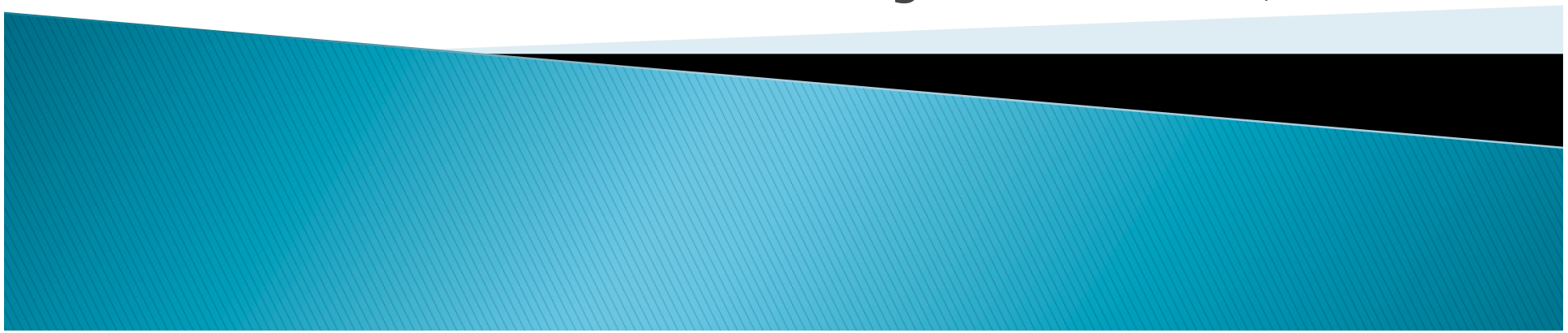
Re-engineering Frameworks for Concurrency

FNAL, 21-22 November 2011

B. Hegner & P. Mato, CERN



Vision

- Universal framework for simulation, reconstruction, analysis, high level trigger applications
- Common framework for use by any experiment
- Decomposition of the processing of each event into 'tasks' that can be executed concurrently
- Ability to process several events concurrently
- Optimal scheduling and associated data structures
- Minimize any processing requiring exclusive access to resources because it breaks concurrency
- Supporting various hardware/software technologies
- Facilitate the integration of existing LHC applications code (algorithmic part)
- Quick delivery of running prototypes. The opportunity of the 18 months LHC shutdown

Why?

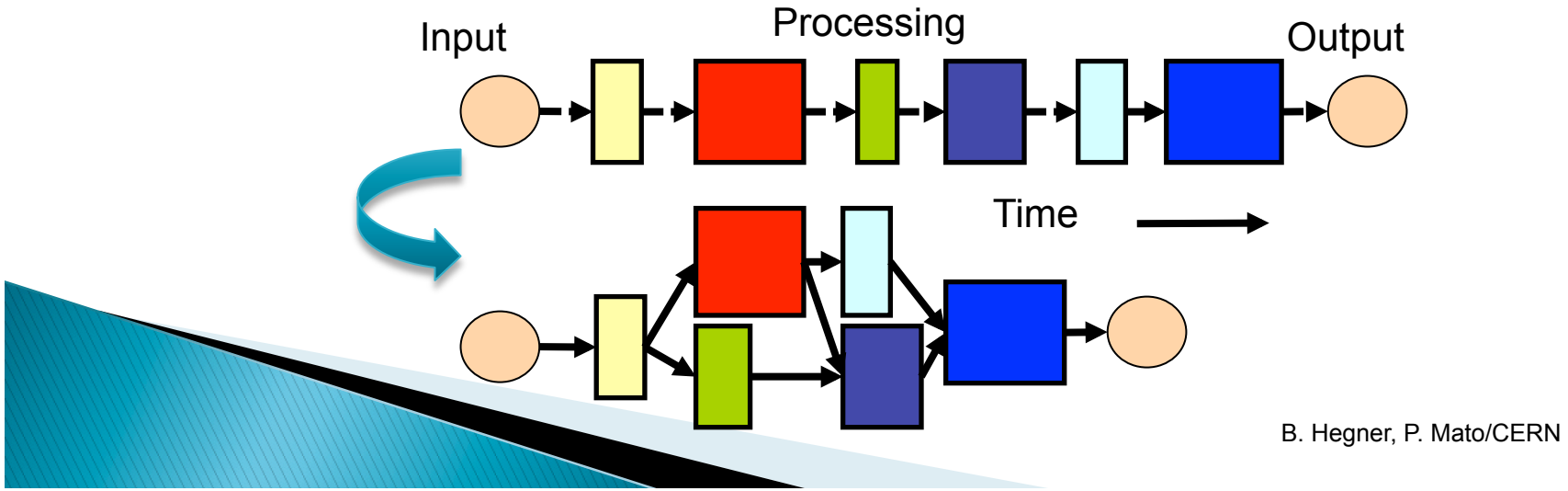
- We need to adapt current applications to the new many-core architectures
  - Expected no change in the overall throughput with respect trivial one-job-per-core parallelism
  - Scaling to a much larger number of cores
- Reducing the required resources per core
  - I/O bandwidth
  - Memory
  - Connections to DB, open files, etc.
- Reduce latency for single jobs (e.g. trigger, user analysis)
  - Run a given job in less time making use of available cores

