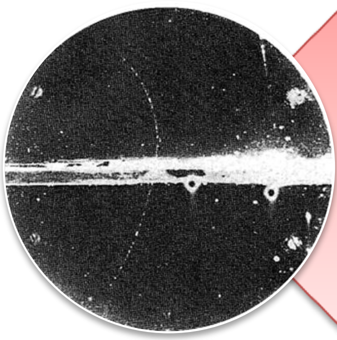
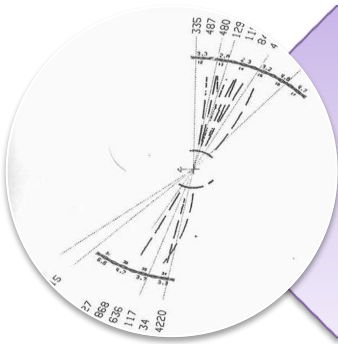
Computing in High Energy Physics



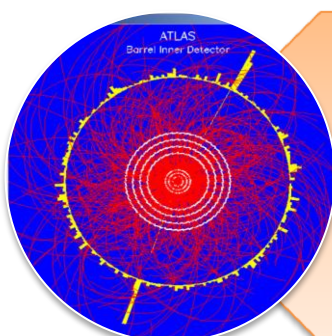
A discovery in 1930-ies

- ~2 scientists in 1country
- Pen and paper



A discovery in 1970-ies

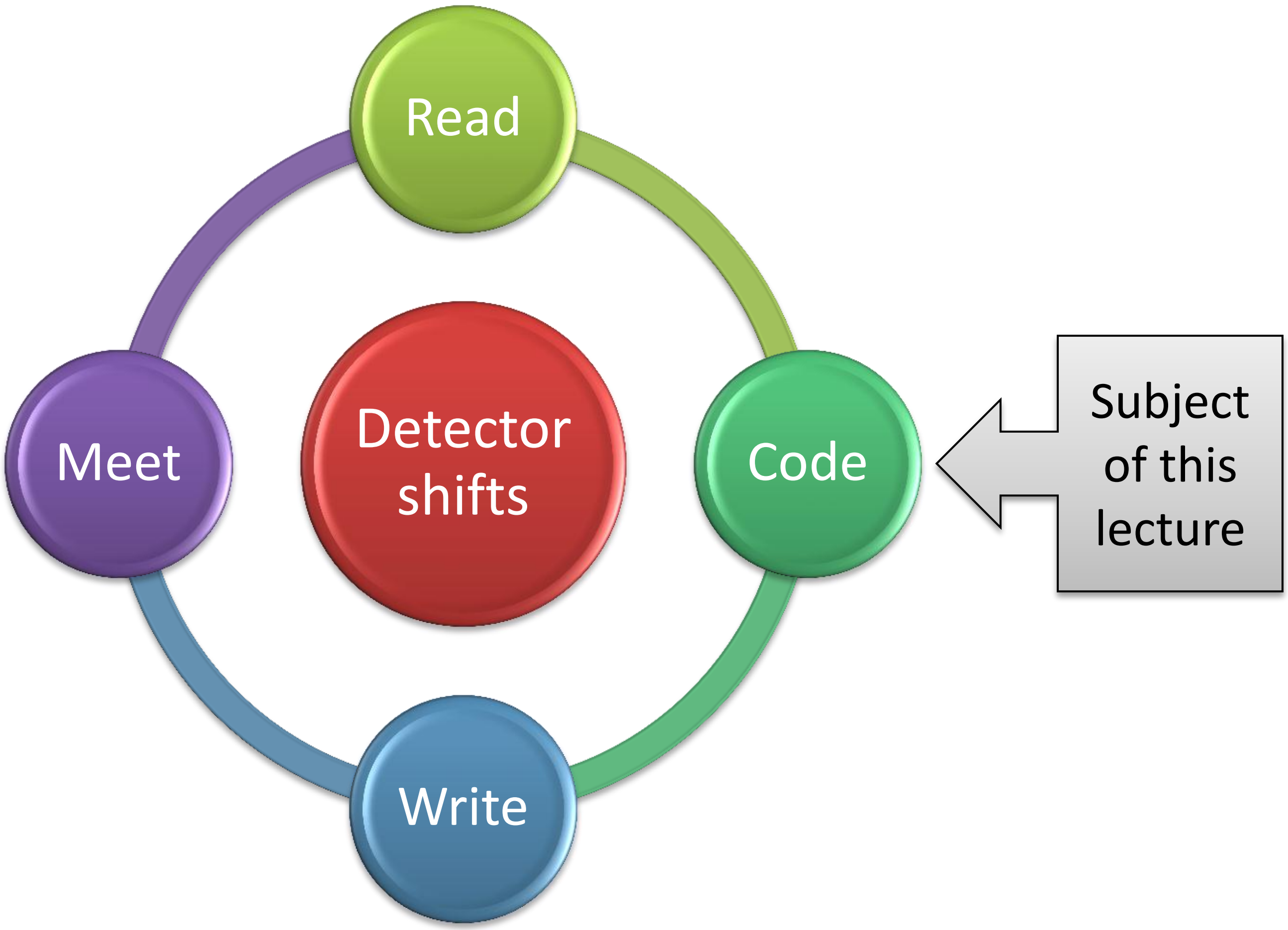
- ~200 scientists in ~10 countries
- Mainframes, supercomputers



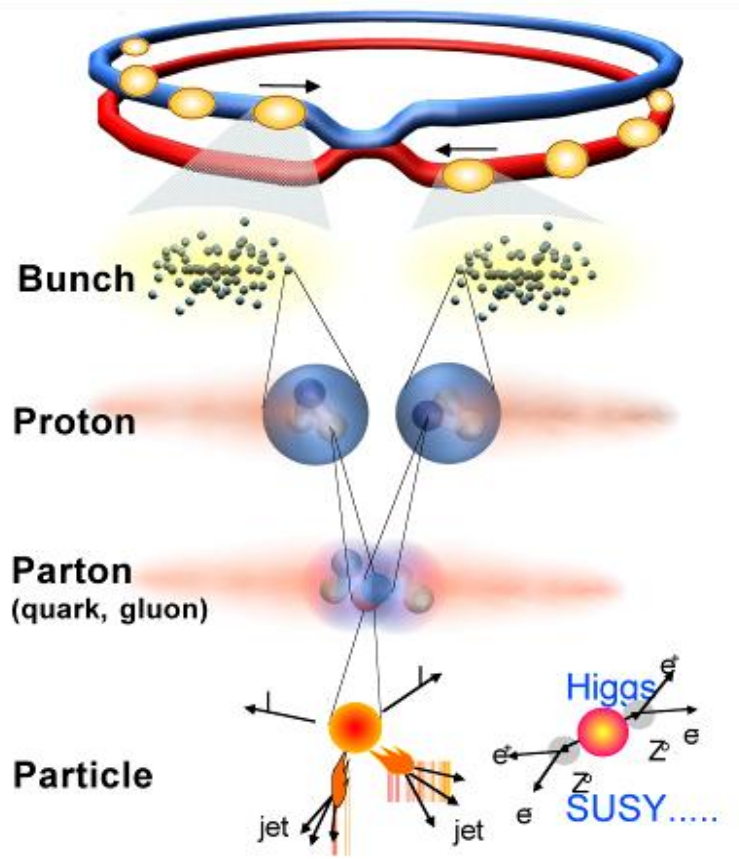
A discovery tomorrow

- ~2000 scientists in ~100 countries
- *Computing and data grids*

A typical LHC physicist's workflow today



Collisions at the LHC



Proton-Proton	2835 bunch/beam
Protons/bunch	10^{11}
Beam energy	7 TeV (7×10^{12} eV)
Luminosity	10^{34} cm^{-2}s^{-1}
Crossing rate	40 MHz
Collisions rate ≈	10^{7} - 10^{9} Hz

New physics rate ≈ .00001 Hz

Event selection:
1 in 10,000,000,000,000

Modern HEP data processing workflow

Event generation (*Monte Carlo*)

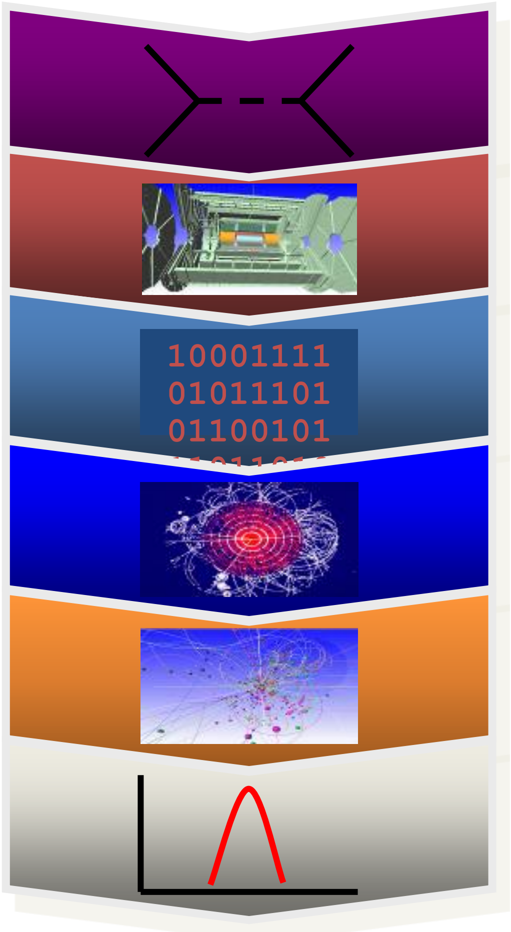
Detector or simulation

Hit digitization

Reconstruction

Analysis data preparation

Analysis, results



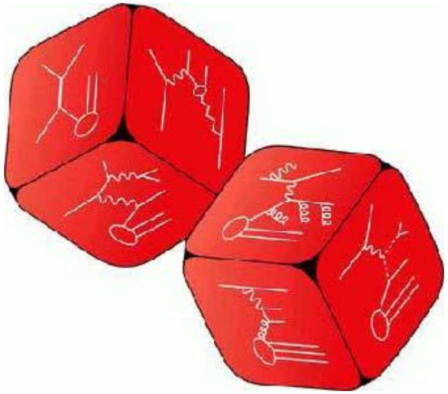
Event generators

- Also known as ***Monte Carlo***
- Are complex software programs that implement our current knowledge of physics
  - Based on the Standard Model
  - Hadronisation (soft QCD) is up to different phenomenological models
  - Allow programmatic inclusion of various processes beyond the Standard Model
  - Are not always exact: precise quantum mechanic effects evaluation may take too much computational resources
- There are many Monte Carlo generators on the market
  - Use different phenomenological models
  - Some are tuned to or specialised in different processes
  - None has true predictive power, though some are better than others
    - Need experimental data for tuning, have many free parameters
  - All written by physicists for physicists
  - Top favorite: **Pythia** (made in Lund)

Why is it called Monte Carlo?

- . . . because Einstein was wrong: God does throw dice!
- Quantum mechanics: amplitudes $\Rightarrow$ probabilities
- Anything that possibly can happen, will! (but more or less often)

- In general, “Monte Carlo” term refers to any numerical method that makes use of random numbers in order to simulate probabilistic processes



Slide adapted from Torbjörn Sjöstrand

Why do we need Monte Carlo?

