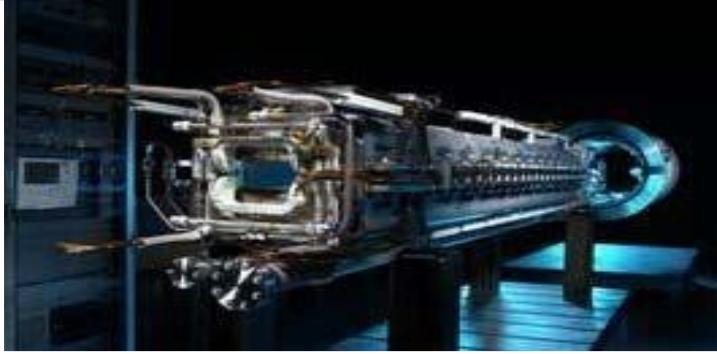
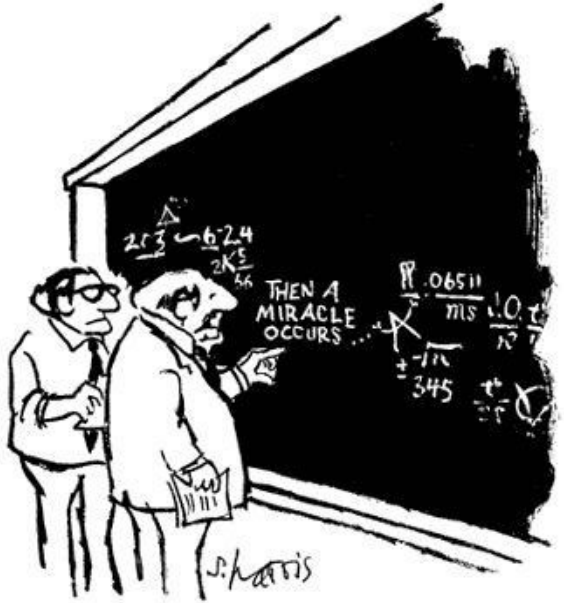


FAIR Accelerators:

Open-Source Challenges & Opportunities

Ralph J. Steinhagen
FAIR Commissioning & Control PL (PSP 2.14.17)

Research Data Management at GSI/FAIR Workshop
4-5 July 2022, GSI, Germany, https://indico.gsi.de/event/14680



"I think you should be more explicit here in step two."

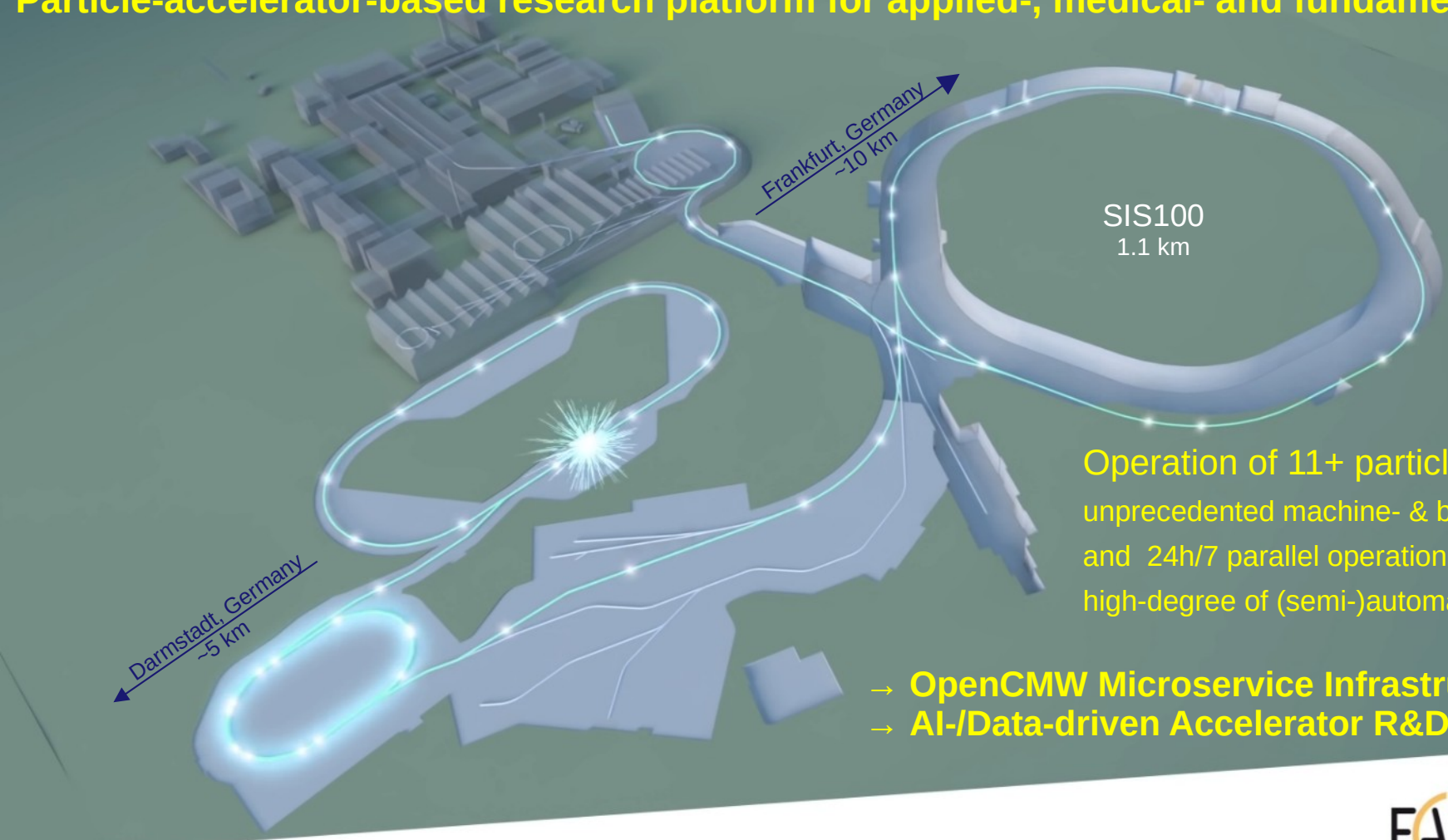
Finland France Germany India Poland Romania Russia Slovenia Sweden UK