Why the Framework managing the concurrency?

- Concrete algorithms can be parallelized with some effort
  - Making use of Threads, OpenMP, MPI, GPUs, etc.
  - But difficult to integrate them in a complete application
    - E.g. MT-G4 with Parallel Gaudi
  - Performance-wise only makes sense to parallelize the complete application and not only parts
- Developing and validating parallel code is difficult
  - 'Physicists' should be saved from this
  - Concurrency will limit what can and can not be done in the algorithmic code (policies)
- At the Framework level you have the overall view and control of the application

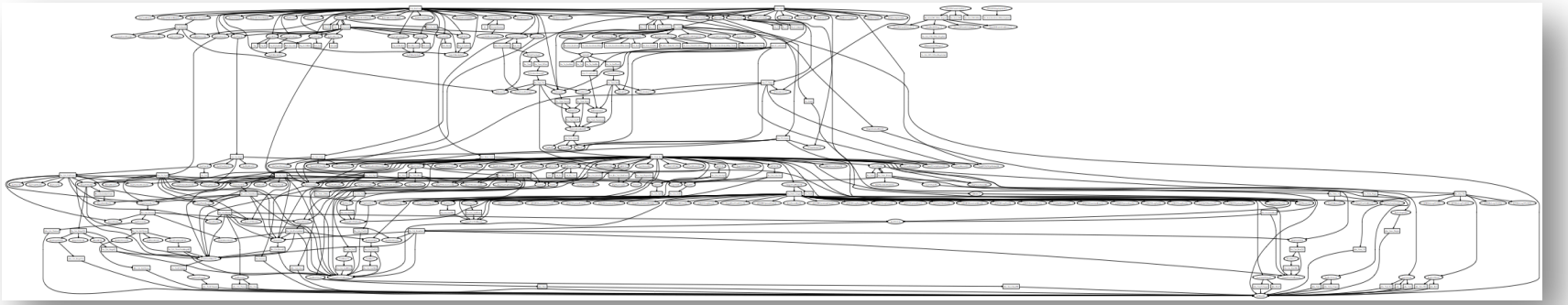
Concurrent 'Task' processing

- Framework with the ability to schedule modules/algorithms concurrently
  - Full data dependency analysis would be required (no global data or hidden dependencies)
  - Need to resolve the DAGs (Direct Acyclic Graphs) statically and dynamically
- Not much gain expected with today's designed 'Tasks'
  - Algorithm decomposition can be influenced by the framework capabilities
- 'Tasks' could be processed by different hardware/software
  - CPU, GPU, threads, process, etc.