- To design new experiments and plan for new searches
  - Any new theory can be coded and plugged into a Monte Carlo generator
- To identify unexpected experimental signal
  - When Monte Carlo prediction does not correspond to experimental data, it <u>may</u> mean we see an unexplained phenomenon (or there is a bug in the program)
- To correct for detector inefficiencies

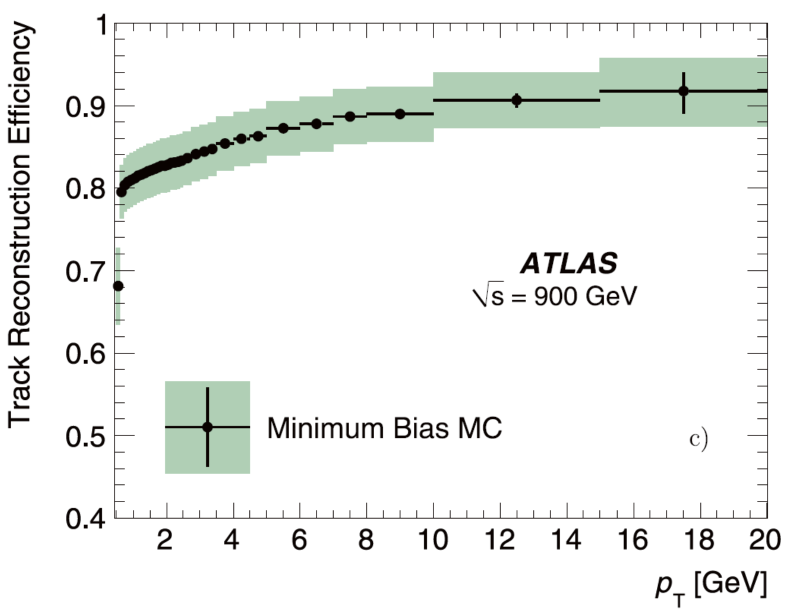


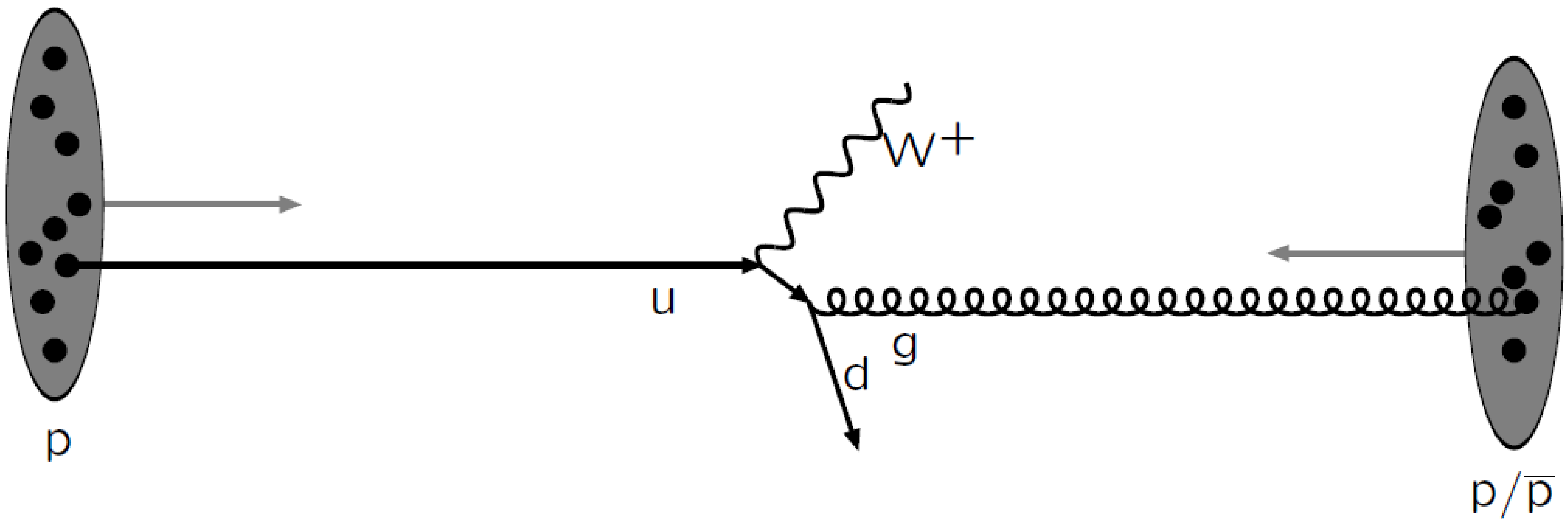
Figure from Phys. Lett. B 688 (2010) 21–42