GSI HELMHOLTZ ASSOCIATION FAIR

Facility for Anti-proton and Ion Research

"The universe in the laboratory"

Particle-accelerator-based research platform for applied-, medical- and fundamental sciences



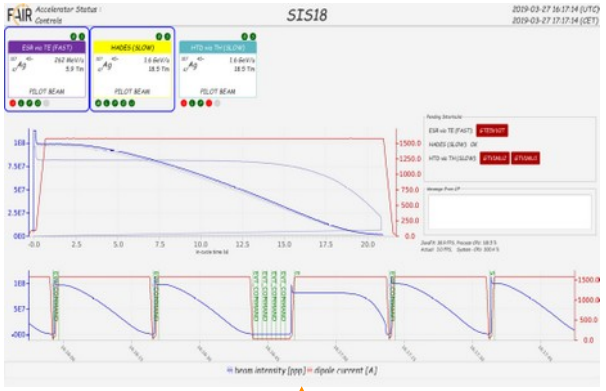
Operation of 11+ particle accelerators:
unprecedented machine- & beam-parameters,
and 24h/7 parallel operation require
high-degree of (semi-)automation

→ **OpenCMW Microservice Infrastructure**
→ **AI-/Data-driven Accelerator R&D**

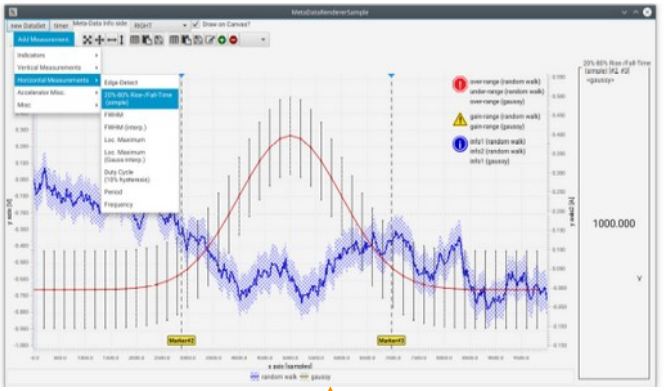
FAIR GSI

(Semi)-Automation of FAIR through Microservices

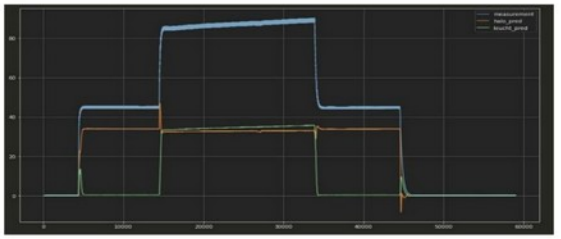
monitoring/web-type status applications



interactive control applications



AI-based
research applications



user-interfaces

micro-services

O(10 → 100 kS/s)

ZeroMQ+{REST}-based middleware

measurement systems

digitizer

x2k

sensor data

feedback controllers

$f(\langle input \rangle) \rightarrow \langle actuator \rangle$

(fairly sub-domain specific)

x15 x N_{acc}

corrections

device controller

DAC

x1k

setting supply

nano-sec. timing

low-level hardware

O(0.1 → 250 MS/s)

particle beam

moving at ~300000 km/s

Experiment

(energy, position, particle-type, -angle, -spot size, -spill-rate, bunch-length, ...)

O(0.001 → 2 GS/s)

>100k diagnostics sensors

>1000 magnets, RF cavities, ...

