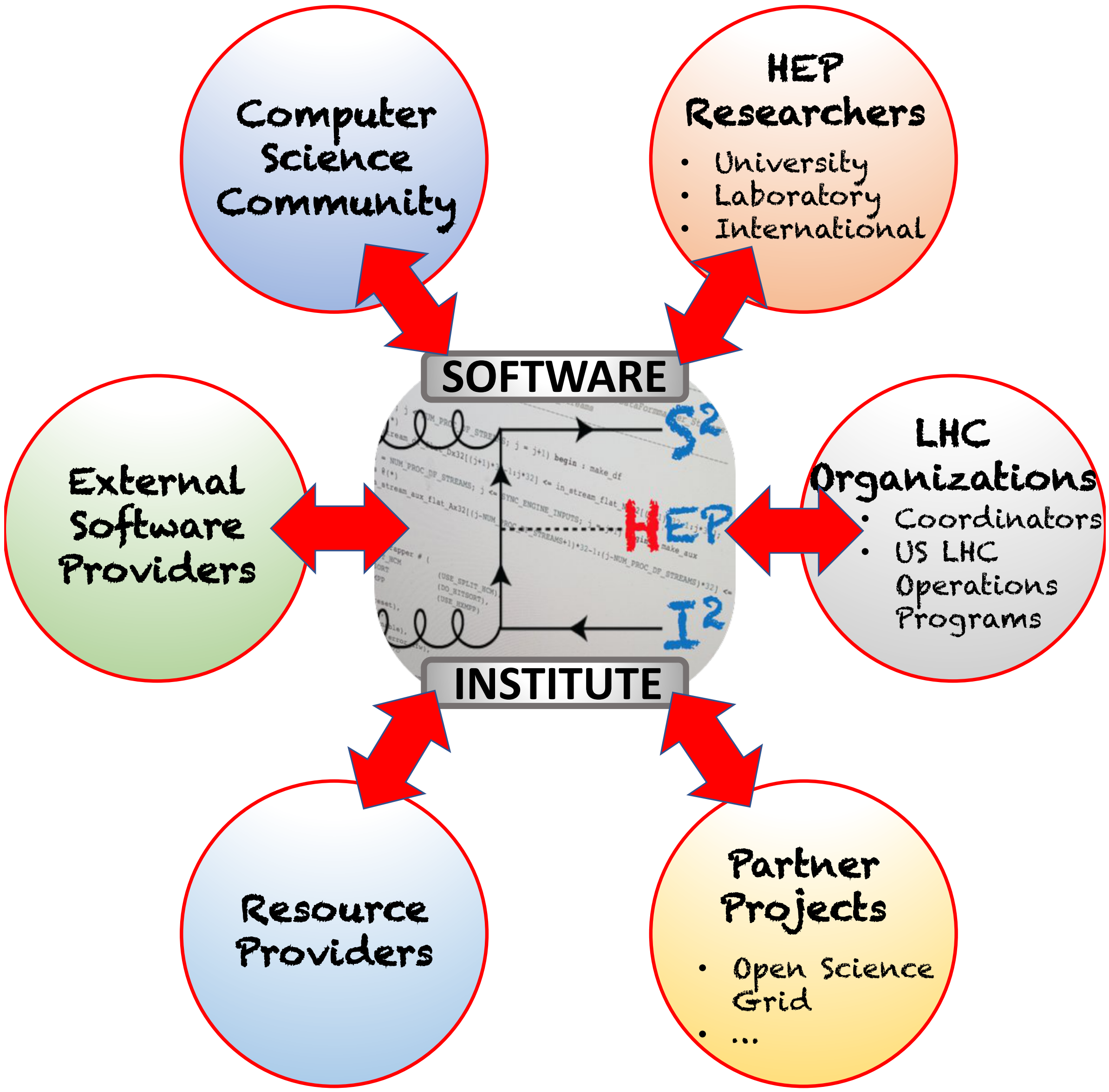

Figure 8: Relationship of the Institute to other entities

10 Metrics for Success (Physics, Software, Community Engagement)

11 Training and Workforce Development, Education and Outreach

This is a text dump of various text bits, I am still working to edit into a coherent section. – updated October 2 mds, in blue font

HEP algorithms and their implementations are designed and written by individuals with a broad spectrum of expertise in the underlying technologies, be it physics, or data science, or principles or computing, or software engineering. Almost all Ph.D. students write analysis software, as do most post-docs. Many students and post-docs write software to acquire data, calibrate and reconstruct it, and reduce data sets to sizes managable for analysis by teams and individuals. Some of these people have very high levels of domain and software engineering expertise, and some are raw recruits. For example, most experiments have dedicated teams for developing and maintaining code for tracking charged particles. The most senior members of these teams generally have many years of experience and have developed deep understandings of the current algorithms and their performances, both in terms of physics performance and resource usage. This wisdom in passed along in a somewhat haphazard way through what amounts to an unofficial apprenticeship program. In addition, teams of core developers are responsible for designing and implementing software for workflow and workload management. These people are often responsible for managing use of these tools to run what are often commonly "central productions" of reconstruction, strippng, and simulation campaigns. Members of these teams are considered software professionals, although many have been formally trained in HEP rather than computer science or software engineering. Matching the educational and training oppurtunities to the needs of the various levels of software developers across the full spectrum of the community will require carefully assessing what skills and experise will have the biggest impact on physics. In addition, as most people earning Ph.D.'s in experimental particle physics eventually leave the field, providing educational and training opportunities that prepare them for other career trajectories must be a consideration in setting priorities.