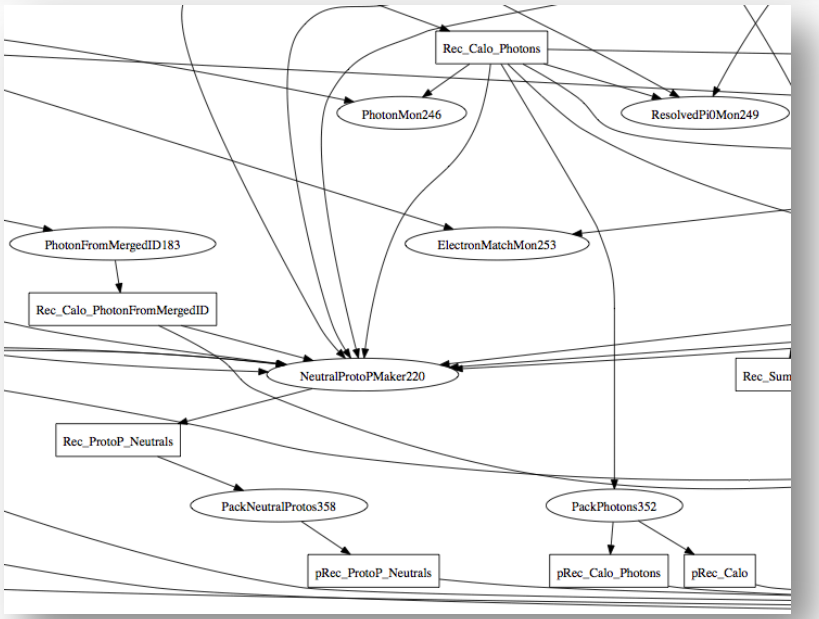


B. Hegner, P. Mato/CERN

Example: LHCb Reconstruction



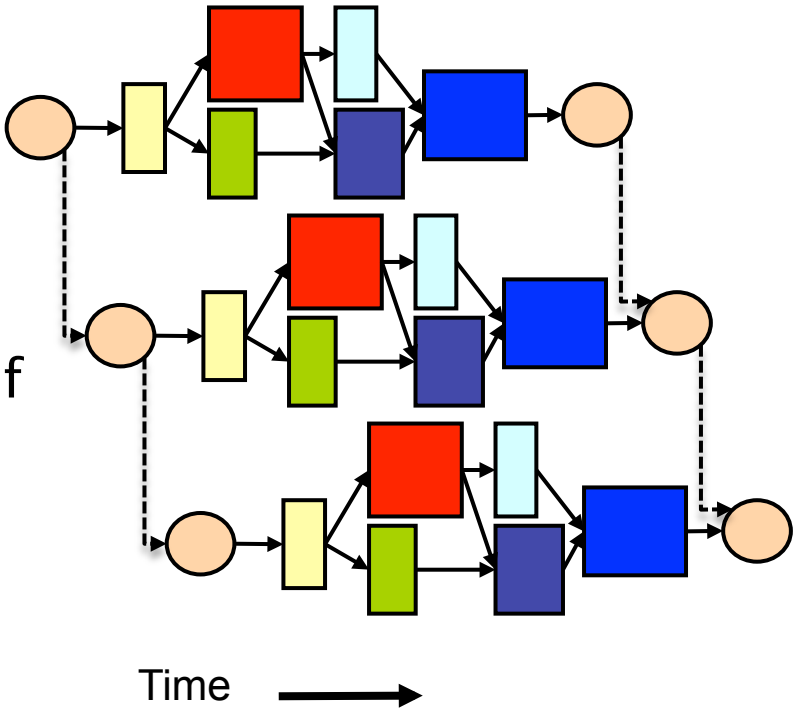
- DAG of Brunel
  - Obtained from the existing code instrumented with 'Auditors'
  - Probably still missing 'hidden or indirect' dependencies (e.g. Tools)
- Can serve to give an idea of potential 'concurrency'
  - Assuming no changes in current reconstruction algorithms



B. Hegner, P. Mato/CERN

Many Concurrent Events

- Need to deal with the tails of sequential processing
- Introducing Pipeline processing
  - Never tried before!
  - Exclusive access to resources or non-reentrant algorithms can be pipelined e.g. file writing
- Need to design or use a powerful and flexible scheduler
- Need to define the concept of an “event context”