Accelerator Open-Source Core Activities

https://github.com/fair-acc

A: OpenCMW – Open Common Middle-Ware
library for accelerator equipment- and beam-based control systems at FAIR

- efficient, sustainable, and secure middle-ware minimising distributed micro-service architecture boiler-plate code
- an open reusable standard that facilitates contributions and collaborations with external partners

B: gr-digitizer – generic digitizer and SDR platform for FAIR
i.e. “high-bandwidth distributed oscilloscope & spectrum analysers with thousands of signals, distributed across the FAIR facility, and synchronised from ms- down to the ns-level”

- based on GNU Radio modular high-performance signal processing framework
- part of FAIR’s digitisation and FCC ‘Digital Control Room’ strategy

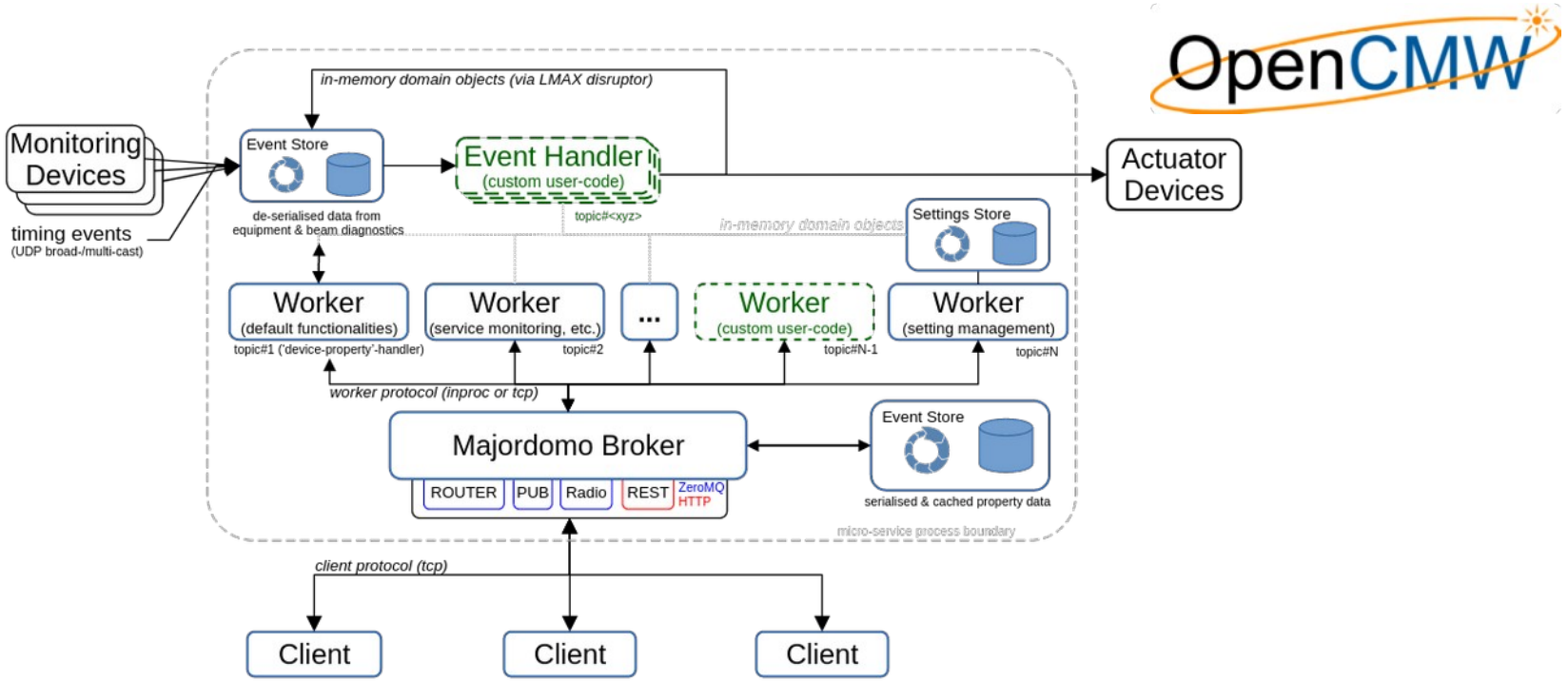
C: ChartFX
scientific charting library focused on performance optimised real-time data visualisation at 25 Hz update rates for datasets with a few 10 thousand up to 5 million data points.

- used in-house by a large number of control room UI application
- large user-community >300+ in non-physics academia and industry (robotics, bio-medical, finance, …)

... following slides briefly cover their scopes and why we opted for an open-source strategy

Open Common Middle-Ware – OpenCMW

https://github.com/fair-acc/opencmw-cpp



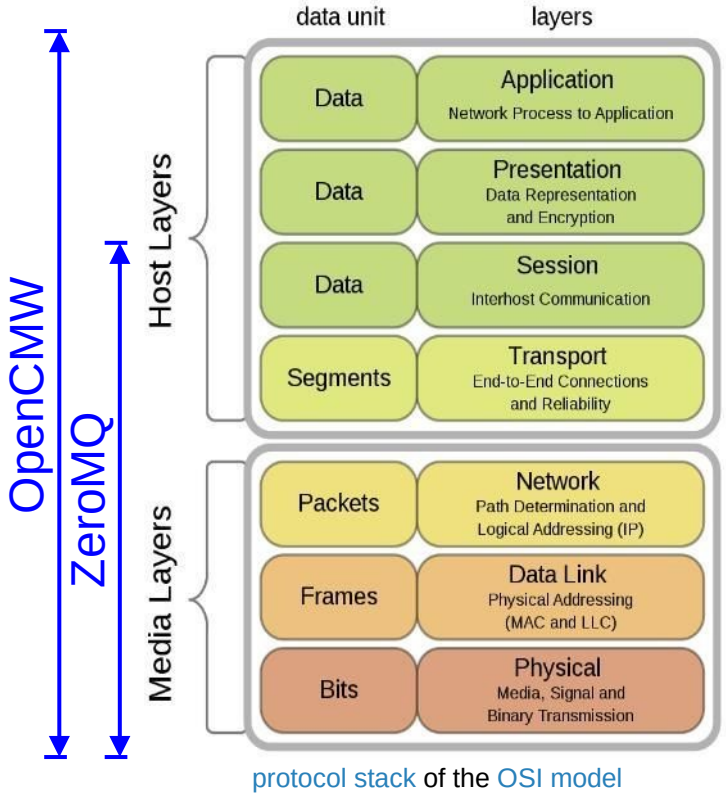
TL;DR summary:

- low-entry threshold for beginners with sufficient domain-expertise but limited C++ expertise
- open, extendable, auditable, and long-term maintainable by limited pool of C++ experts

Intros: https://indico.gsi.de/event/14294/#2-introduction-to-opencmw, https://indico.gsi.de/event/14978/

Open Common Middle-Ware – OpenCMW
Core: Majordomo Transport Protocol

… aims at an extendable full protocol stack implementation…



protocol stack of the OSI model

… with (optional) ‘batteries’ included:

- Transport protocols:
  - Majordomo (ZeroMQ: RFC 7/MDP & 18/MDP),
  - RDA3 (proprietary GSI/CERN transport
  - HTTP/REST (long-polling, SSE): web-services, routable to non-GSI/FAIR networks
  - RADIO/DISH (low-latency UDP) – WIP
  - *… <add your version here>*
- Serialisers
  - YaS (binary): annotated type- and physical unit-safe
  - CmwLight (binary): ACC-specific binary protocol
  - JSON (text): data exchange with web-based REST clients
  - YML (text): service config management (human readable)
  - HTML: server-side rendered fixed-displays (+WASM), expert diagnostics tools, ...
  - *… <add your version here>*
- RPC/Streaming call-backs: lambda → convenience classes
- lock-free circular buffers → event sourcing pattern
- thread-affinity, -tools & -pools
- settings management, ...

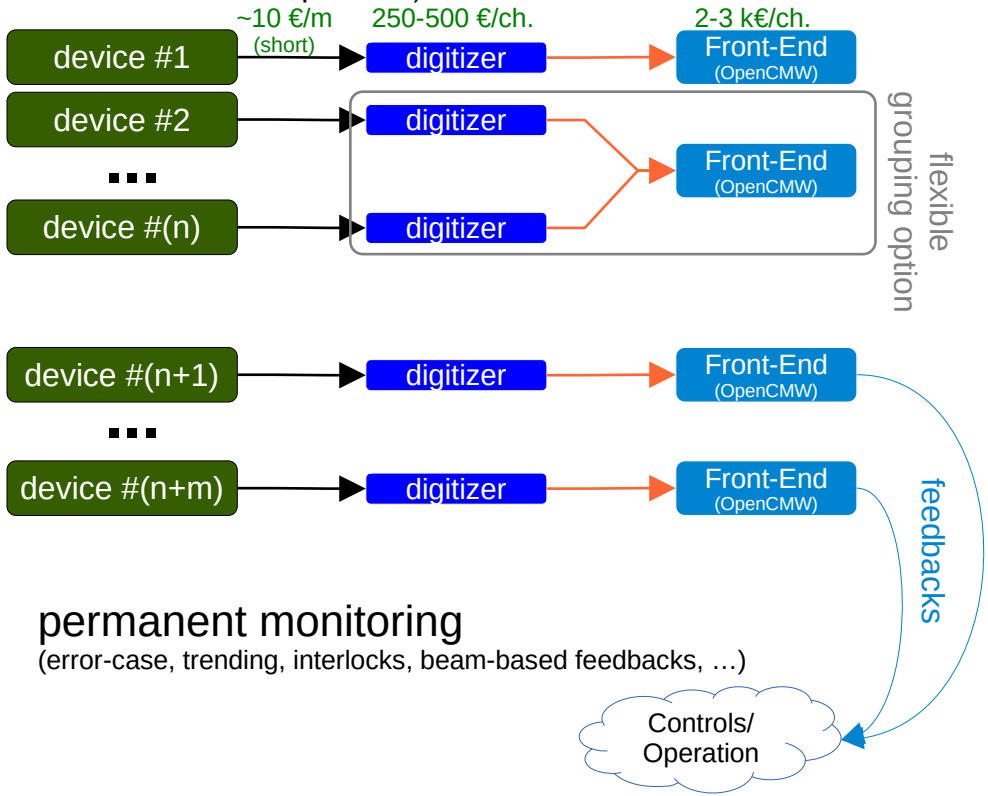
Digitization of Analog Signals at FAIR I/II

https://edms.cern.ch/document/1823376/

ranges: DC → 10 MHz, DC → 200 MHz, DC → 0.5 (1) GHz
courtesy M. Thieme, V. Kleip, K. Lueghausen (ACO)

\+ LimeSDR
0.1 MHz → 4 GHz

- new open-source concept
  (underlying assumption: scopes/digitizers are cheap, RF switches are expensive)