Many people in the field believe that core elements of computer science, computer programming, and software engineering should be required of *all* students embarking on a Ph.D. in experimental HEP. Some undergraduate programs provide good opportunities in this regard, but there is no universal expectation that this is prerequisite to beginning graduate level study in a U.S. university. Nor do most Ph.D. programs offer formal coursework like this. As a result, the HEP community needs to decide what it expects all of its students to know, and to prepare appropriate pedagogic material that can be used, either in the formal classroom or for independent study. Elements of this material have been assembled by individual instructors, or is taught piecemeal by experiments, but a coherent approach should be developed.

HEP has a set of concepts and a software infrastructure for analyzing data that is approximately domain-specific and transcends individual experiments. The most common analysis framework is the ROOT library developed principally at CERN. It encodes methods for selecting datasets, visualizing data, extracting parameters that describe data, etc. The community is rapidly adopting similar tools from the larger scientific Python community. Some students are introduced to these very informally by mentors who give them tutorials and/or working examples to get started. Some are provided experiment-specific tutorials (in-person or online) to get started. A software institute can take a leading role in collecting, developing, and maintaining a curated set of educational materials that addresses the common software needs of all students starting to do analysis. It can also organize video-based classes or in-person "summer schools" to teach this material.

In addition to writing analysis code, many members of the HEP community write software which becomes part of the experimental infrastructure. Examples of this are reconstruction software, event selection software (at either the trigger level or the offline "stripping" level), simulation software, and data visualization software. Each of these requires both domain expertise and algorithmic design plus software engineering expertise. Providing the training to build high-quality, performant, sustainable software for these types of applications is qualitatively different – it requires a much higher level of instructor expertise, and the target audience is generally smaller. As such a large fraction of the processing power is deployed for reconstruction, training the lead developers how to use performance tools to study hot spots and memory access patterns, how to design data structures and algorithms to take advantage of vector processors in modern architectures, and how to write thread-safe algorithms is absolutely critical to using computing resources efficiently. Similarly, if we want event selection software to use algorithms built on top of ML learning tools, we must train the developers of that software the underlying principles of ML, what tools exist, how to use those tools to train neural networks or BDTs efficiently, and how to deploy inference engines that execute quickly. In many cases, the state-of-the-art is evolving very rapidly. This means that developers will need continuing education, and much of it should be hands-on and interactive. An Institute will be a natural home for this type of training.

Where appropriate, training programs should take advantage of developments in pedagogy, such as active learning[1] or peer learning[2]. In some cases, it may be advantageous to have code samples that are purposely broken or flawed, and ask students to fix or improve them. Learning material so that it sticks with the students often takes more effort by both the students and the instructors; it often takes more time than we would prefer. However, it is the best way to ensure an educated community that can fully contribute to the physics programs at large, which is really the ultimate goal training programs.

A difficulty that has emerged in the past with respect to implementation of training courses is the lack of funding along with the lack of available time by experts in the field. People with enough expertise or insight in the field have usually no time to devote to prolonged periods of student's training, and, even when they can find some, the cost of setting up a training course in an effective way is often beyond what's made available by funding agencies (funds for travel, hosting, setting up a room with a computing infrastructure to allow interactive hands-on session, etc.). A possible way out is a completely different approach to training (but complementary to the already existing and successful classical efforts such as the CERN School of Computing's Bertinoro and Kit ones): instead of directly teaching to students, trainees could make use of a web-based platform to provide training materials to students. This complementary approach has several advantages over traditional ones:

11.1 Knowledge that needs to be transferred

At all stages of software & computing training, we should take care to encourage Good Practices Across the Community (GPAC), such as error checking, modularity of code design, writing tests, etc. All the key concepts addressed in the training should not be specific to a particular experiment or field of application, but general enough to be useful for the whole HEP community and possibly beyond. In this section, we present a list of specific concepts that need to be taught to members of the community, in order to guarantee the base level of competence needed to write efficient code for the different tasks performed in HEP experiments.

Base knowledge to be transferred includes basic programming concepts, data structures, basics of code design, error checking, code management tools, validation and debugging tools. More advanced topics include modularity of code design, advanced data structures, evaluation metrics, writing tests and working with different types of hardware accelerators. Special emphasis should be made on reporting results and documenting them.

- Basic Programming Concepts
  - Object oriented paradigm

[1] `http://www.crlt.umich.edu/tstrategies/tsal`
[2] `https://en.wikipedia.org/wiki/Peer_learning`

  - Compiled langauges (C++)
  - Scripting languages (Python, Javascript,...)
- Algorithmics
  - Boost library
  - STL algorithms for containers
  - R and/or ROOT
- Existing frameworks (development or application level)
  - Qt
  - ROOT
  - experiment specific framework (possibily if of potential interest outside the native experiment)
- Code design (design patterns)
- Development tools
  - IDEs (Integrated Development Environment)
  - Debuggers
  - Profilers
- Evaluation metrics
- Trust metrics such as data driven tests
- Specific software implementation training
- Good practices
- Code style and clarity
- Scripting and data cleaning
- Reporting results reproducibly
- Writing Documentation

11.2 Challenges

HEP collaborations typically provide opportunities for members to learn software tools. For example, the week-long CMS Data Analysis School (CMSDAS) [49] pairs software experts with new collaborators to build and run end-to-end examples of real analysis applications. Other collaboration have similar programs. A number of summer schools focused on software and computing topics also exist in the global HEP community including the CERN School of Computing [REF], the GridKa school [REF], the "Developing Efficient Large Scale Scientific Applications (ESC)" [**?**], school organized by the Istituto Nazionale di Fisica Nucleare (INFN) and (more recently) the "Computational and Data Science for High Energy Physics (CoDaS-HEP)" school [REF] in the U.S.

There are a lot of experiment-specific training efforts. But we have some common needs. We should probably strive to extract that common knowledge and build common training from that, because it enables us to duplicate less effort on experiment-specific training, and to do the shared training better by accumulating more expertise into it.