Proton-proton collision event

Incoming beams: parton densities

Slide by Torbjörn Sjöstrand

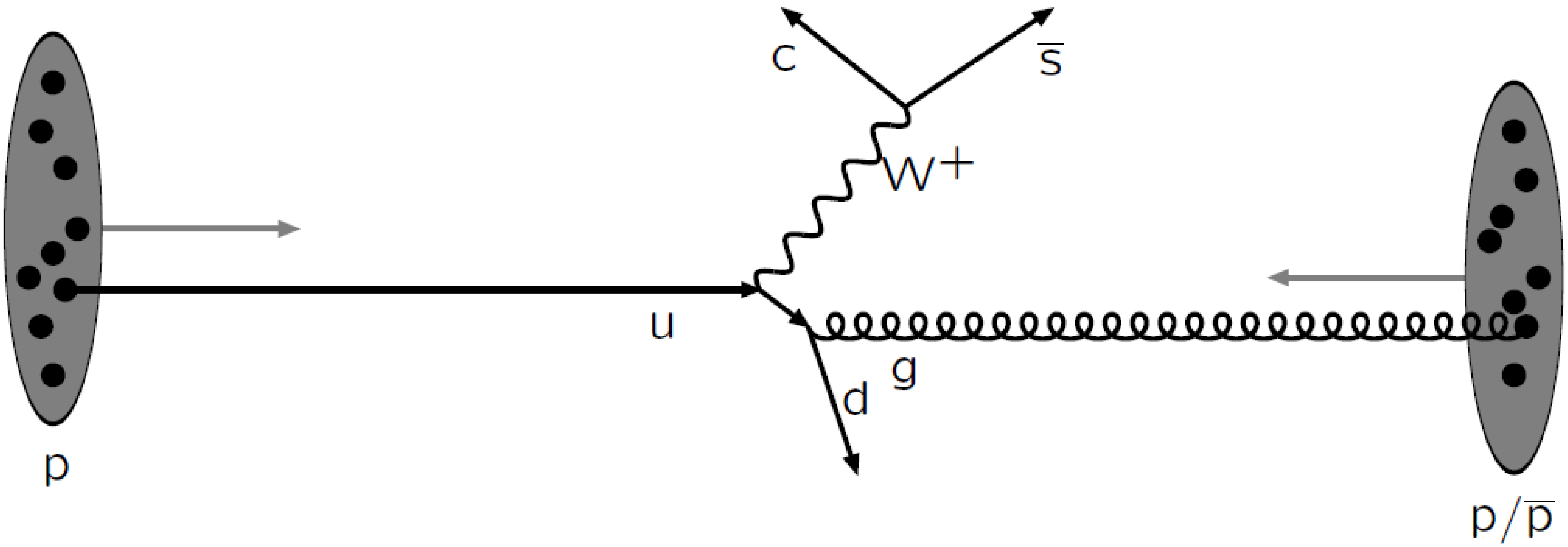
Proton-proton collision event



Hard subprocess: described by matrix elements

Slide by Torbjörn Sjöstrand

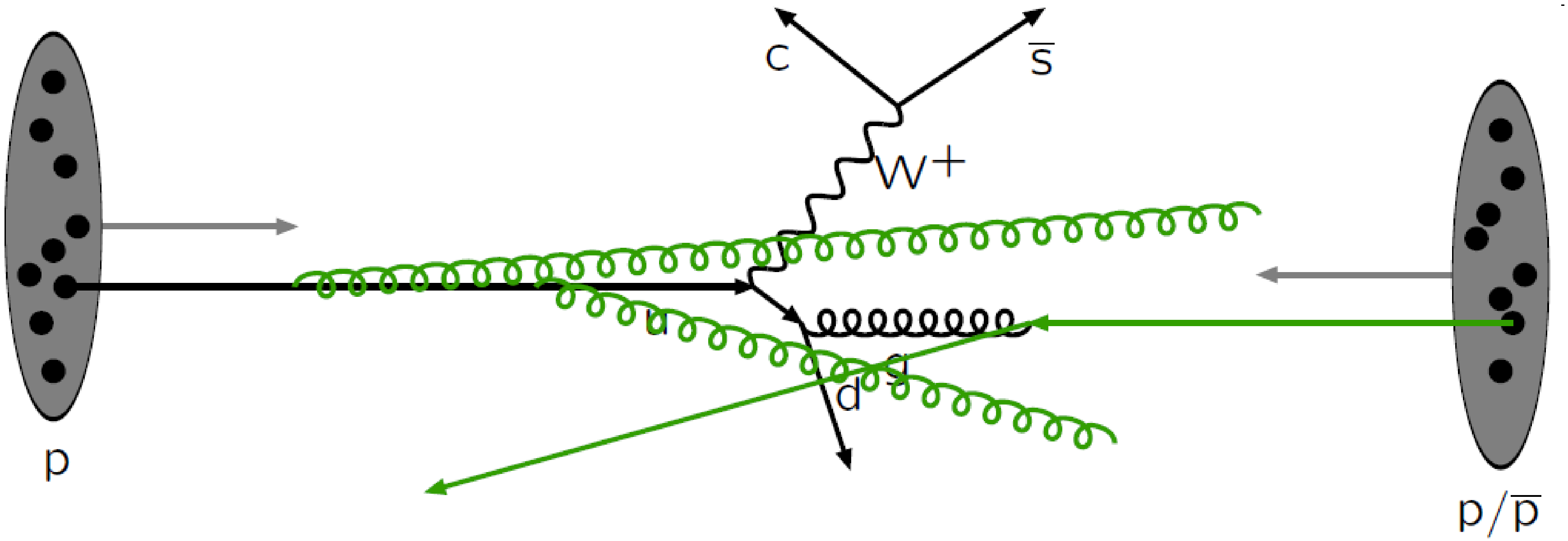
Proton-proton collision event



Resonance decays: correlated with hard subprocess

Slide by Torbjörn Sjöstrand

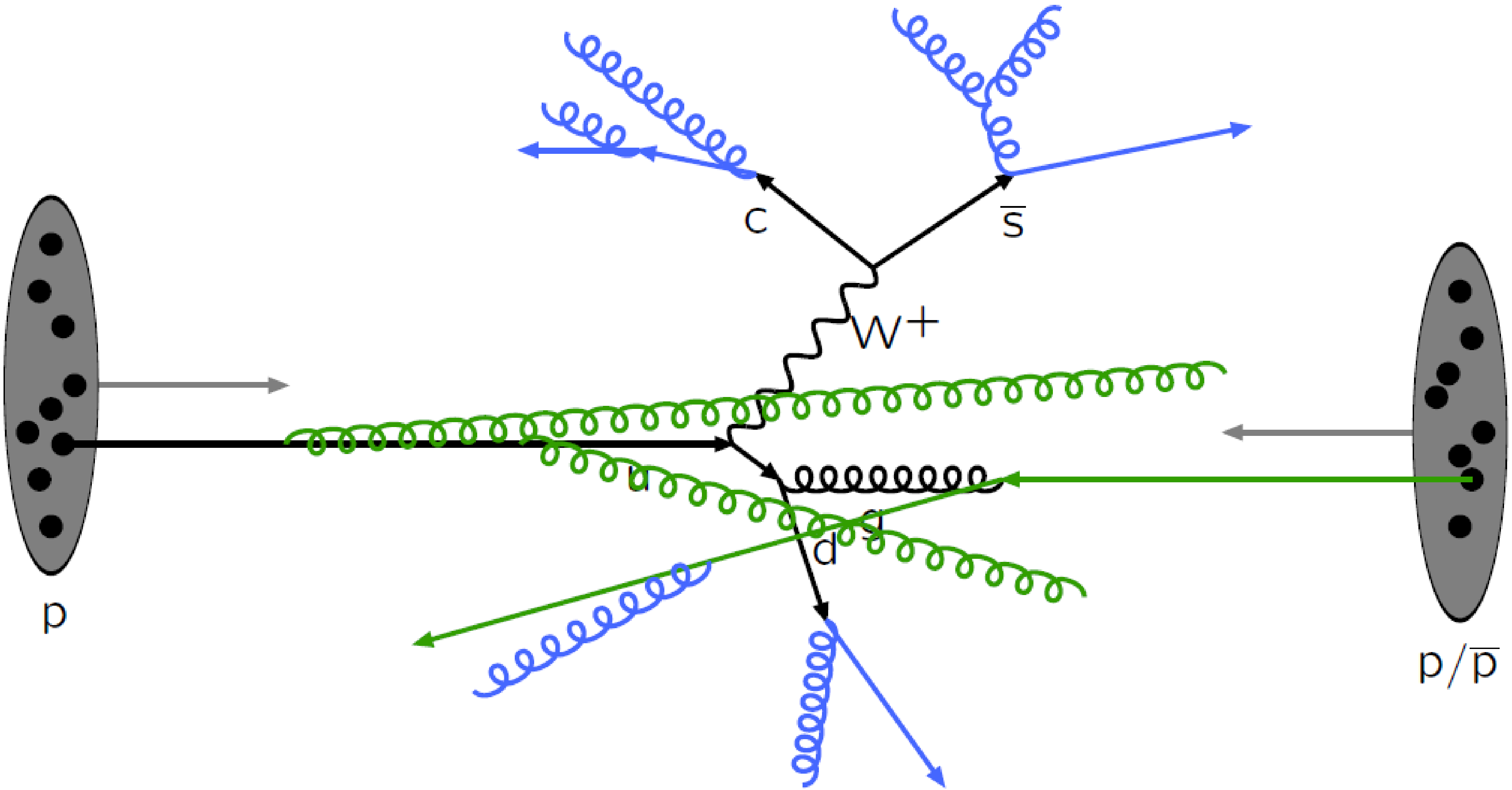
Proton-proton collision event



Initial-state radiation: spacelike parton showers

Slide by Torbjörn Sjöstrand

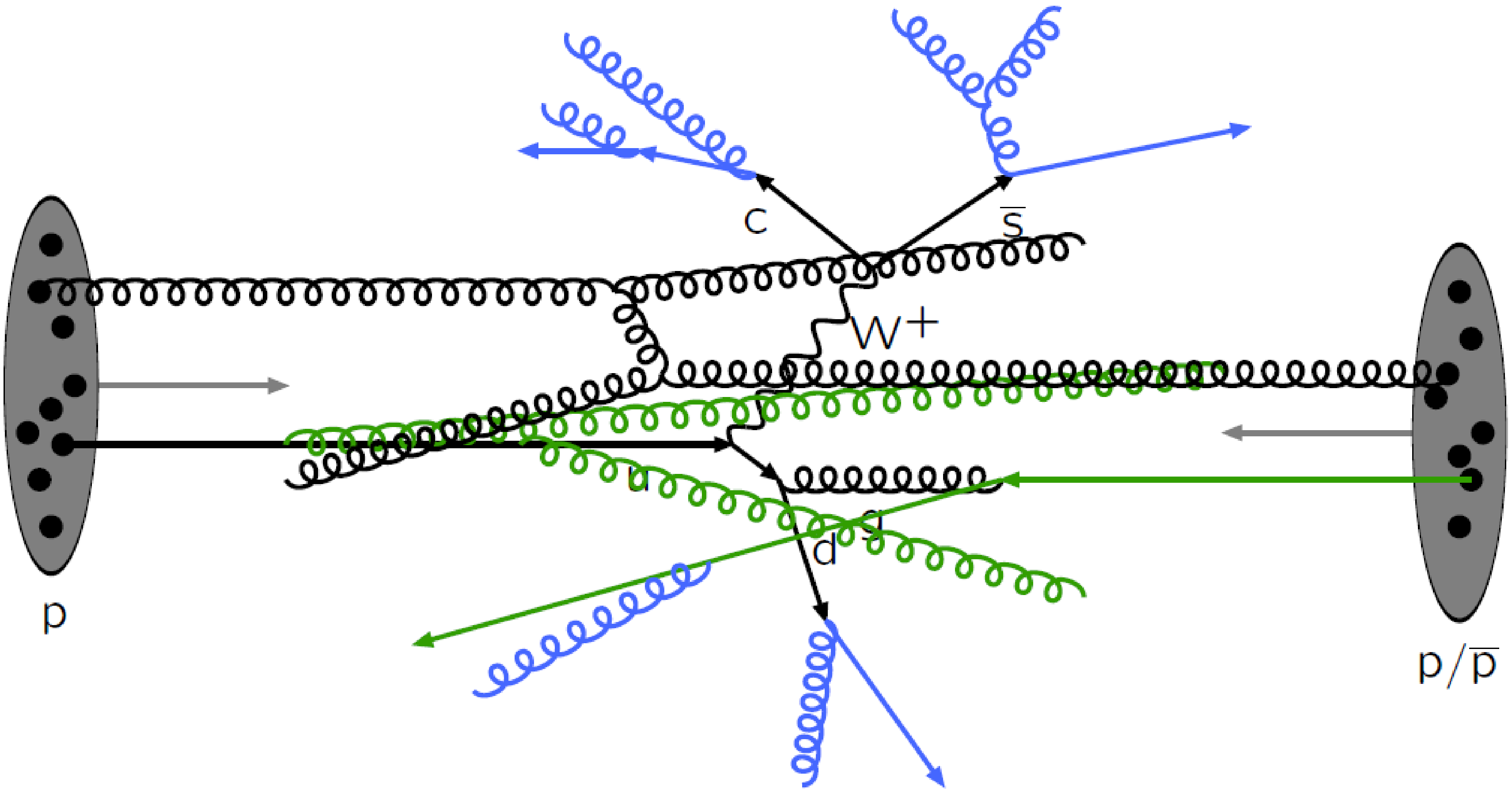
Proton-proton collision event



Final-state radiation: time-like parton showers

Slide by Torbjörn Sjöstrand

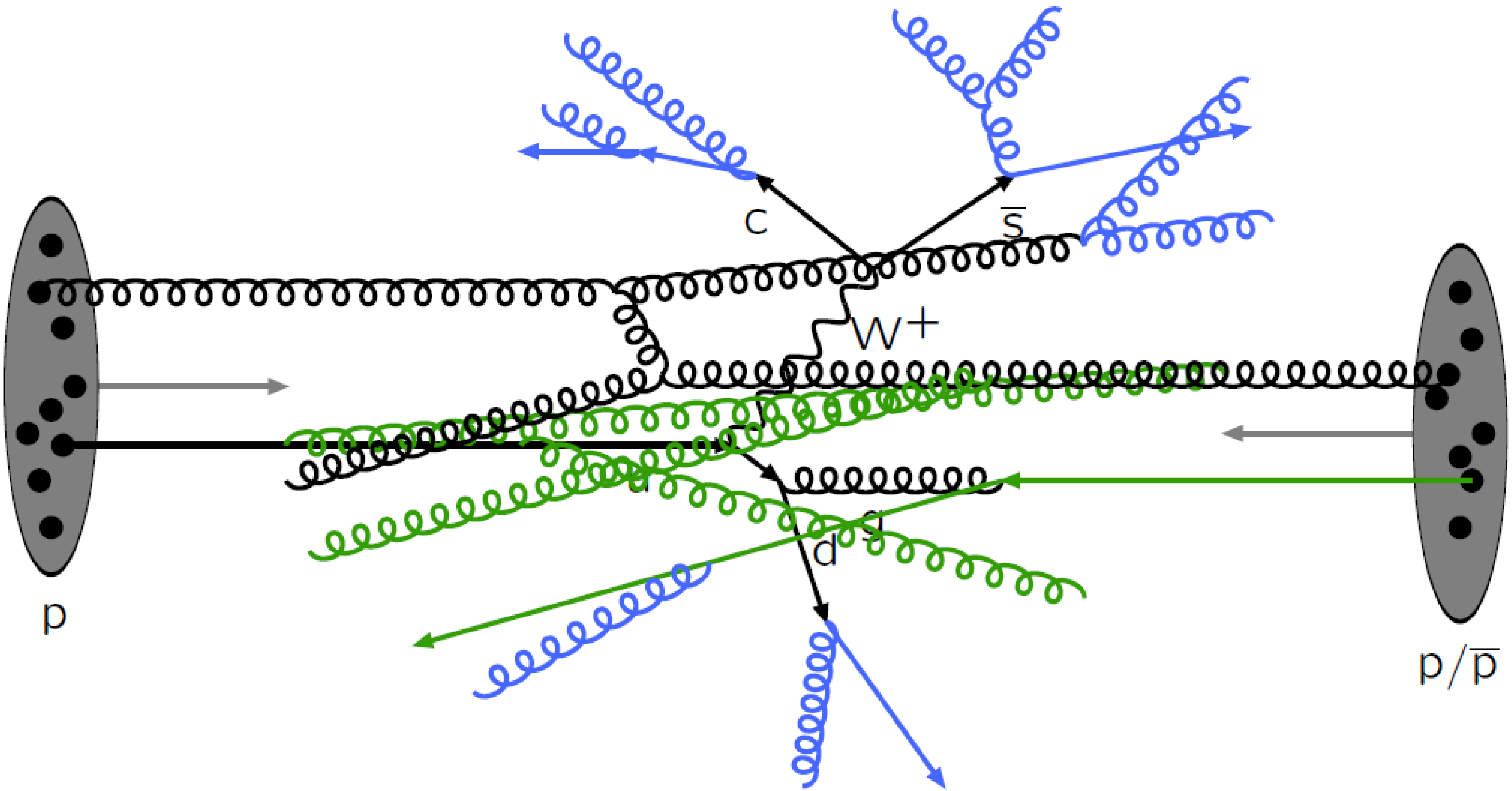
Proton-proton collision event



Multiple parton–parton interactions…