B. Hegner, P. Mato/CERN

Concurrency Programming

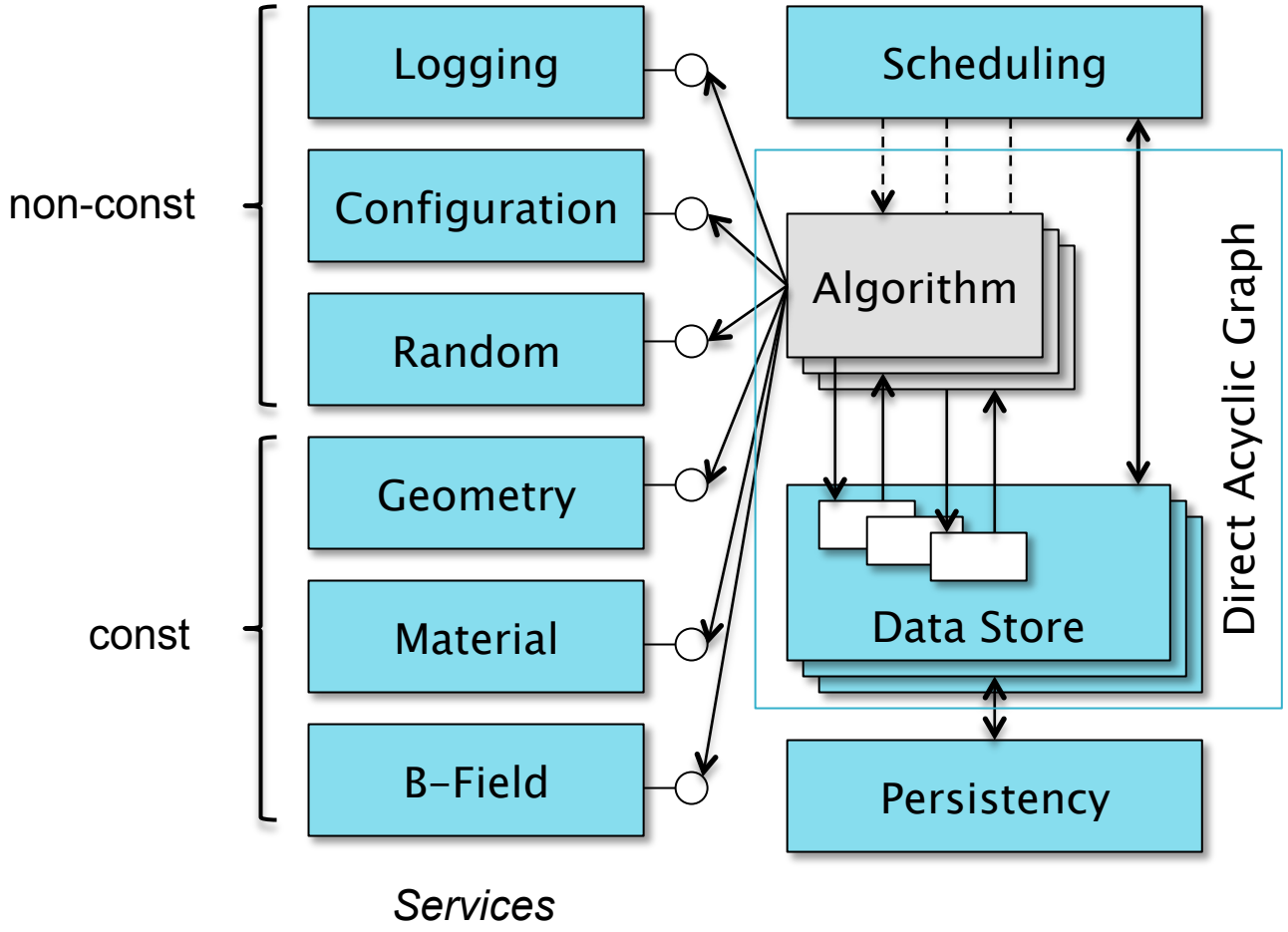
- It is not simple but we are not alone
  - Technologies like the Apple's Grand Central Dispatch (GCD) are designed to help write applications without having to fiddle directly with threads and locking (and getting it terribly wrong)

- New paradigms for concurrency programming
  - Developer needs to factor out the processing in 'chunks' with their dependencies and let the framework (system) to deal with the creation and management of a 'pool' of threads that will take care of the execution of the 'chunks'
  - Tries to eliminates lock-based code and makes it more efficient

Framework as a Set of Services

- Better than a “new” complete and self-contained framework, LHC experiments would like to see a set of functional components from where to pick and choose what to incorporate into their frameworks
  - Experiments have a huge investment in ‘algorithmic’ code and configuration based of a specific framework
- Complete solution should be provided for new experiments
  - The previous constraint does not apply to new experiments
  - The timing is less critical for them

Framework Services



(*) Any resemblance to Gaudi is pure coincidence

Single "Memory Model" is Essential

- Algorithm scheduling will be driven by the semantics of the memory model
  - Knowing what data items an Algorithm "consumes" (reads) and "produces" (modifies, creates) determines when it can be scheduled without conflicts
- Thorough design of shared "Services"
  - Ensure state integrity (e.g. caches)
  - Avoid case-by-case ad-hoc solutions
- Products like ROOT and Geant4 will need to be accommodated to the same memory model