Within a single experiment, different skill sets are needed. In addition to a base skill set that contains basic programming language knowledge, testing and code management tools and experiment-specific framework knowledge, there are more specialized skills that only a subset of the community needs to know, such as software optimization, or low-level hardware interfaces.

11.3 Roadmap

- Work with the Carpentries (software & data) to customize (focusing on what is needed in HEP, making examples HEP-specific) general/basic software training for new students
- Work with HPC centers when training needs/goals overlap, e.g. DOE Lab and university computing centers that provide live, virtual, and recorded training
- Summer schools
- Focused webinars on specific topics (both beginner and advanced)
- Focused webinars on specific topics (both beginner and advanced), this could be collaborative with software, HPC, data science communities
- Provide advanced/focused hands-on in-person and virtual training on a variety of HEP-specific topics (following CMS-HATS model)
  - Coordinate with experiments & LHC physics centers, for content, instructors, and training venues
  - Initial topics: Analysis in python, analysis in R, histogramming, PyROOT and rootpy, ML to improve Physics Objects, tracking tagging, Modern Tools for Physics Analysis-Roofit, MVA
  - Method for bringing in new topics:
    * Suggestions from users and developers, user survey
    * Find willing instructors (from LHC Experiments etc)
    * Institutes role is coordinator, not funder, not instructor (though maybe will fund/help students?, pay for instructor travel for in-person training?)

11.4 Outreach

Outreach and use of HEP data by researchers in other fields and members of the public with scientific interests (linked to software/data preservation and reproducibility within Analysis focus area)

- Provide data and tools to the non-HEP researchers, e.g. computer scientists who want to work on big data problems
- Provide data and tools to the interested public
- Document data and tools and provide examples of usage
- How do members of the public get access to enough computing to work with this data? (HEP data analysis science gateway/portal?)
- Bringing together Inreach and Outreach community

12 Broadening Participation

People are the key to successful software. Computing hardware becomes obsolete after 3 – 5 years. Specific software implementations of algorithms can have somewhat longer lifetimes (or shorter). Developing, maintaining, and evolving algorithms and implementations for HEP experiments can continue for many decades. Using the LEP tunnel at CERN for a hadron collider was first considered at a workshop in 1984; the ATLAS and CMS collaborations submitted letters of intent in 1992; the CERN Council approved construction of the LHC in late 1994, and it first delivered beams in 2008. A decade later, the accelerator and the detectors are exceeding their design specifications, producing transformative science. The community is building hardware upgrades and planning for a High Luminosity LHC era which will *start* collecting data circa 10 years from now, and then acquire data for at least another decade. People, working together, across disciplines and experiments, over several generations, are the real cyber infrasturcture underlying sustainable software.

Much of the software used by HEP experiments is highly domain specific and requires domain expertise to design and build it. At the same time, developing high-quality algorithms and writing performant software implementations often requires expertise beyond HEP. The LHC community has identified the speed of reconstruction as a potential bottleneck on the path to doing the best possible HL-LHC science. Taking advantage of emerging compute and storage architectures requires working with software engineers and computer scientists who understand how to take advantage of them. Similarly, replacing the most time consuming trigger and reconstruction algorithms with radically new algorithms based on machine learning (ML) will require working closely with computer scientists and data scientists who develop the underlying ML tools we use. The software that is not so domain specific can benefit from even stronger collaborations with the worlds of computer science, network engineering, etc. A large fraction of the computing effort is expended running "centralized productions". While some of the issues of workload management and workflow management are specific to the field, and even to individual experiments, the big picture issues are much more generic. Real collaboration across disciplines, cooperation by experiments within HEP, and effective communication are necessary foundations for building sustainable cyber infrastructure to enable the full reach of the hardware investments in the HL-LHC program.

13 Sustainability

14 Risks and Mitigation

15 Estimated Costs

This section should contain the statements about what would be possible at various funding levels. (Perhaps with a better title....)

A Appendix - S^2I^2 Strategic Plan Elements

The original S2I2-HEP proposal was in response to solicitation NSF 15-553 [50]. This solicitation specified that: "The product of a conceptualization award will be a strategic plan for enabling science and education through a sustained software infrastructure that will be freely available to the community, and will address the following elements:"

- the science community and the specific grand challenge research questions that the S^2I^2 will support;
- specific software elements and frameworks that are relevant to the community, the sustainability challenges that need to be addressed, and why addressing these challenges will be transformative;
- appropriate software architectures and lifecycle processes, development, testing and deployment methodologies, validation and verification processes, end usability and interface considerations, and required infrastructure and technologies;
- the required organizational, personnel and management structures and operational processes;
- the requirements and necessary mechanisms for human resource development, including integration of education and training, mentoring of students, postdoctoral fellows as well as software professionals, and proactively addressing diversity and broadening participation;
- potential approaches for long-term sustainability of the software institute as well as the software; and
- potential risks including risks associated with establishment and execution, necessary infrastructure and associated technologies, community engagement, and long-term sustainability.

Moreover the solicitation states that "The strategic plan resulting from the conceptualization phase is expected to serve as the conceptual design upon which a subsequent S^2I^2 Implementation proposal could be based.". In this "Strategic Plan" document, we have attempted to respond to to these criteria.