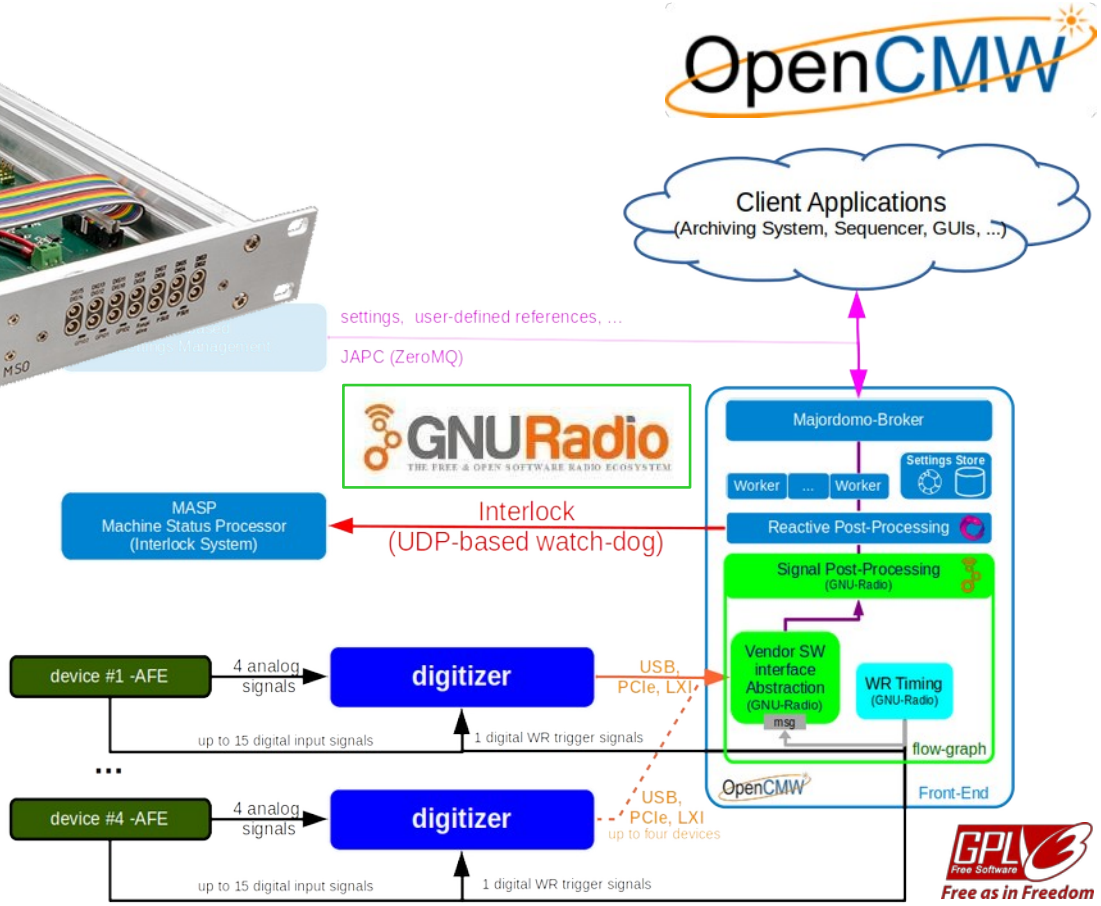


permanent monitoring
(error-case, trending, interlocks, beam-based feedbacks, …)

Open-Source SW & 19” Mechanical Integration

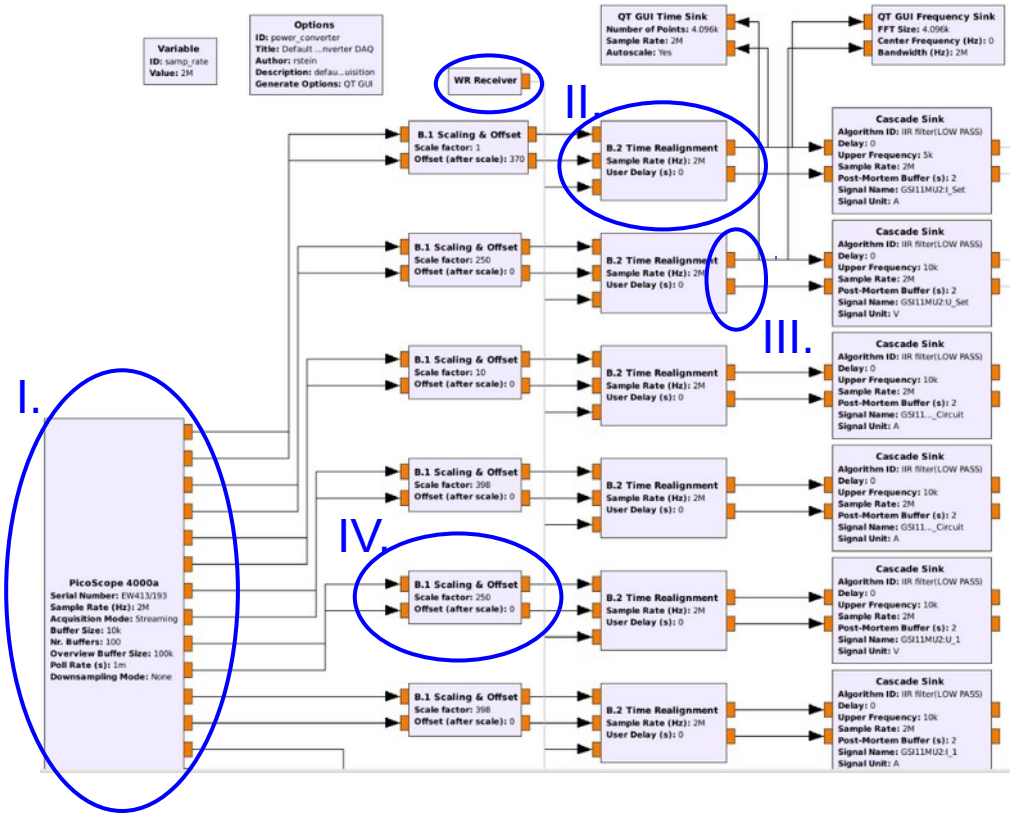
both are open reusable designs: GPLv3 & CERN OHL