Development of Key Services

- "Concurrent White Board" (multi-event data store)
  - Data declaration (in, out, update)
  - Get synchronized data access (being executed)
  - API for input, output, update and commit
- "Dispatch Service" (scheduler)
  - Management of task queues and threads
  - For example could be based on GCD
- "Logging Service"
  - Ensuring message integrity
  - Sorting by event

Prototyping of Physics Services

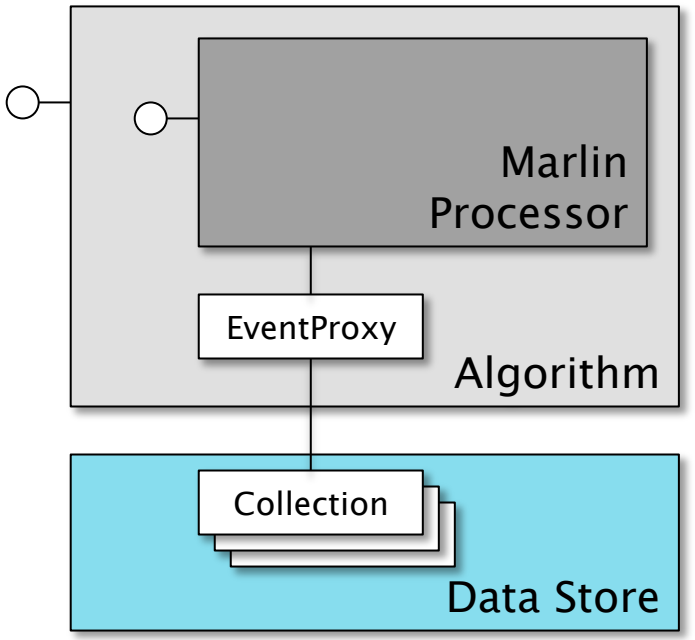
- Modeling them as ‘servers’
  - Genuinely asynchronous
  - Supporting concurrent clients (caching issues)
  - Possible use of new hardware architectures (e.g. GPU, MIC)
- E.g. Random Service
  - Reproducibility in a concurrent environment
- E.g. Magnetic Field Service
  - Given a point, return the best estimate of the B-field
  - It may involve complex interpolations and/or parameterizations
- E.g. Material Service
  - Given two points, return the best estimate of material between them

Concurrency in Geant4

- With an approach like the GDC we could exercise different factorizations
  - Processing each event (set of primary particles) could be the 'task' (same as GeantMT)
- We could also go at the sub-event level
  - Development of Rene's ideas of 'baskets' of particles organized by particle type, volume shape, etc.
  - Would need to develop an efficient summing ('reduce') of the results
  - Would require to study the reproducibility of results (random number sequence)

Re-using Algorithmic Code

- It is essential to be able to re-use existing algorithmic code
- We need to explore whether existing modules/ processors/algorithms could be wrapped and interfaced to the new services
  - Performance would not be great but could be used to evaluate the real benefits for concurrency
  - Adiabatic adaption
- Obviously the issues of re-entrance and thread safety remains