We note in addition that the same solicitation (NSF 15-553 [50]) also allowed for implementation proposals for "Chemical and Materials Research" and "Science Gateways". For these implementation proposals the solictation requested the following elements in the (20 page) proposals:

- The overall rationale for the envisioned institute, its mission, and its goals.
- A set of software issues and needs and software sustainability challenges faced by a particular, well-defined yet broad community (that is clearly identified in the proposal) that can best be addressed by an institute of the type proposed, a compelling case these are the most important issues faced by the community, and that these issues are truly important.
- A clear and compelling plan of activities that shows how the proposed institute will address these issues and needs by involving (and leveraging) the community, including its software developers, in a way that will benefit the entire community.
- If there are other NSF-funded activities that might appear to overlap the institute's activities, a discussion clarifying how the funding of each activity will be independent and non-overlapping.
- Metrics of how success will be measured, that include at least impact on the developer and user communities.
- Evidence that the people involved in planning and setting up the institute have the organizational, scientific, technical, and sociocultural skills to undertake such a task, and that they are trusted and respected by the community as a whole.

- Evidence of a high degree of community buy in that a) these are the urgent/critical needs and b) this institute is the way to address them.
- A plan for management of the institute, including 1) the specific roles of the PI, co-PIs, other senior personnel and paid consultants at all institutions involved, 2) how the project will be managed across institutions and disciplines, 3) identification of the specific coordination mechanisms that will enable cross-institution and/or cross-discipline scientific integration, and 4) pointers to the budget line items that support these management and coordination mechanisms.
- A steering committee composed of leading members of the targeted community that will assume key roles in the leadership and/or management of the institute. A brief biography of the members of the steering committee and their role in the conceptualization process should be included.
- A plan for how the institute activities will continue and/or the value of the institutes products will be preserved after the award, particularly if it does not receive additional funds from NSF.

As these criteria are general enough to be relevant also for an S2I2 for HEP, we have included also some initial information on these items in this document.

B Appendix - Workshop List

During the process we have organized a number of workshops and sessions at preexisting meetings. These included (in chronological order):

S2I2 HEP/CS Workshop
Date: 7-9 Dec, 2016
Location: University of Illinois at Urbana-Champaign
URL: `https://indico.cern.ch/event/575443/`
Summary report: `http://s2i2-hep.org/downloads/s2i2-hep-cs-workshop-summary.pdf`
Description: This workshop brought together attendees from both the particle physics and computer science (CS) communities to understand how the two communities could work together in the context of a future NSF Software Institute aimed at supporting particle physics research over the long term. While CS experience and expertise has been brought into the HEP community over the years, this was a fresh look at planned HEP and computer science research and brainstorm about engaging specific areas of effort, perspectives, synergies and expertise of mutual benefit to HEP and CS communities, especially as it relates to a future NSF Software Institute for HEP.

HEP Software Foundation Workshop
Date: 23-26 Jan, 2017
Location: UCSD/SDSC (La Jolla, CA)
URL: `http://indico.cern.ch/event/570249/`
Description: This HSF workshop at SDSC/UCSD was the first workshop supporting the CWP process. There were plenary sessions covering topics of general interest as well as parallel sessions for the many topical working groups in progress for the CWP.

S2I2-HEP/OSG/USCMS/USAtlas Panel
Date: 8 Mar, 2017
Location: UCSD/SDSC (La Jolla, CA)
URL: `https://indico.fnal.gov/conferenceTimeTable.py?confId=12973#20170308`
Description: This panel took place at Open Science Grid All Hands Meeting (OSG-AHM). Participants included Kaushik De (USAtlas), Peter Elmer (S2I2-HEP, USCMS), Oli Gutsche (USCMS) and Mark Neubauer (S2I2-HEP, USAtlas), with Frank Wuerthwein (OSG, USCMS) as moderator. The goal was to inform the OSG community about the CWP and S2I2-HEP processes and learn from the OSG experience.

Software Triggers and Event Reconstruction WG meeting
Date: 9 Mar, 2017
Location: LAL-Orsay (Orsay, France)
URL: `https://indico.cern.ch/event/614111/`
Description: This was a meeting of the Software Triggers and Event Reconstruction CWP working group. It was held as a parallel session at the "Connecting the Dots" workshop, which focuses on forward-looking pattern recognition and machine learning algorithms for use in HEP.

IML Topical Machine Learning Workshop
Date: 20-22 Mar, 2017
Location: CERN (Geneva, Switzerland)
URL: `https://indico.cern.ch/event/595059`

Description: This was a meeting of the Machine Learning CWP working group. It was held as a parallel session at the "Inter-experimental Machine Learning (IML)" workshop, an organization formed in 2016 to facilitate communication regarding R&D on ML applications in the LHC experiments.

Community White Paper Follow-up at FNAL

Date: 23 Mar, 2017
Location: FNAL (Batavia, IL)
URL: `https://indico.fnal.gov/conferenceDisplay.py?confId=14032`
Description: This one-day workshop was organized to engage with the experimental HEP community involved in computing and software for Intensity Frontier experiments at FNAL. Plans for the CWP and the S2I2-HEP project were described, with discussion about commonalities between the HL-LHC challenges and the challenges of the FNAL neutrino and muon experiments.

CWP Visualization Workshop

Date: 28-30 Mar, 2017
Location: CERN (Geneva, Switzerland)
URL: `https://indico.cern.ch/event/617054/`
Description: This workshop was organized by the Visualization CWP working group. It explored the current landscape of HEP visualization tools as well as visions for how these could evolve. There was participation both from HEP developers and industry.

2nd S2I2 HEP/CS Workshop

Date: 1-3 May, 2017
Location: Princeton University (Princeton, NJ)
URL: `https://indico.cern.ch/event/622920/`
Description: This 2nd HEP/CS workshop built on the discussions which took place at the the first S2I2 HEP/CS workshop to take a fresh look at planned HEP and computer science research and brainstorm about engaging specific areas of effort, perspectives, synergies and expertise of mutual benefit to HEP and CS communities, especially as it relates to a future NSF Software Institute for HEP.