open source hardware



presently deployed ~200 systems
→ 300+ systems @FAIR
(many different internal and external groups involved)

General Signal-Flow Scheme – simple example dipole circuit monitoring



Noteworthy things:

I. Integrated high-performance digitizer & SDRs

II. ns-level signal synchronisation across 300++ front-end controllers (FECs) via
(https://github.com/fair-acc/gr-digitizers)
   a) 'White-Rabbit' timing receiver
   b) GPS pps signals
   c) SW-trigger (i.e. UDP multicast)

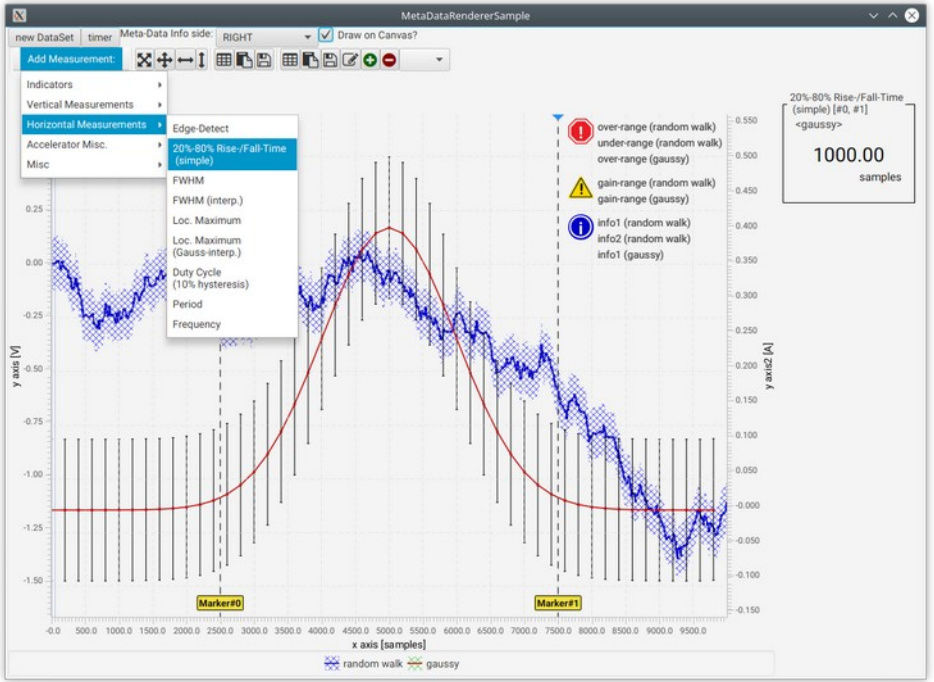
III. mean + stdev processing
   a) … scientific rigour
   b) … signal-integrity checks
   ↔ used in feed-back loops (automatic stop/fail-safe)

IV. run-time flow-graph modifications
(https://github.com/fair-acc/gr-flowgraph)
   a) block parameters
   (e.g. gains, timing-triggered threshold/interlock functions, χ^2-fits, conditional processing, …)
   b) online- & user-defined post-processing
   (~T&M equipment)

- modular open-source high-performance post-processing frame-work (SIMD, GPU-accelerated, ...)
  (large industrial & academic community/eco-system)
- "Best of both worlds":
  - extensible by non-SW experts (UI support, lab-setups)
  - specific/performance-optimised processing blocks written by SW-experts (C++ & Python)