Slide by Torbjörn Sjöstrand

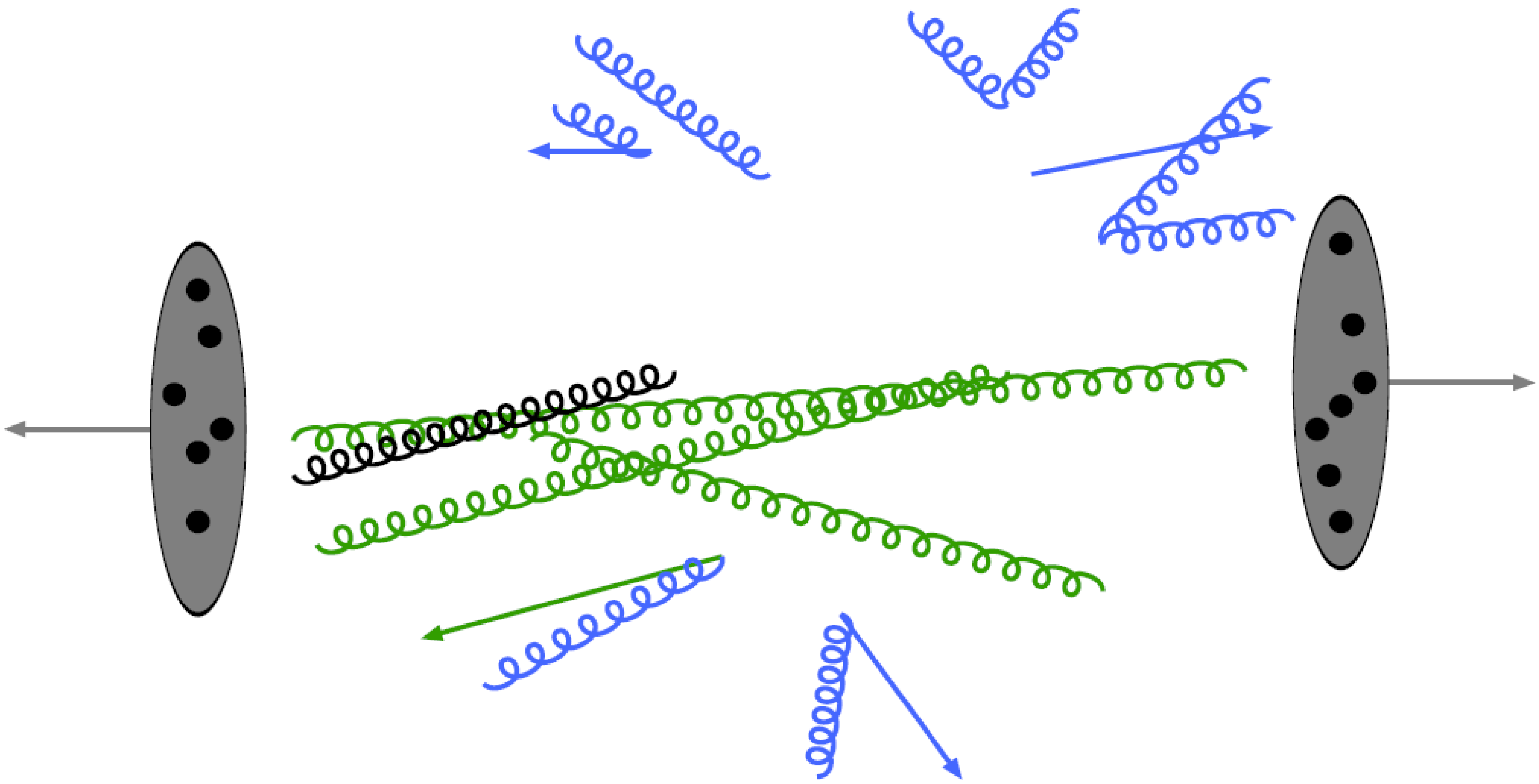
Proton-proton collision event



. . . with its initial- and final-state radiation

Slide by Torbjörn Sjöstrand

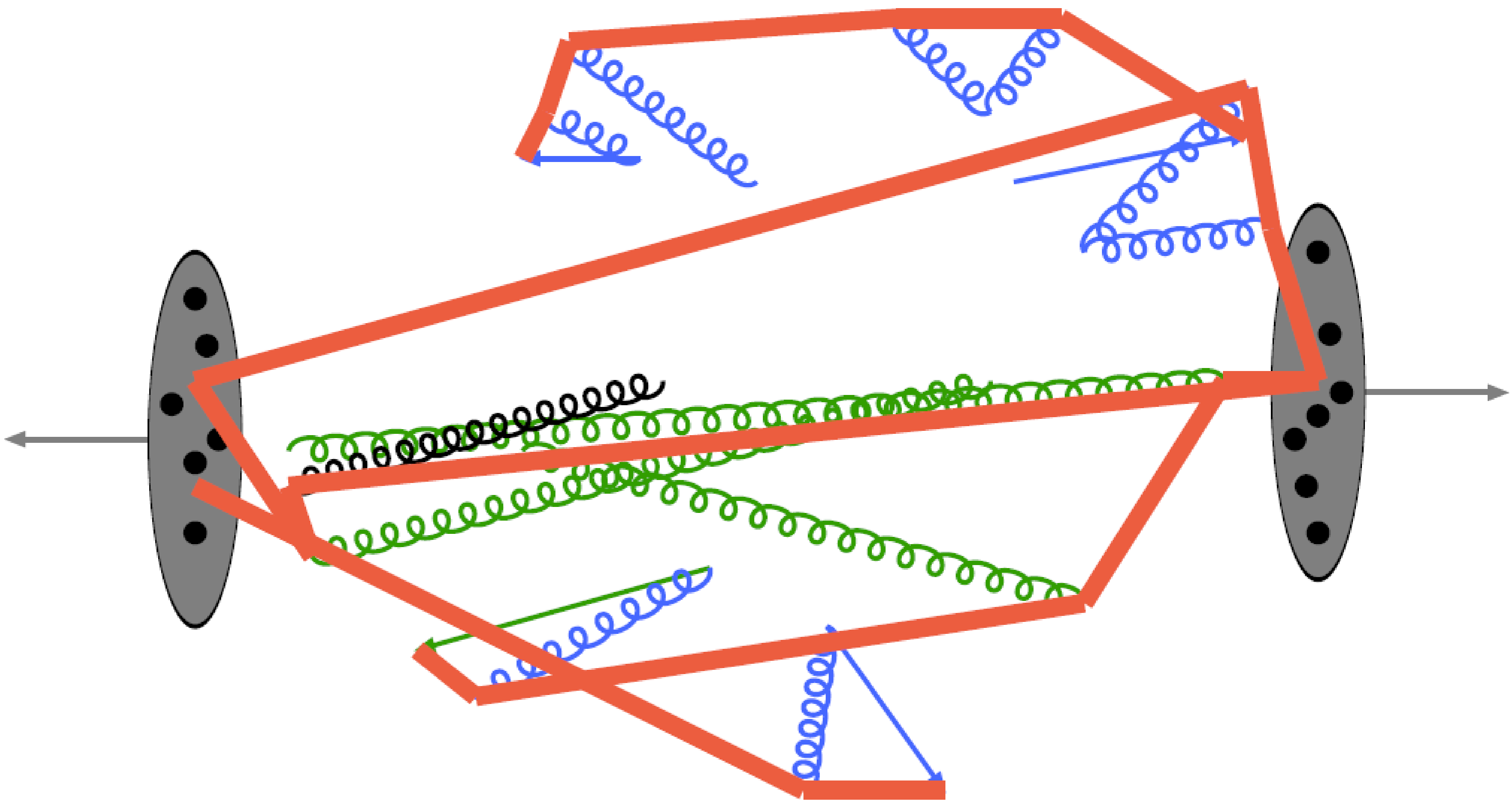
Proton-proton collision event



Beam remnants and other outgoing partons

Slide by Torbjörn Sjöstrand

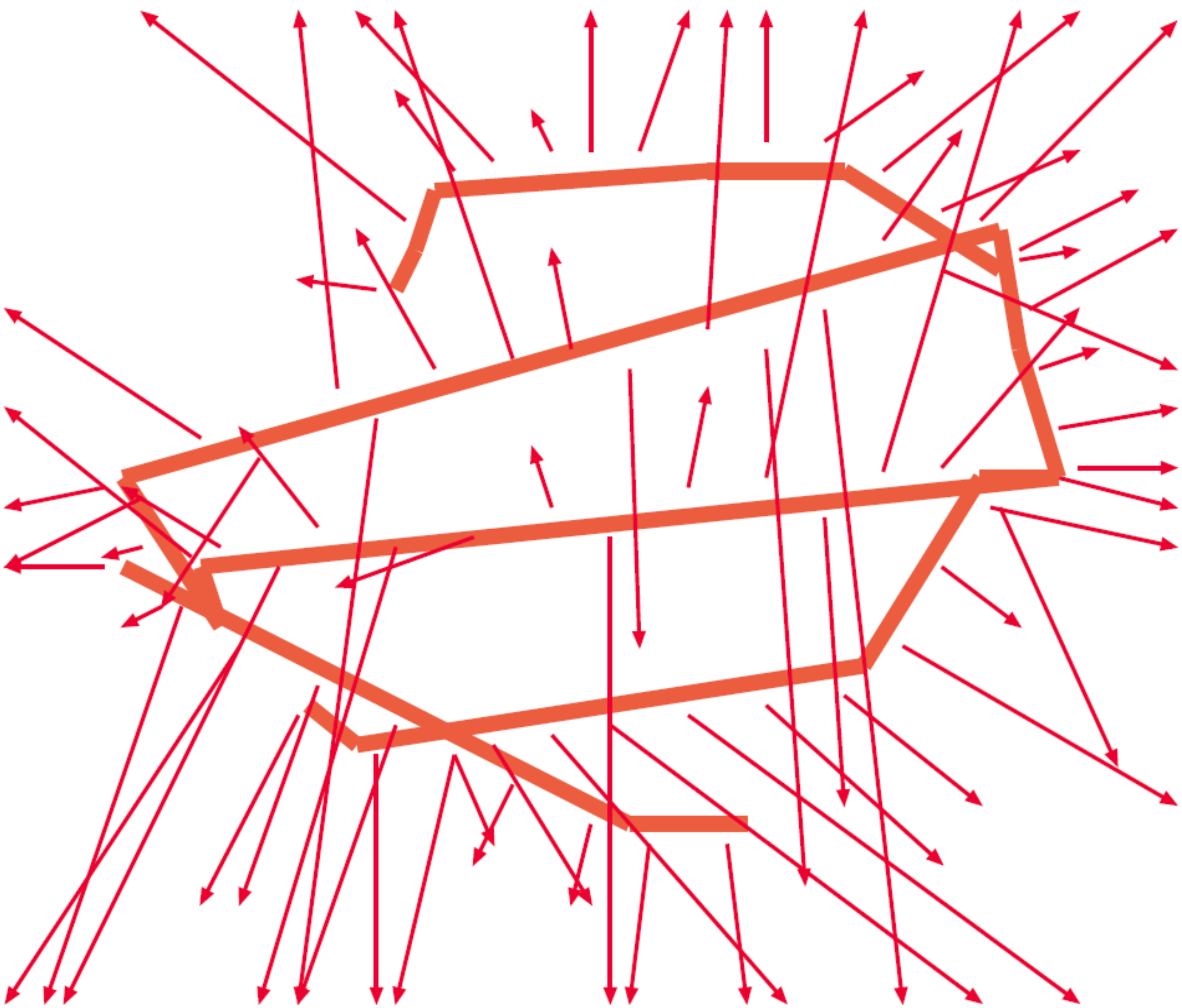
Proton-proton collision event



Everything is connected by colour confinement strings

Slide by Torbjörn Sjöstrand

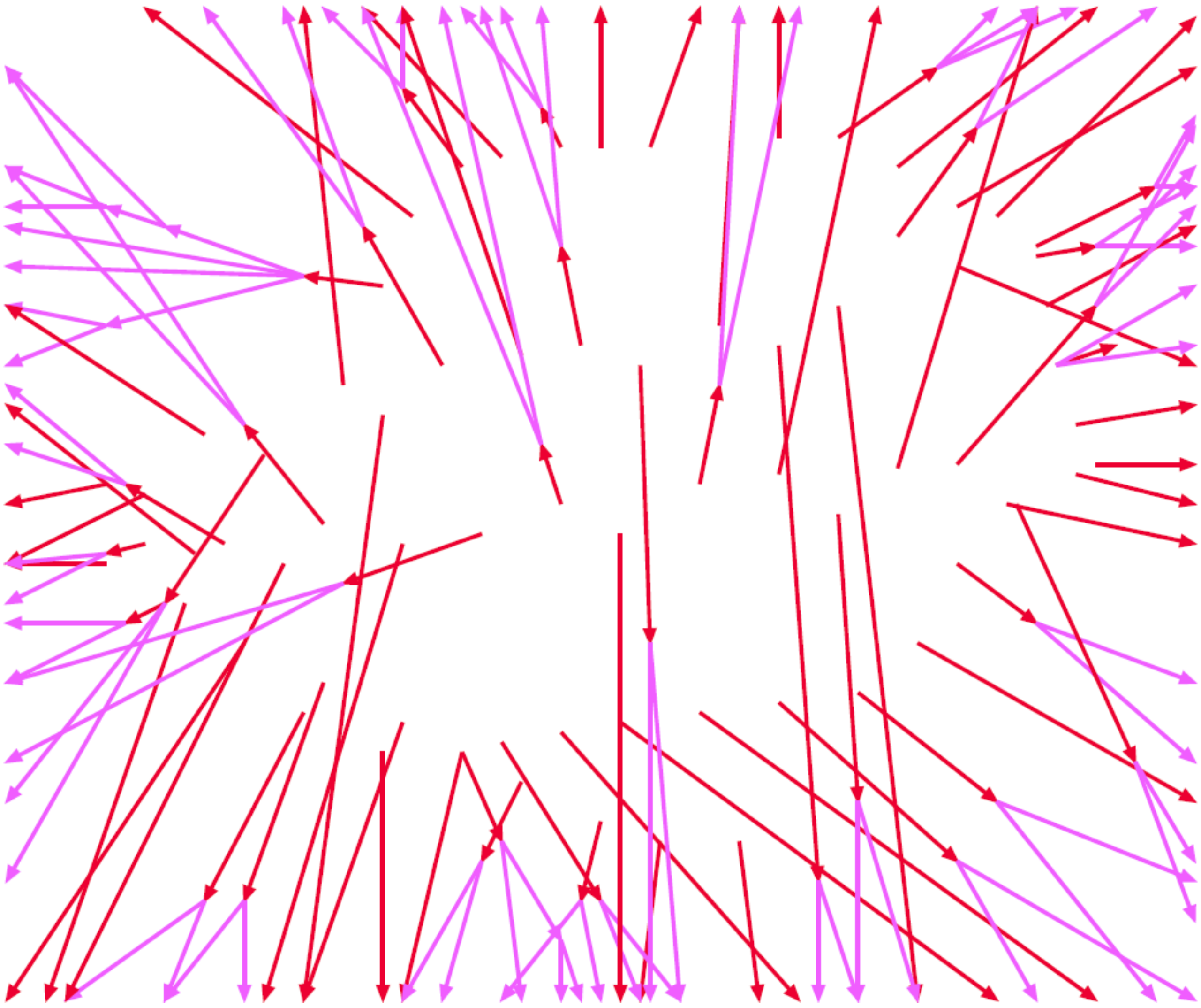
Proton-proton collision event



The strings fragment to produce primary hadrons

Slide by Torbjörn Sjöstrand

Proton-proton collision event



Many hadrons are unstable and decay further

Slide by Torbjörn Sjöstrand

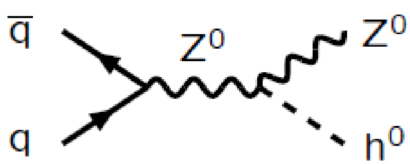
Event physics overview

Matrix elements (ME):