DS@HEP 2017 (Data Science in High Energy Physics)

Date: 8-12 May, 2017
Location: FNAL (Batava, IL)
URL: `https://indico.fnal.gov/conferenceDisplay.py?confId=13497`
Description: This was a meeting of the Machine Learning CWP working group. It was held as a parallel session at the "Data Science in High Energy Physics (DS@HEP)" workshop, a workshop series begun in 2015 to facilitate communication regarding R&D on ML applications in HEP.

HEP Analysis Ecosystem Retreat

Date: 22-24 May, 2017
Location: Amsterdam, the Netherlands
URL: `http://indico.cern.ch/event/613842/`
Summary report: `http://hepsoftwarefoundation.org/assets/AnalysisEcosystemReport20170804.pdf`
Description: This was a general workshop, organized about the HSF, about the ecosystem of anal-

ysis tools used in HEP and the ROOT software framework. The workshop focused both on the current status and the 5-10 year time scale covered by the CWP.

CWP Event Processing Frameworks Workshop
Date: 5-6 Jun, 2017
Location: FNAL (Batavia, IL)
URL: `https://indico.fnal.gov/conferenceDisplay.py?confId=14186`
Description: This was a workshop held by the Event Processing Frameworks CWP working group.

HEP Software Foundation Workshop
Date: 26-30 Jun, 2017
Location: LAPP (Annecy, France)
URL: `https://indico.cern.ch/event/613093/`
Description: This was the final general workshop for the CWP process. The CWP working groups came together to present their status and plans, and develop consensus on the organization and context for the community roadmap. Plans were also made for the CWP writing phase that followed in the few months following this last workshop.

S2I2-HEP Workshop
Date: 23-26 Aug, 2017
Location: University of Washington, Seattle (Seattle, WA)
URL: `https://indico.cern.ch/event/640290/`
Description: This final S2I2-HEP workshop was held as a satellite workshop of the ACAT 2017 Conference. The workshop built on the emerging consensus from the CWP process and focused on the role an NSF-supported Software Institute could play. Specific discussions focused on establishing which areas would be both high impact and appropriate for leadership role in the U.S. universities. In addition the relative roles of an Institute, the US LHC Ops programs and the international LHC program were discussed, along with possible management structures for an Institute.

This full list of workshops and meetings (with links) is also available on the `http://s2i2-hep.org` website. In addition there were "internal" sessions regarding the CWP in the LHC experiment collaboration meetings, which are not listed above.

More than 250 people participated in one or more of the workshops which had an explicit registration and participant list. This does not include those who participated in the many "outreach" or panel sessions at pre-existing workshops/meetings such as DS@HEP, the OSG AHM, the IML Workshop or the sessions at LHC experiment collaboration meetings which not listed above, for which no explicit participant list was tracked. The combined list of known registered participants is:

Aaron Elliott (Aegis Research Labs), Aaron Sauers (Fermilab), Aashrita Mangu (California Institute of Technology), Abid Patwa (DOE), Adam Aurisano (University of Cincinnati), Adam Lyon (FNAL), Ajit Majumder (Wayne State), Alexei Klimentov (Brookhaven National Lab), Alexey Svyatkovskiy (Princeton University), Alja Mrak Tadel (Univerity California San Diego), Amber Boehnlein (Jefferson Lab), Amir Farbin (University of Texas at Arlington), Amit Kumar (Southern Methodist), Andrea Dotti (SLAC National Accelerator Laboratory), Andrea Rizzi (INFN-Pisa), Andrea Valassi (CERN), Andrei Gheata (CERN), Andrew Gilbert (KIT), Andrew Hanushevsky (SLAC National Accelerator Laboratiry), Anton Burtsev (University of California, Irvine), Anton Poluektov (University of Warwick), Antonio Augusto Alves Junior (University of Cincinnati), Antonio Limosani (CERN / University of Sydney), Anyes Taffard (UC Irvine), Ariel Schwartzman (SLAC), Attila Krasznahorkay (CERN), Avi Yagil (UCSD), Axel Naumann (CERN), Ben Hooberman (Illinois), Benedikt Hegner (CERN), Benedikt Riedel (University of Chicago), Benjamin Couturier (CERN), Bill Nitzberg (Altair), Bo Jayatilaka (FNAL), Bogdan Mihaila (NSF), Brian Bockelman (University of Nebraska - Lincoln), Burt Holzman (Fermilab), Carlos Maltzahn (University of California - Santa Cruz), Catherine Biscarat (CNRS), Cecile Barbier (LAPP), Charles Leggett (LBNL), Charlotte Lee (University of Washington), Chris Green (FNAL), Chris Tunnell (University of Chicago, KICP), Christopher Jones (FNAL), Claudio Grandi (INFN), Conor Fitzpatrick (EPFL), Dan Katz (University of Illinois at Urbana-Champaign/NCSA), Dan Riley (Cornell University), Daniel Whiteson (UC Irvine), Daniele Bonacorsi (University of Bologna), Danko Adrovic (DePaul), Dario Berzano (CERN), Dario Menasce (INFN Milano-Bicocca), David Abdurachmanov (University of Nebraska-Lincoln), David Lange (Princeton University), David Lesny (Illinois), David Malon (Argonne National Laboratory), David Rousseau (LAL-Orsay), David Smith (CERN), Dick Greenwood (Louisiana Tech University), Dirk Duellmann (CERN), Dirk Hufnagel (Fermilab), Don Petravick (Illinois/NCSA), Dorian Kcira (California Institute of Technology), Doug Benjamin (Duke University), Doug Thain (Notre Dame), Douglas Thain (University of Notre Dame), Dustin Anderson (California Institute of Technology), Dustin Tran (Columbia University), Eduardo Rodrigues (University of Cincinnati), Elizabeth Sexton-Kennedy (FNAL), Enric Tejedor Saavedra (CERN), Eric Lancon (BNL), Eric Vaandering (FNAL), Farah Hariri (CERN), Federico Carminati (CERN), Fernanda Psihas (Indiana University), Fons Rademakers (CERN), Frank Gaede (DESY), Frank Wuerthwein (University of California at San Diego/SDSC), Frederique Chollet (LAPP), Gabriel Perdue (Fermilab), Gerardo Ganis (CERN), Gerhard Raven (Nikhef), Giacomo Govi (FNAL), Giacomo Tenaglia (CERN), Gianluca Cerminara (CERN), Giulio Eulisse (CERN), Gloria Corti (CERN), Gordon Watts (University of Washington), Graeme Stewart (University of Glasgow), Graham Mackintosh (IBM), Hadrien Grasland (Universite de Paris-Sud), Harvey Newman (Caltech), Helge Meinhard (CERN), Henry Schreiner III (University of Cincinnati), Horst Severini (University of Oklahoma), Ian Bird (CERN), Ian Collier (RAL), Ian Cosden (Princeton University), Ian Fisk (Simons Foundation), Ian Stockdale (Altair Engineering), Ilija Vukotic (University of Chicago), Isobel Ojalvo (Princeton University), Ivo Jimenez UC (University of California - Santa Cruz), Jakob Blomer (CERN), Jamie Bedard (Siena College), Jean Jacquemier (LAPP), Jean-Roch Vlimant (California Institute of Technology), Jeff Carver (University of Alabama), Jeff Hammond (Intel), Jeff Porter (LBNL), Jeff Templon (Nikhef), Jeffrey Carver (University of Alabama), Jerome Lauret (BNL), Jim Kowalkowski (FNAL), Jim Pivarski (Princeton University), Johannes Albrecht (TU Dortmund), John Apostolakis (CERN), John Harvey (CERN), John Towns (Illinois/NCSA), Joon Kim (Princeton University), Joseph Boudreau (University of Pittsburgh), Justas Balcas (Caltech), Justin Wozniak (University of Chicago/ANL), Karan Bhatia (Google Cloud), Karen Tomko (Ohio Supercomputer Center), Kathryn Huff (Illinois), Kaushik De (University of Texas at Arlington), Ken Bloom (University of Nebraska-Lincoln), Kevin Lannon (University of Notre Dame), Konstantin Toms (University of New Mexico), Kurt Rinnert (U.Liverpool), Kyle Chard (University of Chicago), Kyle Cranmer (New York University), Kyle Knoepfel (FNAL), Lawrence R Frank (UCSD), Lindsey Gray (Fermilab), Liz Sexton-Kennedy (FNAL), Lorenzo Moneta (CERN), Lothar Bauerdick (FNAL), Louis Capps (NVIDIA), Lukas Heinrich (New York University), Lukasz Kreczko (Bristol), Madeline Hagen (Siena College), Makoto Asai (SLAC), Manish Parashar (Rutgers University), Marc Paterno (FNAL), Marc Verderi (Ecole Polytechnique), Marcin Nowak (CERN), Maria Girone (CERN), Maria Spiropulu (Caltech), Mario Lassnig (CERN), Mark Neubauer (University of Illinois at Urbana-Champaign), Markus Klute (MIT), Markus Schulz (CERN), Martin Ritter (LMU Munich), Matevz Tadel (UCSD), Matthew Bellis (Siena College), Matt Zhang (Illinois), Matthew Feickert (Southern Methodist University), Matthew Turk (University of Illinois), Matthieu Lefebvre (Princeton University), Max Baak (KPMG), Meghan Frate (University of California, Irvine), Meghan Kane (SoundCloud, MIT), Michael Andrews (Carnegie Mellon University/CERN), Michael

Kirby (FNAL), Michael Sevilla (University of California, Santa Cruz), Michael Sokoloff (University of Cincinnati), Michel Jouvin (LAL/Universite de Paris-Sud), Michela Paganini (Yale University), Michela Taufer (University of Delaware), Mike Hildreth (University of Notre Dame), Mike Williams (MIT), Miron Livny (University of Wisconsin-Madison), Mohammad Al-Turany (GSI), Nadine Neyroud (LAPP), Nan Niu (University of Cincinnati), Nancy Wilkins-Diehr (University of California San Diego), Nathalie Rauschmayr (CERN), Neil Ernst (Software Engineering Institute), Noah Watkins (University of California, Santa Cruz), Oliver Gutsche (FNAL), Oliver Keeble (CERN), Paolo Calafiura (LBNL), Parag Mhashilkar (Fermilab), Patricia Mendez Lorenzo (CERN), Patrick Bos (Netherlands eScience Center), Patrick Skubic (University of Oklahoma), Patrick de Perio (Columbia University), Paul Laycock (CERN), Paul Mattione (Jefferson Lab), Paul Rossman (Google Inc.), Pere Mato (CERN), Peter Elmer (Princeton University), Peter Hristov (CERN), Peter Onyisi (University of Texas at Austin), Philippe Canal (FNAL), Pierre Aubert (LAPP), Rajesh Ranganath (Princeton University), Riccardo Maria Bianchi (University of Pittsburgh), Richard Hay Jr (Princeton University), Richard Mount (SLAC), Rick Wagner (Globus), Rob Gardner (University of Chicago), Rob Kutschke (FNAL), Rob Quick (Indiana University), Robert Illingworth (Fermilab), Robert Kalescky (Southern Methodist), Robert Knight (Princeton University), Robert Kutschke (Fermilab), Roger Jones (Lancaster), Ruslan Mashinistov (University of Texas at Arlington), Sabine Elles (LAPP), Sally Seidel (New Mexico), Sandra Gesing (University of Notre Dame), Sandro Wenzel (CERN), Sascha Caron (Nikhef), Sebastien Binet (IN2P3/LPC), Sergei Gleyzer (University of Florida), Shantenu Jha (Rutgers University), Shawn McKee (University of Michigan), Simone Campana (CERN), Slava Krutelyov (University of California at San Diego), Spencer Smith (McMaster University), Stefan Roiser (CERN), Steven Schramm (Universite de Geneve), Sudhir Malik (University of Puerto Rico Mayaguez), Sumanth Mannam (DePaul), Sumit Saluja (Princeton University), Sunita Chandrasekaran (University of Delaware), Tanu Malik (Depaul University), Taylor Childers (Argonne Nat. Lab), Thomas Hacker (Purdue University), Thomas Kuhr (LMU), Thomas McCauley (University of Notre Dame), Thomas Vuillaume (LAPP), Thorsten Kollegger (GSI), Tom Gibbs (NVIDIA), Tommaso Boccali (INFN Pisa), Torre Wenaus (BNL), V. Daniel Elvira (Fermilab), Vakho Tsulaia (LBNL), Valentin Kuznetsov (Cornell University), Vassil Vassilev (Princeton University), Vincent Croft (Nikhef), Vinod Gupta (Princeton University), Vladimir Gligorov (CNRS), Wahid Bhimji (NERSC/LBNL), Wenjing Wu (Institute of High Energy Physics, Beijing), Wouter Verkerke (Nikhef)