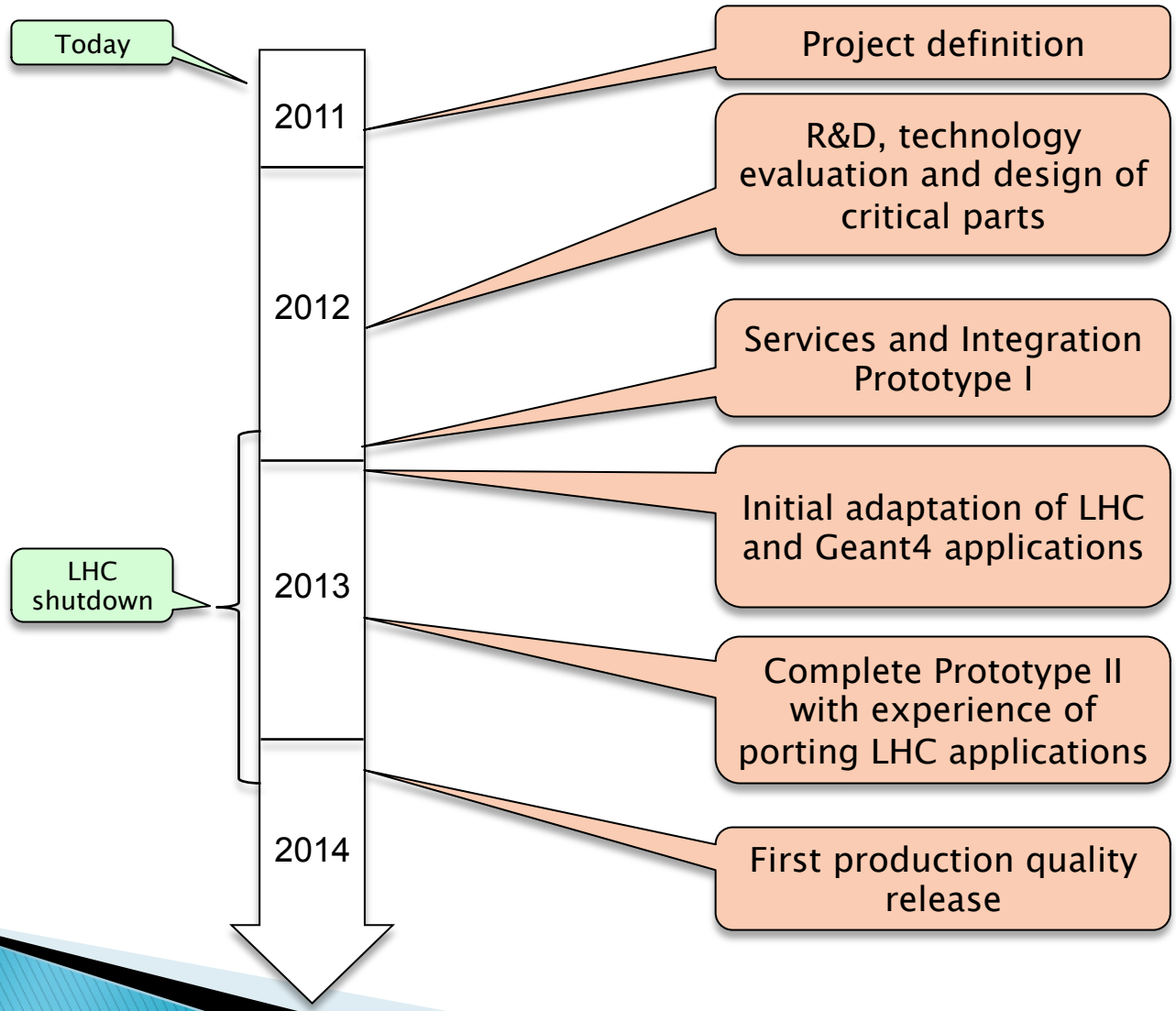
Project

- Collaboration of framework developers of CERN, FNAL, LBL, DESY and possible other Labs
  - Start with small number of people (at the beginning)
  - Open to people willing to collaborate
  - Strong collaboration with ATLAS and CMS (and others)
    - E.g. Instrumentation of existing applications to provide requirements
  - Strong collaboration with Geant4 team
- Quick delivery of running prototypes (I and II)
  - First prototype in 12 months :-)
- Agile project management with 'short' cycles
  - Weekly meetings to review progress and update plans

R&D Activities

- Investigate current LHC applications to gather requirements
  - Dependencies, data access patterns, opportunities for concurrency
- Investigate design and implementations of state-of-the-art concurrency frameworks
  - Scheduling (static, dynamic, adaptive), memory model, I/O
- Prototype framework elements
  - Identify 'exemplar' algorithms to be parallelized
  - Data structures and memory allocation strategies
  - New languages (C++11, Go, pypy, …)
  - and libraries (OpenCL, CnC, STM, …)
- The idea would be to organize these R&D activities in short cycles
  - Coordinating the interested people to cover all aspects
  - Coming with conclusions (yes/no) within few months

Straw man Project Timeline



B. Hegner, P. Mato/CERN

Summary

- Presented initial ideas for the development of a set of generic data processing framework services with concurrency to exploit new CPU/GPU architectures
- LHC experiments should be the main players providing specific requirements, participating into the project development and taking advantage of the new framework
  - Would imply some re-engineering of parts of the experiment applications
- Need a R&D program to evaluate existing technologies and development of partial prototypes of critical parts
- Some initial ideas for the project definition being outlined