1) Hard subprocess:

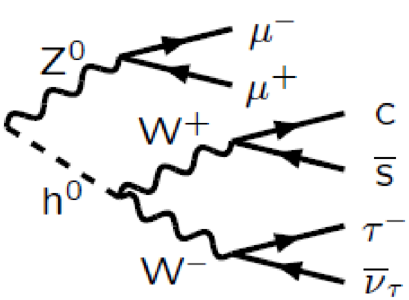
$|\mathcal{M}|^2$, Breit-Wigners,

parton densities.



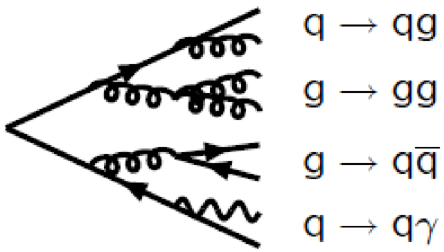
2) Resonance decays:

includes correlations.

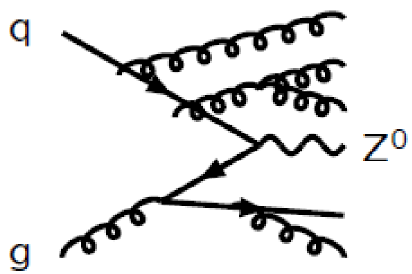


Parton Showers (PS):

3) Final-state parton showers.



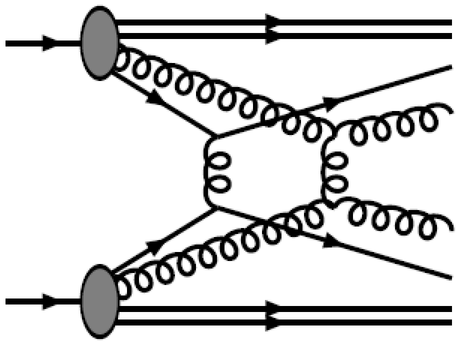
4) Initial-state parton showers.



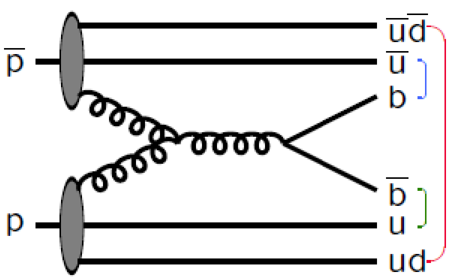
Slide by Torbjörn Sjöstrand

Event physics overview

5) Multiple parton–parton interactions.

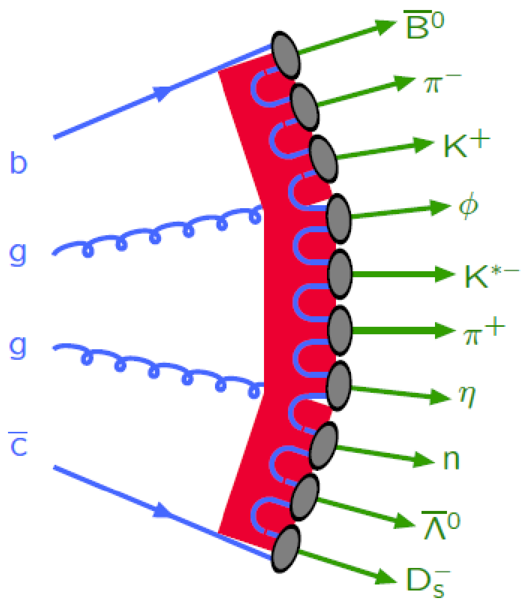


6) Beam remnants, with colour connections.

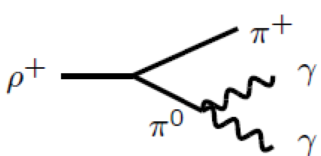


5) + 6) = Underlying Event

7) Hadronization



8) Ordinary decays: hadronic, τ, charm, . . .



Slide by Torbjörn Sjöstrand

Pythia – describes data best

- Pythia was known as the Oracle of Delfi, possessed immense predictive powers (until year 393)
- In 21st century, Pythia is arguably the most successful HEP Monte Carlo generator

- Pythia highlights:
  - Hard processes: Standard Model and beyond, resonance decays etc
  - Showers: initial- and final-state radiation, transverse momentum ordered
  - Underlying event: multiple interactions, colour-connected beam remnants
  - Hadronisation: Lund model, particle decays, Bose-Einstein effects
  - Various auxiliary utilities

Simplest code using Pythia 8 (C++)

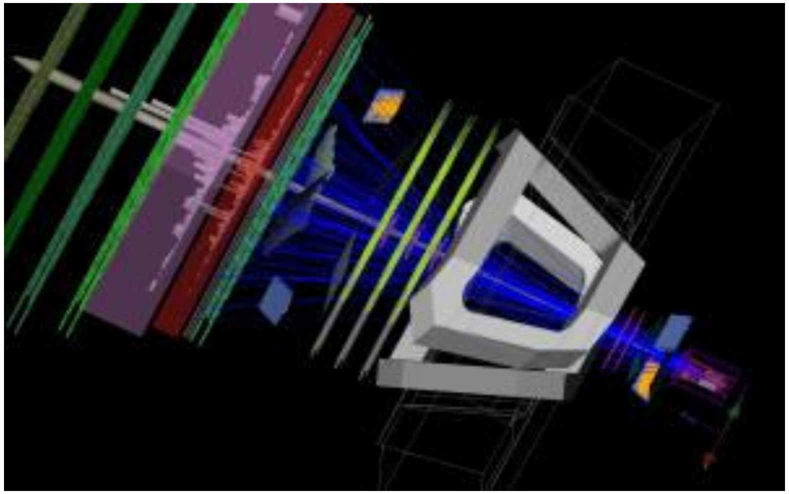
```
// File: main01.cc. The charged multiplicity distribution at the LHC.
#include "Pythia.h"
using namespace Pythia8;
int main() {
  // Generator. Process selection. LHC initialization. Histogram.
  Pythia pythia;
  pythia.readString("HardQCD:all = on");
  pythia.readString("PhaseSpace:pTHatMin = 20.");
  pythia.init( 2212, 2212, 14000.);
  Hist mult("charged multiplicity", 100, -0.5, 799.5);
  // Begin event loop. Generate event. Skip if error. List first one.
  for (int iEvent = 0; iEvent < 100; ++iEvent) {
    if (!pythia.next()) continue;
    if (iEvent < 1) {pythia.info.list(); pythia.event.list();}
    // Find number of all final charged particles and fill histogram.
    int nCharged = 0;
    for (int i = 0; i < pythia.event.size(); ++i)
      if (pythia.event[i].isFinal() && pythia.event[i].isCharged())
        ++nCharged;
    mult.fill( nCharged );
  // End of event loop. Statistics. Histogram. Done.
  }
  pythia.statistics();
  cout << mult;
  return 0;
}
```

Slide by Torbjörn Sjöstrand

Detector effects

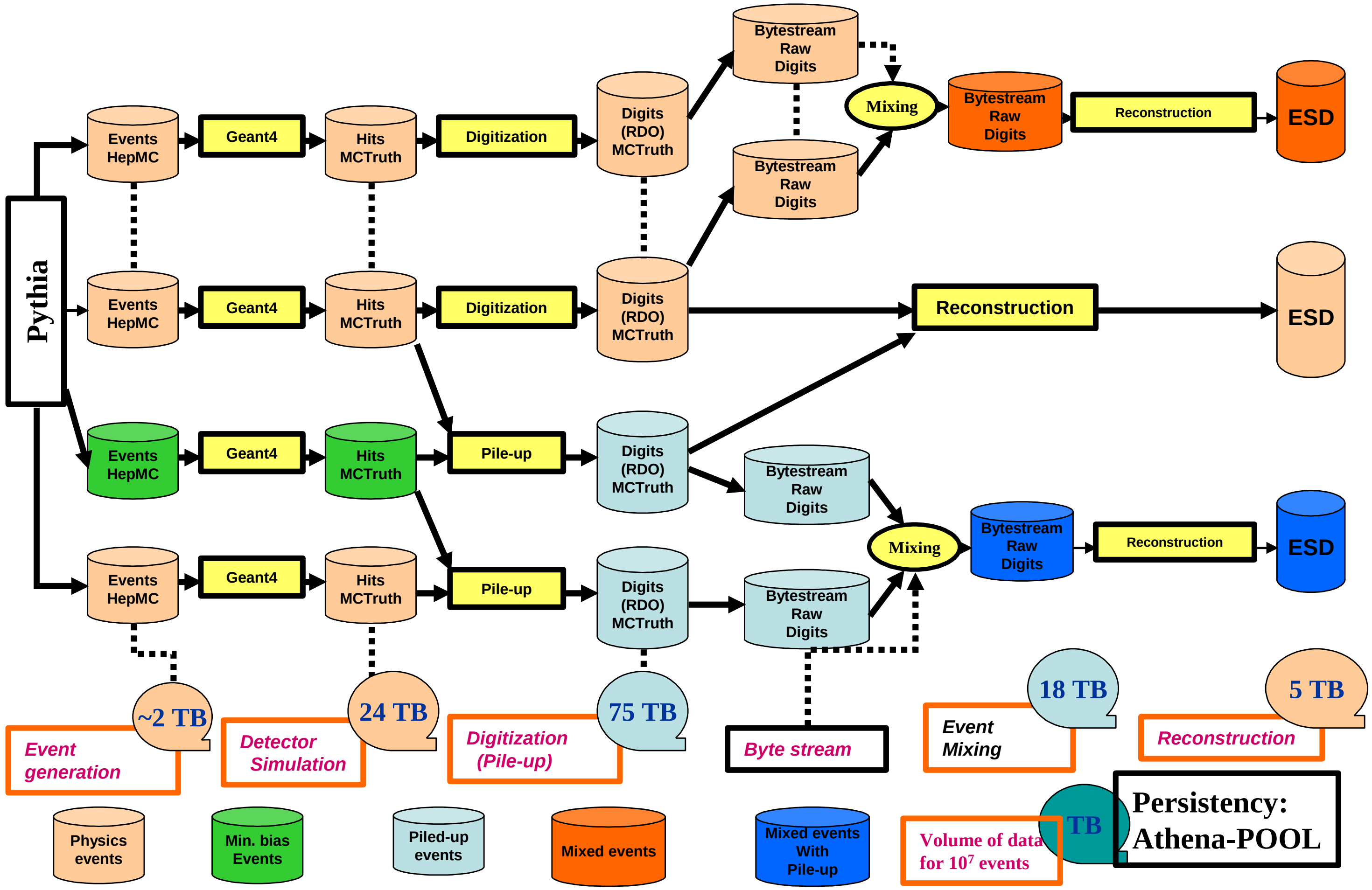
- Accelerators strive to emulate nature
- Monte Carlo strives to reproduce accelerator collision events
- But our detectors are never perfect!