References

[1] G. Aad et al. Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. *Phys.Lett.*, B716:1–29, 2012.

[2] Serguei Chatrchyan et al. Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. *Phys.Lett.*, B716:30–61, 2012.

[3] Gino Isidori, Yosef Nir, and Gilad Perez. Flavor Physics Constraints for Physics Beyond the Standard Model. *Ann.Rev.Nucl.Part.Sci.*, 60:355, 2010.

[4] Particle Physics Project Prioritization Panel. Building for Discovery: Strategic Plan for U.S. Particle Physics in the Global Context. `http://science.energy.gov/~/media/hep/hepap/pdf/May%202014/FINAL_DRAFT2_P5Report_WEB_052114.pdf`.

[5] ALICE Collaboration public website. `http://aliceinfo.cern.ch/`.

[6] Samuel H. Fuller and Editors; Committee on Sustaining Growth in Computing Performance; National Research Council Lynette I. Millett. *The Future of Computing Performance: Game Over or Next Level?* The National Academies Press, 2011.

[7] M. Butler, R. Mount, and M. Hildreth. Snowmass 2013 Computing Frontier Storage and Data Management. *ArXiv e-prints*, November 2013.

[8] ROOT home page. `http://root.cern.ch/drupal/`.

[9] V.N. Ivanchenko. Geant4 toolkit for simulation of HEP experiments. *Nucl.Instrum.Meth.*, A502:666–668, 2003.

[10] John Allison, K. Amako, J. Apostolakis, H. Araujo, P.A. Dubois, et al. Geant4 developments and applications. *IEEE Trans.Nucl.Sci.*, 53:270, 2006.

[11] G. Barrand et al. GAUDI - The software architecture and framework for building LHCb data processing applications. In *Proceedings, 11th International Conference on Computing in High-Energy and Nuclear Physics (CHEP 2000)*, pages 92–95, 2000.

[12] Wouter Verkerke and David P. Kirkby. The RooFit toolkit for data modeling. *eConf*, C0303241:MOLT007, 2003.

[13] Andreas Hoecker, Peter Speckmayer, Joerg Stelzer, Jan Therhaag, Eckhard von Toerne, and Helge Voss. TMVA: Toolkit for Multivariate Data Analysis. *PoS*, ACAT:040, 2007.

[14] Kosyakov S. et al. FRONTIER: HIGH PERFORMANCE DATABASE ACCESS USING STANDARD WEB COMPONENTS IN A SCALABLE MULTI-TIER ARCHITECTURE. In *Proceedings, 14th International Conference on Computing in High-Energy and Nuclear Physics (CHEP 2004)*, 2004.

[15] A Dorigo, P Elmer, F Furano, and A Hanushevsky. XROOTD - A highly scalable architecture for data access. *WSEAS Transactions on Computers*, 4.3, 2005.

[16] Patrick Fuhrmann. dCache: the commodity cache. In *In Twelfth NASA Goddard and Twenty First IEEE Conference on Mass Storage Systems and Technologies*, 2004.

[17] Jakob Blomer, Carlos Aguado-Sanchez, Predrag Buncic, and Artem Harutyunyan. Distributing LHC application software and conditions databases using the CernVM file system. *Journal of Physics: Conference Series*, 331(4):042003, 2011.

[18] Eulisse G. and Tuura L. IgProf profiling tool. In *Proceedings, 14th International Conference on Computing in High-Energy and Nuclear Physics (CHEP 2004)*, 2004.

[19] Douglas Thain and Miron Livny. Parrot: Transparent user-level middleware for data-intensive computing. *Scalable Computing: Practice and Experience*, 6(3), 2005.

[20] T. Gleisberg, Stefan. Hoeche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert, and J. Winter. Event generation with SHERPA 1.1. *JHEP*, 02:007, 2009.

[21] Michelangelo L. Mangano, Fulvio Piccinini, Antonio D. Polosa, Mauro Moretti, and Roberto Pittau. ALPGEN, a generator for hard multiparton processes in hadronic collisions. *Journal of High Energy Physics*, 2003(07):001, 2003.

[22] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. FastJet User Manual. *Eur. Phys. J.*, C72:1896, 2012.