ChartFx – Scientific Charting of Real-Time Beam & Accelerator Data

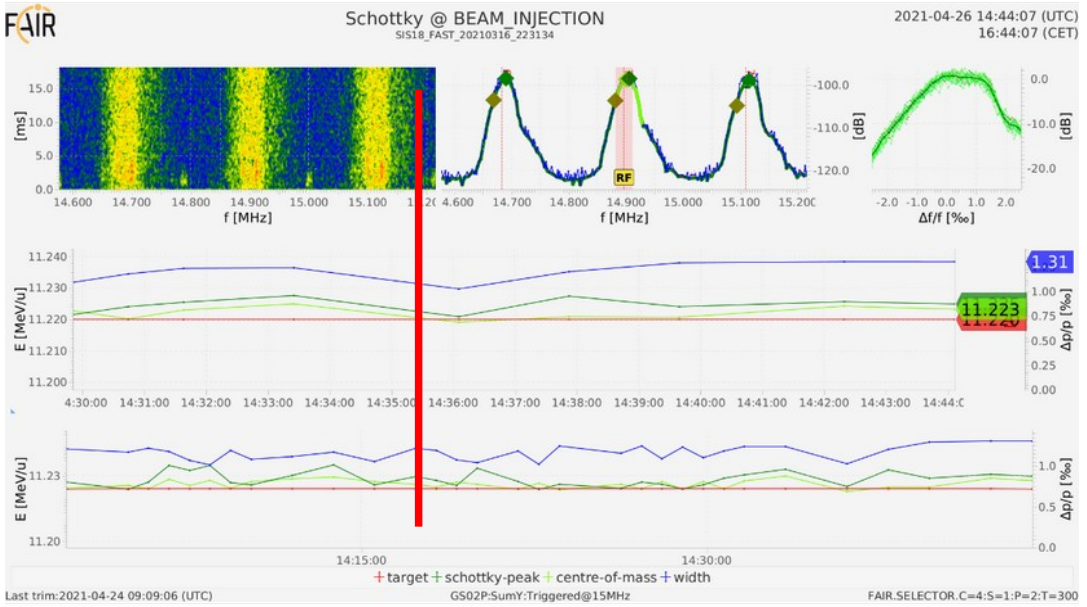
- original design:
  ”all ACC-UIs shall be Java-based”
  N.B. scientific computing/signal-processing remains a niche in Java
- no open-source/commercial charting library available at that time
  → co-use of CERN’s ‘JDataViewer’
  (proprietary use, closed-source → no rights to extend usage to external GSI/FAIR collaborators)
- in addition: became obsolete with Java’s ‘Swing→JavaFX’ transition
  technology effectively died out with most not wanting to publicly invest into this eco-system

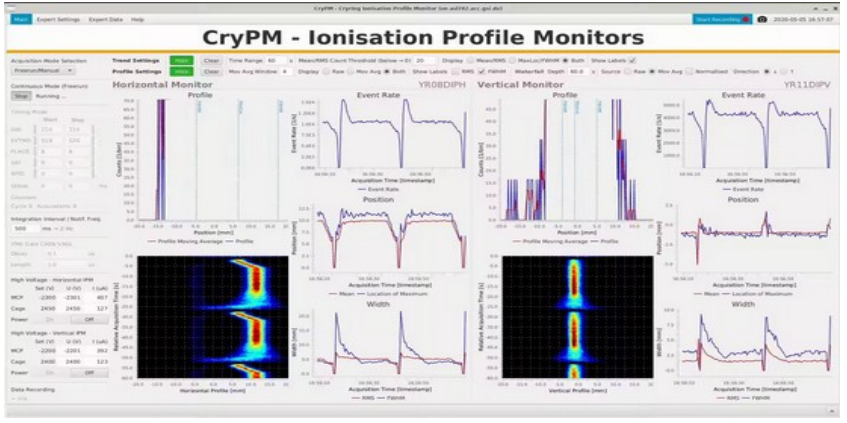


- needed to stick with Java, rolled-out own version using JavaFX → became ‘ChartFX’
- primary design goals:
  A) scientifically accurate and high-performance plotting of real-time data
  B) FLOSS from the beginning: open eco-system anyone can use, ...

ChartFx – Scientific Charting of Real-Time Beam & Accelerator Data

- ‘successful ↔ open-source’ project: versatile, performance, openness, ...
  - used in-house by a large number of control room UI application
  - large user-community >300+ in non-physics academia and industry (robotics, bio-medical, finance, …)
- attracted many 3rd party contributions: new features, review-comments, bug-fixes, ...