- Every detector can be simulated by software
  - Making use of knowledge of particle interactions with matter
  - Needs precise knowledge of detector geometry, magnetic field, gas status etc
  - Although largely deterministic, has some probabilistic effects as well



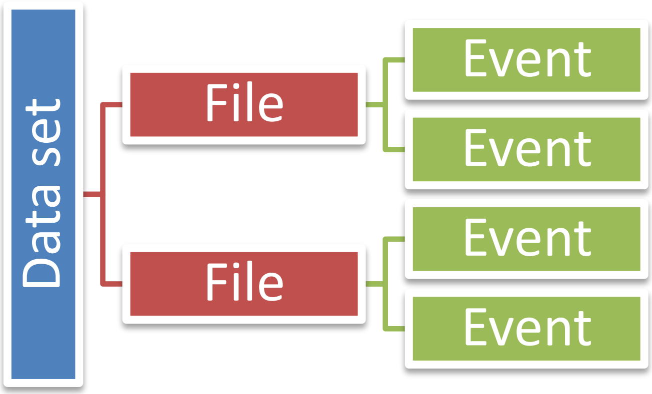
- Most complete detector simulation software: GEANT (version 4 is the latest)
  - Pythia (or other good Monte Carlo) and GEANT are absolutely necessary to calculate corrections for detector inefficiencies

Monte Carlo data production flow (10 Mevents)

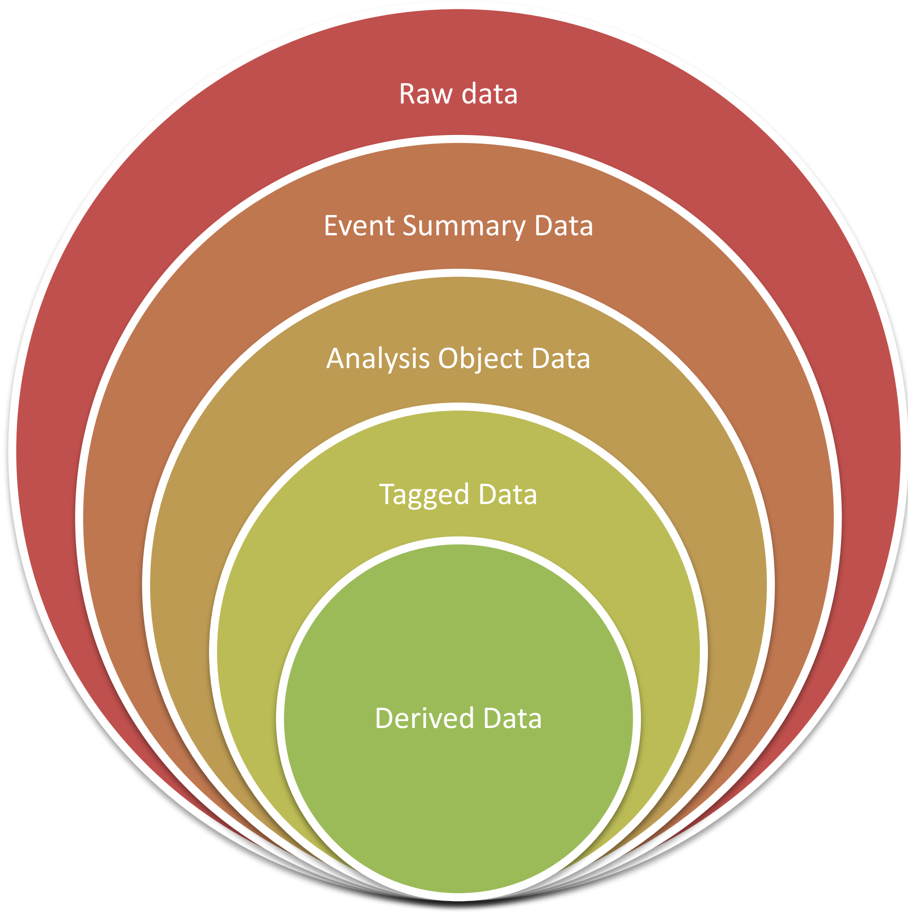


- Very different tasks/algorithms (ATLAS experiment in this example)
- Single “job” lasts from 10 minutes to 1 day
- Most tasks require large amounts of input and produce large output data

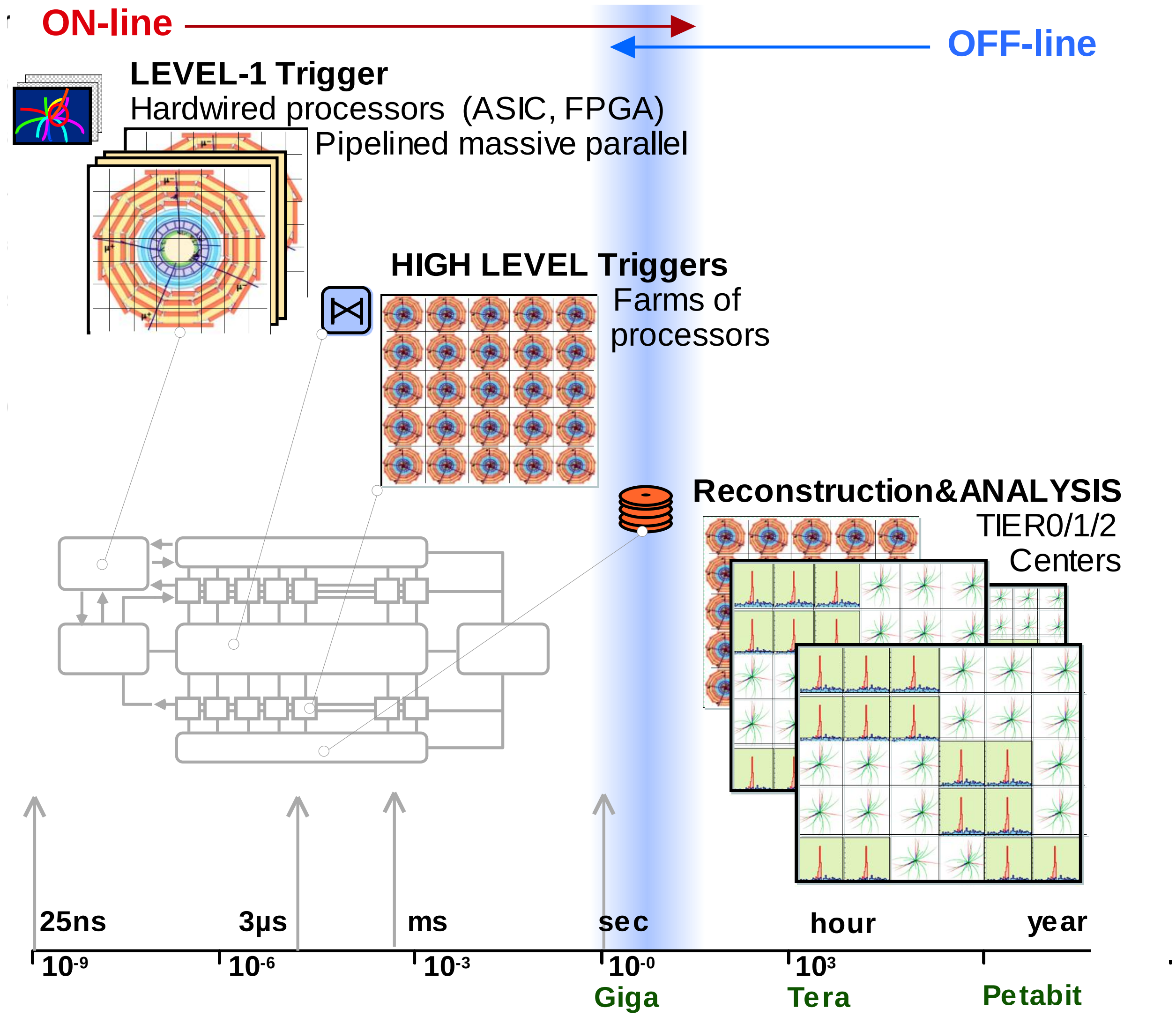
HEP data hierarchies



- Different experiments use different data models
- Since collision events are independent, they can be collected into files almost arbitrarily
  - Typically, a data set corresponds to a specific physics or simulation episode (“run”)
- Data sets are derived from each other: from raw data to analysis objects



Physics selection at LHC - CMS experiment example

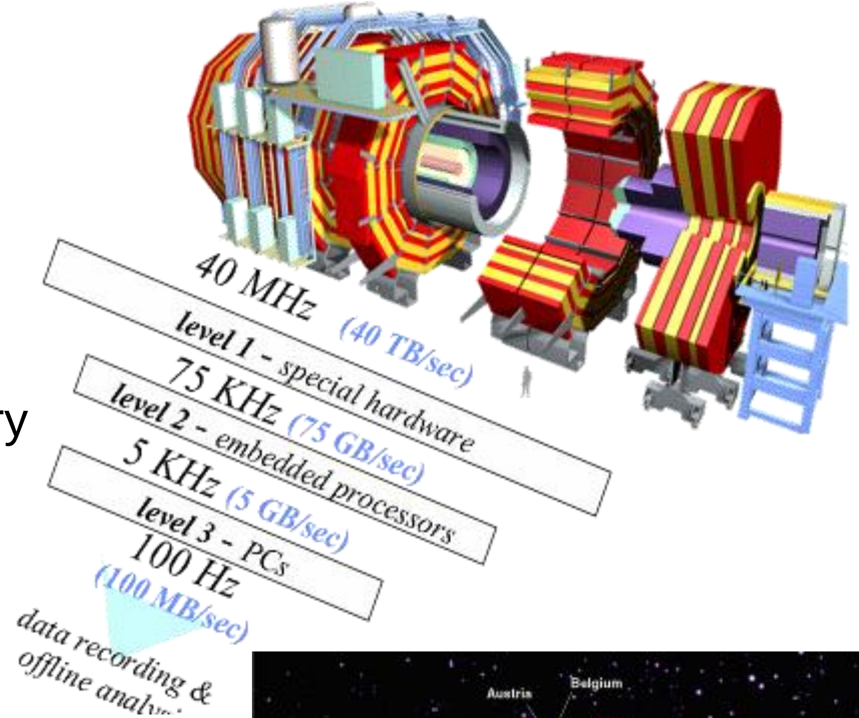


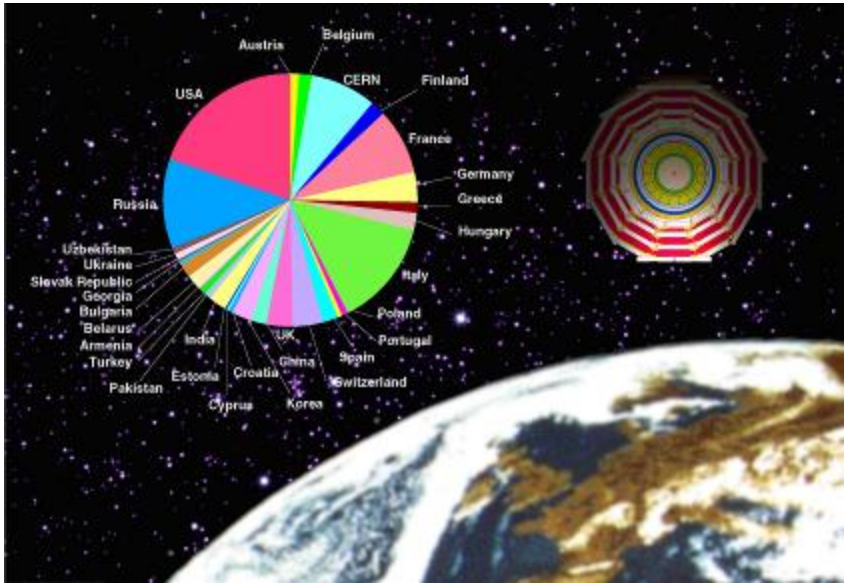
Data analysis frameworks

- HEP experiments are fiercely protective of their data
- Data are stored in proprietary formats and are not available for people outside experiments
  - This restriction is normally lifted after experiment stops operating
- Every experiment develops own set of analysis software
  - Facilitates detector-specific activities (Monte Carlo, GEANT, reconstruction, calibration etc)
  - Implements computing model
  - Handles remote access to date
  - etc
- Typically, all HEP software frameworks have command line interfaces and are developed for Linux
  - HEP community has own Linux flavour: Scientific Linux (RedHat derivative)