[23] Matteo Cacciari and Gavin P. Salam. Dispelling the N^3 myth for the k_t jet-finder. *Phys. Lett.*, B641:57–61, 2006.

[24] Computing Resources Scrutiny Group Report, April 2015. `http://cds.cern.ch/record/2002240/files/CERN-RRB-2015-014.pdf`.

[25] Charge for producing a hsf community white paper. `http://hepsoftwarefoundation.org/assets/CWP-Charge-HSF.pdf`.

[26] Hsf community white paper webpages. `http://hepsoftwarefoundation.org/activities/cwp.html`.

[27] S.Campana, presentation to the 2016 Aix-les-Bains ECFA HL-LHC workshop, 3 Oct 2016. `https://indico.cern.ch/event/524795/contributions/2236590/attachments/1347419/2032314/ECFA2016.pdf`.

[28] ATLAS Phase-II Upgrade Scoping Document. Technical Report CERN-LHCC-2015-020. LHCC-G-166, CERN, Geneva, Sep 2015.

[29] D Contardo, M Klute, J Mans, L Silvestris, and J Butler. Technical Proposal for the Phase-II Upgrade of the CMS Detector. Technical Report CERN-LHCC-2015-010. LHCC-P-008. CMS-TDR-15-02, Geneva, Jun 2015. Upgrade Project Leader Deputies: Lucia Silvestris (INFN-Bari), Jeremy Mans (University of Minnesota) Additional contacts: Lucia.Silvestris@cern.ch, Jeremy.Mans@cern.ch.

[30] LHCb Trigger and Online Upgrade Technical Design Report. Technical Report CERN-LHCC-2014-016. LHCB-TDR-016, May 2014.

[31] P Buncic, M Krzewicki, and P Vande Vyvre. Technical Design Report for the Upgrade of the Online-Offline Computing System. Technical Report CERN-LHCC-2015-006. ALICE-TDR-019, Apr 2015.

[32] I Bird, P Buncic, F Carminati, M Cattaneo, P Clarke, I Fisk, M Girone, J Harvey, B Kersevan, P Mato, R Mount, and B Panzer-Steindel. Update of the Computing Models of the WLCG and the LHC Experiments. Technical Report CERN-LHCC-2014-014. LCG-TDR-002, Apr 2014.

[33] Fons Rademakers and Rene Brun. Root: an object-oriented data analysis framework. *Linux J.*, page 6.

[34] R. Aaij, S. Amato, L. Anderlini, S. Benson, M. Cattaneo, M. Clemencic, B. Couturier, M. Frank, V.V. Gligorov, T. Head, C. Jones, I. Komarov, O. Lupton, R. Matev, G. Raven, B. Sciascia, T. Skwarnicki, P. Spradlin, S. Stahl, B. Storaci, and M. Vesterinen. Tesla : an application for real-time data analysis in High Energy Physics. *Comput. Phys. Commun.*, 208(CERN-LHCB-DP-2016-001. CERN-LHCB-DP-2016-001):35–42. 8 p, Apr 2016. 14 pages, 8 figures.

[35] R Abreu. The upgrade of the ATLAS High Level Trigger and Data Acquisition systems and their integration. Technical Report ATL-DAQ-PROC-2014-002, CERN, Geneva, May 2014.

[36] CMS Collaboration. Search for narrow resonances in dijet final states at $\sqrt{s} = 8$ TeV with the novel CMS technique of data scouting. *Phys. Rev. Lett.*, 117(CMS-EXO-14-005. CMS-EXO-14-005. CERN-EP-2016-090):031802. 17 p, Apr 2016. Replaced with published version. All the figures and tables can be found at http://cms-results.web.cern.ch/cms-results/public-results/publications/EXO-14-005/index.html.

[37] SciKit-Learn webpage: http://scikit-learn.org/.

[38] F. Chollet, Keras (2017), GitHub, https://github.com/fchollet/keras.

[39] A. Rogozhnikov *et al.*, rep (2017), GitHub https://github.com/yandex/rep.

[40] A. Rogozhnikov *et al.*, HepML (2017), GitHub https://github.com/arogozhnikov/hep_ml.

[41] J. Snoek, Spearmint (2017), Github: https://github.com/HIPS/Spearmint.

[42] Philip Ilten, Mike Williams, and Yunjie Yang. Event generator tuning using Bayesian optimization. 2016.

[43] TuneMC GitHub repository: https://github.com/yunjie-yang/TuneMC.

[44] https://cms-big-data.github.io. `https://cms-big-data.github.io`.

[45] Ipcc root - princeton/intel parallel computing center to modernize the root math and i/o libraries. `https://ipcc-root.github.io`.

[46] Hepcloud: a new paradigm for particle physics computing. `http://hepcloud.fnal.gov`.

[47] Hepcloud: Provisioning 160,000 compute cores for science. `http://hepcloud.fnal.gov/wp-content/uploads/2016/05/HEPCloud-DPF.pdf`.

[48] B. Holzman, L. A. T. Bauerdick, B. Bockelman, D. Dykstra, I. Fisk, S. Fuess, G. Garzoglio, M. Girone, O. Gutsche, D. Hufnagel, H. Kim, R. Kennedy, N. Magini, D. Mason, P. Spentzouris, A. Tiradani, S. Timm, and E. W. Vaandering. HEPCloud, a New Paradigm for HEP Facilities: CMS Amazon Web Services Investigation. *ArXiv e-prints*, September 2017.

[49] S. Malik, F. Hoehle, K. Lassila-Perini, A. Hinzmann, R. Wolf, et al. Maintaining and improving of the training program on the analysis software in CMS. *J.Phys.Conf.Ser.*, 396:062013, 2012.

[50] NSF 15-553. `https://www.nsf.gov/pubs/2015/nsf15553/nsf15553.htm`.