HEP computing specifics

- Data-intensive tasks
  - Large datasets, large files
  - Lengthy processing times
  - Large memory consumption
  - High throughput is necessary
- Very distributed resources
  - Distributed computing resources of _modest_ size
  - Produced and processed data are also distributed
  - Issues of coordination, synchronization and authorization are outstanding
- HEP is not unique in its demands, but we are first, we are many, and we ***have*** to solve it





Software for HEP experiments

Massive pieces of software	• Written by very many different authors in different languages (C++, Python, FORTRAN etc) • Dozens of external components
Frequent releases	• Occupy as much as ~10 GB of disk space each release • Releases can come as often as once a month
Difficult to set up outside the lab	• Software is often tuned to the specifics of one lab (e.g. Scientific Linux operating system)
Unfeasible to be maintained locally by small university teams	• Plan A: do everything centrally at the lab • Plan B: use **Grid** to connect large external computing "plants" managed by teams of experts

Origin of the word “Grid”

- Coined by **Ian Foster** and **Carl Kesselman** around 1997
- Refers to computing grids as analogy of power grids
  - Many producers
  - Competing providers
  - Simple for end-users
- Spelled “grid” or “Grid”

Grid is a result of IT progress

Network vs. computer performance:

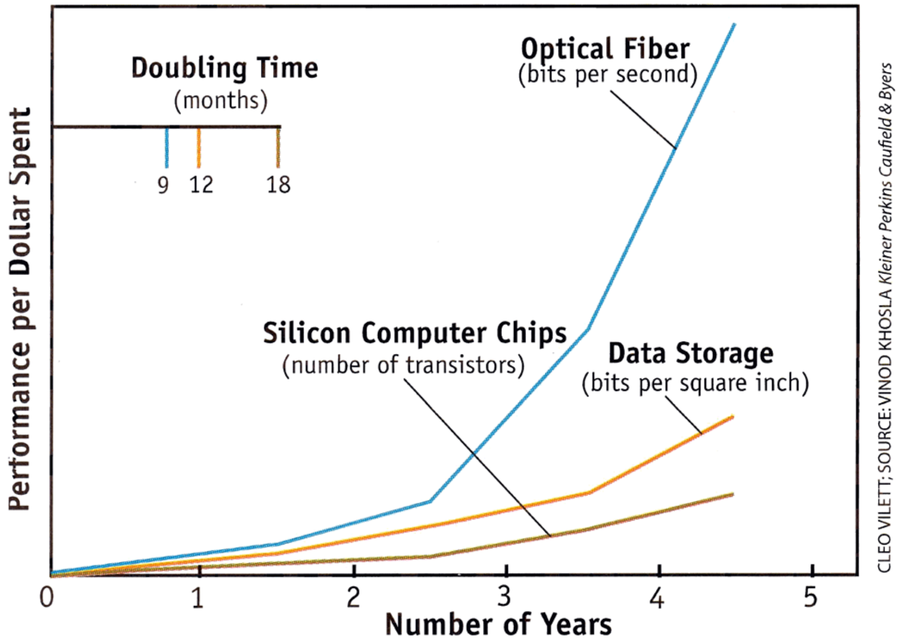
- Computer speed doubles every 18 months
- Network speed doubles every **9** months

1986 to 2000:

- Computers: 500 times faster
- Networks: 340000 times faster

2001 to 2010 (projected):

- Computers: 60 times faster
- Networks: 4000 times faster

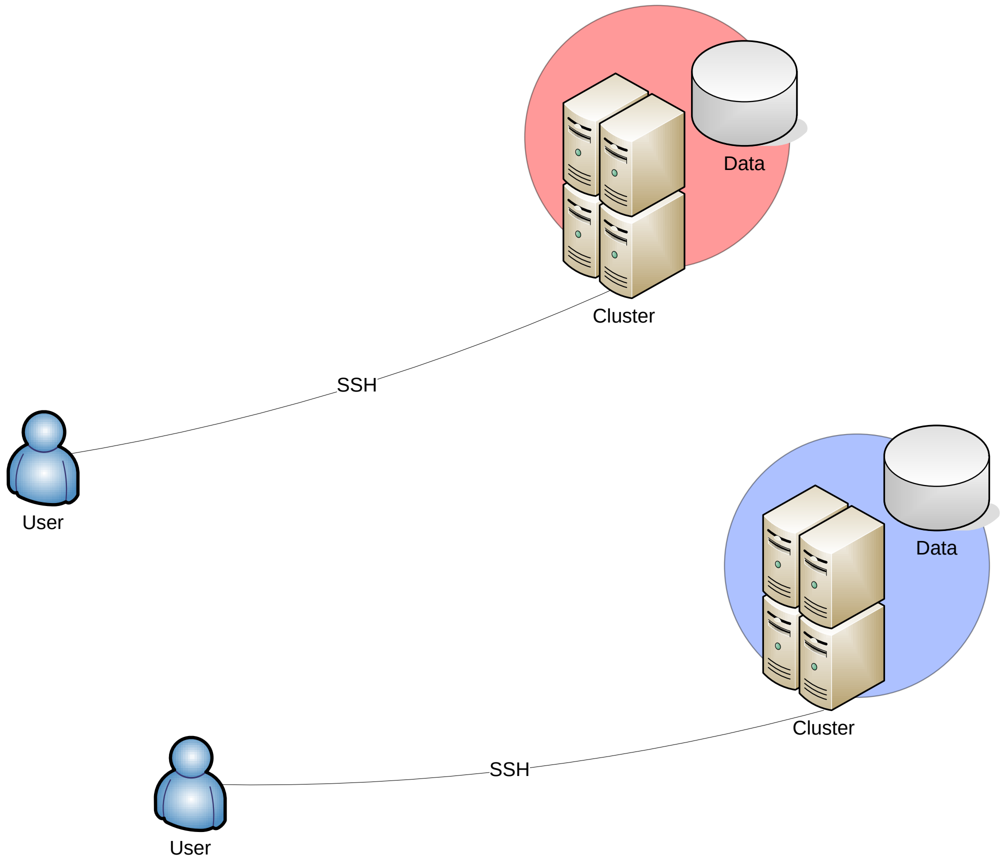


Graph from "The Triumph of the Light", G. Stix, Sci. Am. January 2001

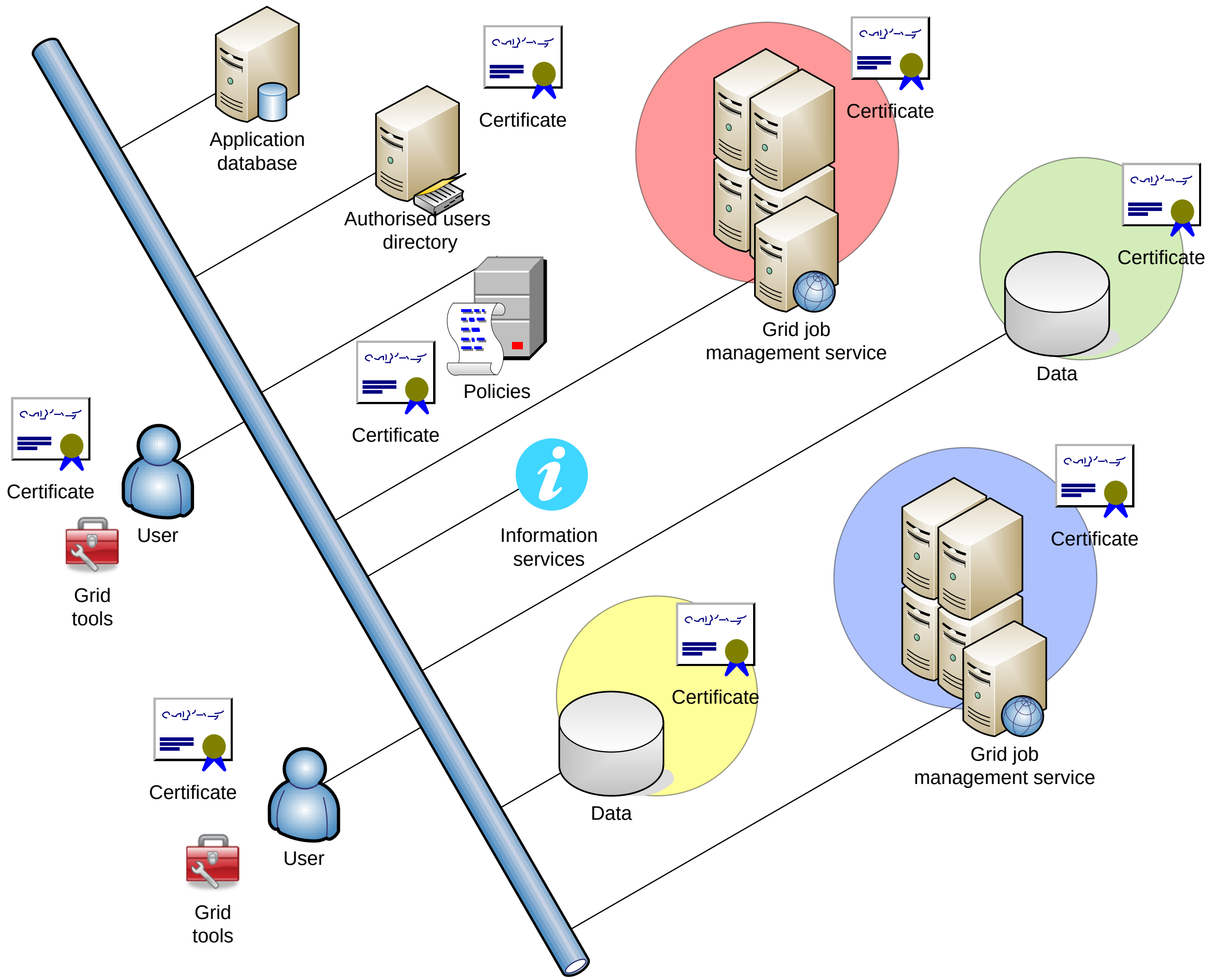
Excellent wide area networks provide for a distributed supercomputer – ***the Grid***

"Operating system" of such a computer is ***Grid middleware***

From the conventional computing…

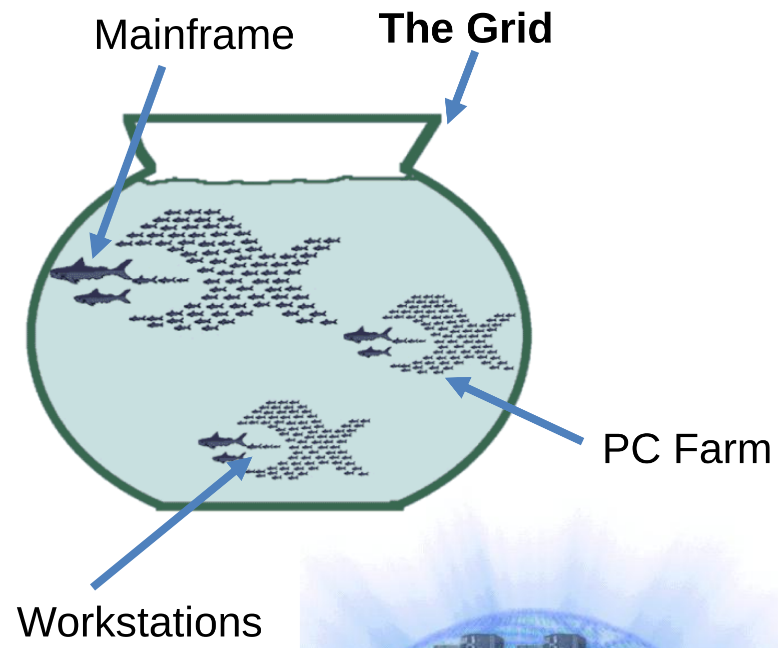


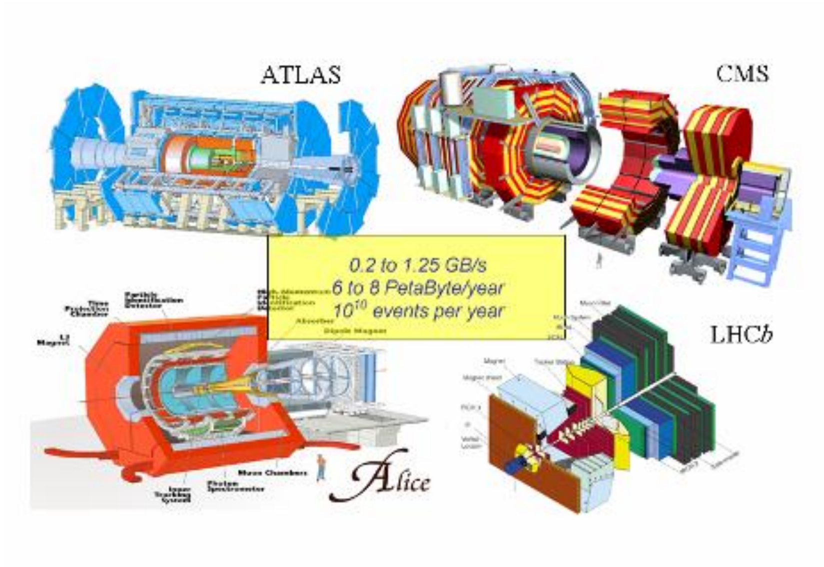
To the Grid

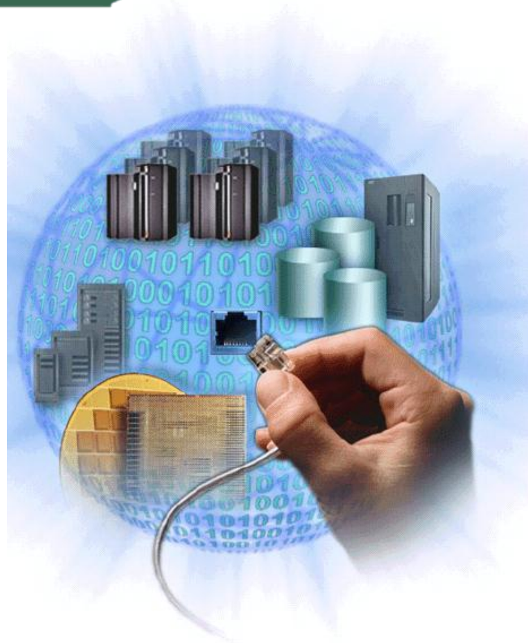

Application
database
Certificate
Authorised users
directory
Certificate
Grid job
management service
Certificate
Data
Policies
Certificate
Certificate
User
Information
services
Grid
tools
Certificate
Grid job
management service
Certificate
Data
Certificate
User
Grid
tools

The Grid Paradigm

- **Distributed “supercomputer”** making use of fast networks
- Access to the great variety of resources by a **single pass** – a digital certificate
- **Distributed data** management
- **A new scientific tool**







Grids in LHC experiments

- Almost all Monte Carlo and data processing today is done via Grid
- There are 20+ Grid flavors out there
  - Almost all are tailored for a specific application and/or specific hardware
- LHC experiments make use of only 3 Grid flavors:
  - gLite
  - ARC
  - OSG

Enabling Grids for E-sciencE

NORDUGRID Nordic Testbed for Wide Area Computing and Data Handling

Open Science Grid

- All experiments develop own higher-level Grid middleware layers
  - ALICE – AliEn
  - ATLAS – PANDA, GANGA, DDM
  - LHCb – DIRAC, GANGA
  - CMS – ProdAgent, CRAB, PhEDEx

What is an LHC "Tier1" center

- ❖ WLCG: Worldwide LHC Computing Grid
  - – A CERN project aiming to provide HEP computing infrastructure
  - – Tiered structure: Tier0 at CERN, a dozen of regional Tier1s, local Tier2s
- ❖ WLCG Tier1 is:
  - – Storage for replicated data (tapes and disks)
  - – Data indexing service
  - – Computing power
  - – 24/7 on-call support system
  - – Infrastructure: network, power, cooling, safety etc
  - – File transfer services between Tiers
  - – Experiment-specific interfaces ("VOBoxes")
  - – Database services
  - – etc

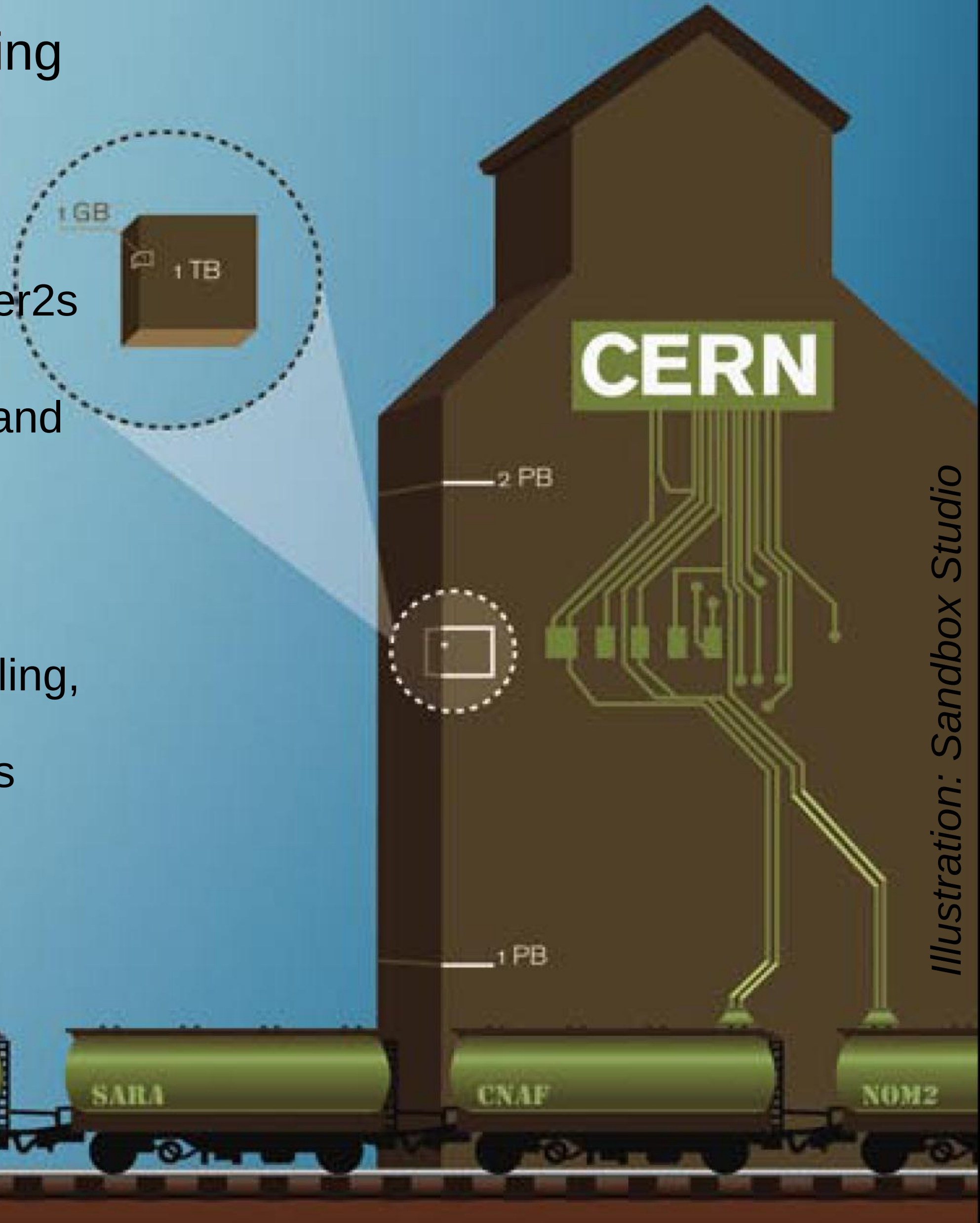
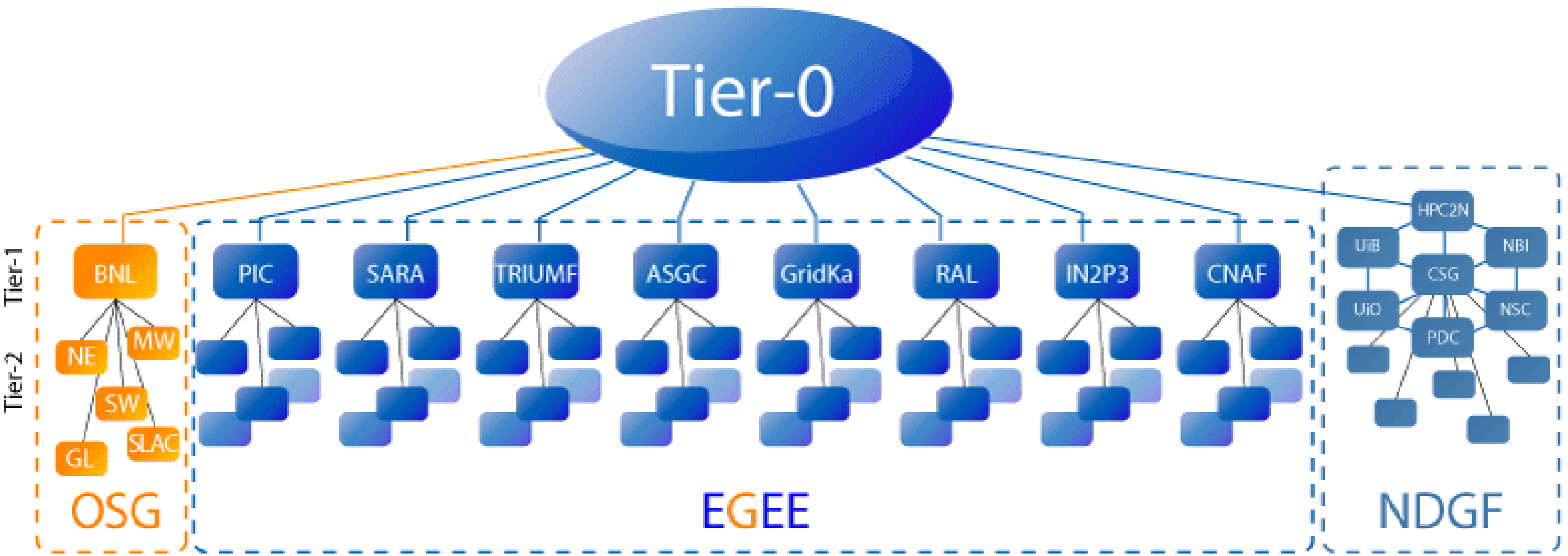


Illustration: Sandbox Studio

ATLAS Multi-Grid Infrastructure



Graphics from a slide by A.Vaniachine

Where do I start on my way to Grid?

- Ask local sysadmin to install a Grid client
  - ARC standalone client is the easiest: you can install it yourself
- Apply for a “passport”: the Grid certificate
  - Every country has a Certificate Authority
- Ask a knowledgeable person which Grid is adopted by your collaboration/group
- Apply for a “visa”: become an appropriate Virtual Organization (VO) member
- Read the manual

What future holds

- So far, Grid concept works well for HEP
  - All LHC data today are processed on the Grid
  - All Monte Carlo for LHC is run on the Grid
- Still, there are many Grid flavors around
  - Some are tailored for HEP, some are not
- European Union aims to create one common European Grid for all sciences
  - Radioastronomy is even more challenging, data-wise
- Competitor: Cloud computing
  - Grid is about federating different resources, good for scientists
  - Cloud is about renting out well-controlled resources, good for business (Amazon, Google, Microsoft)
  - HEP will probably stay with Grid for a while