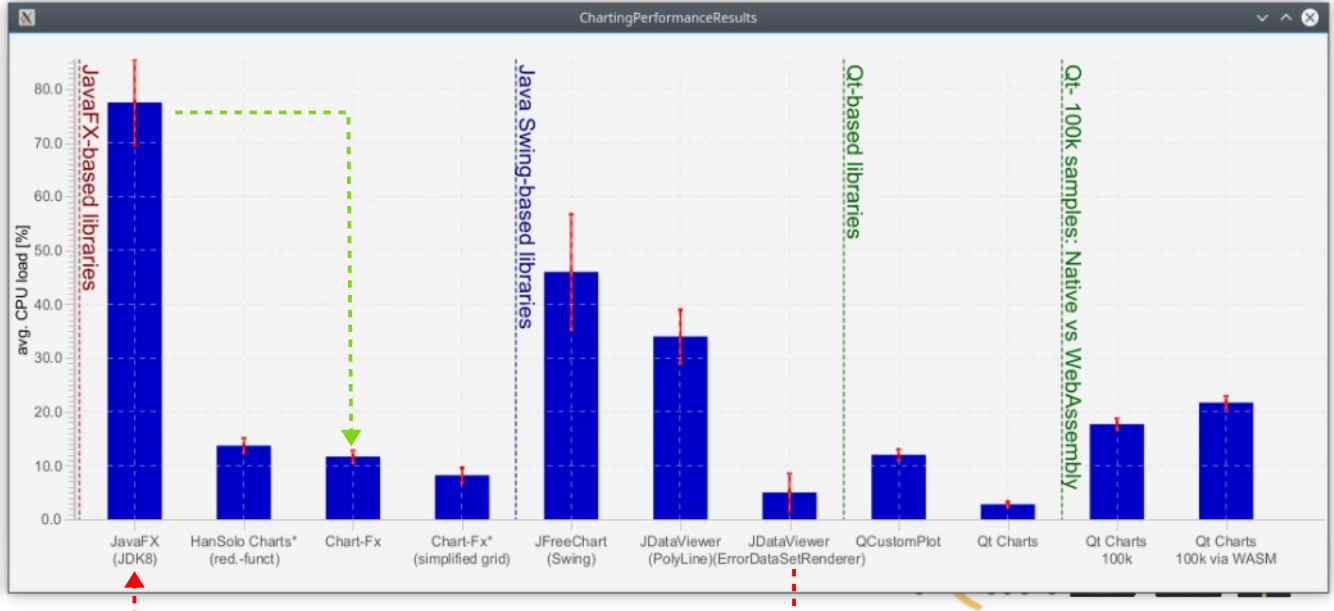




FAIR GSI

Chart Fx – Scientific Charting of Real-Time Beam & Accelerator Data

- ‘successful ↔ open-source’ project: versatile, performance, openness, ...
  - used in-house by a large number of control room UI application
  - large user-community >300+ in non-physics academia and industry (robotics, bio-medical, finance, …)
- attracted many 3rd party contributions: new features, review-comments, bug-fixes, ...



ChartFx – Scientific Charting of Real-Time Beam & Accelerator Data

- ‘successful ↔ open-source’ project: versatile, performance, openness, ...
  - used in-house by a large number of control room UI application
  - large user-community >300+ in non-physics academia and industry (robotics, bio-medical, finance, …)
- attracted many 3rd party contributions: new features, review-comments, bug-fixes, ...

Why Free and Open-Source – Benefits I/III

[Any] Science's central theme:

Why Free and Open-Source – Benefits I/III

[Any] Science's central theme:

A) improve our knowledge about how nature works (the 'R' in R&D)
   i.e. collect evidence (measurement data) to support our hypothesis

Why Free and Open-Source – Benefits I/III

[Any] Science's central theme:

A) improve our knowledge about how nature works (the 'R' in R&D)
i.e. collect evidence (measurement data) to support our hypothesis

B) improve the experimental sensitivity for more subtle/rare effects (the 'D' in R&D)
i.e. improve the support for the evidence (instrumentation, data analysis schemes, tools, …)

FAIR GSI

Why Free and Open-Source – Benefits I/III

[Any] Science's central theme:

A) improve our knowledge about how nature works (the 'R' in R&D)
i.e. collect evidence (measurement data) to support our hypothesis

B) improve the experimental sensitivity for more subtle/rare effects (the 'D' in R&D)
i.e. improve the support for the evidence (instrumentation, data analysis schemes, tools, …)

C) Document/proof that our hypothesis is correct, for example, via **Bayes' theorem**:
any scientifically supported research hypotheses require to publicly document implicit assumptions, tooling, and data in order to be replicable, verifiable, and to gain strong support for any level of confidence, notably we …

FAIR GSI

Why Free and Open-Source – Benefits I/III

[Any] Science's central theme:

A) improve our knowledge about how nature works (the 'R' in R&D)
i.e. collect evidence (measurement data) to support our hypothesis

B) improve the experimental sensitivity for more subtle/rare effects (the 'D' in R&D)
i.e. improve the support for the evidence (instrumentation, data analysis schemes, tools, …)

C) Document/proof that our hypothesis is correct, for example, via **Bayes' theorem**:
any scientifically supported research hypotheses require to publicly document implicit assumptions, tooling, and data in order to be replicable, verifiable, and to gain strong support for any level of confidence, notably we …
- … prefer simplicity → Ockham's Razor → limit/understand scope and 'degrees of freedom'

FAIR GSI

Why Free and Open-Source – Benefits I/III

[Any] Science's central theme:

A) improve our knowledge about how nature works (the 'R' in R&D)
i.e. collect evidence (measurement data) to support our hypothesis

B) improve the experimental sensitivity for more subtle/rare effects (the 'D' in R&D)
i.e. improve the support for the evidence (instrumentation, data analysis schemes, tools, …)

C) Document/proof that our hypothesis is correct, for example, via **Bayes' theorem**:
any scientifically supported research hypotheses require to publicly document implicit assumptions, tooling, and data in order to be replicable, verifiable, and to gain strong support for any level of confidence, notably we …

- … prefer simplicity → Ockham's Razor → limit/understand scope and 'degrees of freedom'
- … prefer universal and produce stable underlying laws

FAIR GSI

Why Free and Open-Source – Benefits I/III

[Any] Science's central theme:

A) improve our knowledge about how nature works (the 'R' in R&D)
i.e. collect evidence (measurement data) to support our hypothesis

B) improve the experimental sensitivity for more subtle/rare effects (the 'D' in R&D)
i.e. improve the support for the evidence (instrumentation, data analysis schemes, tools, …)

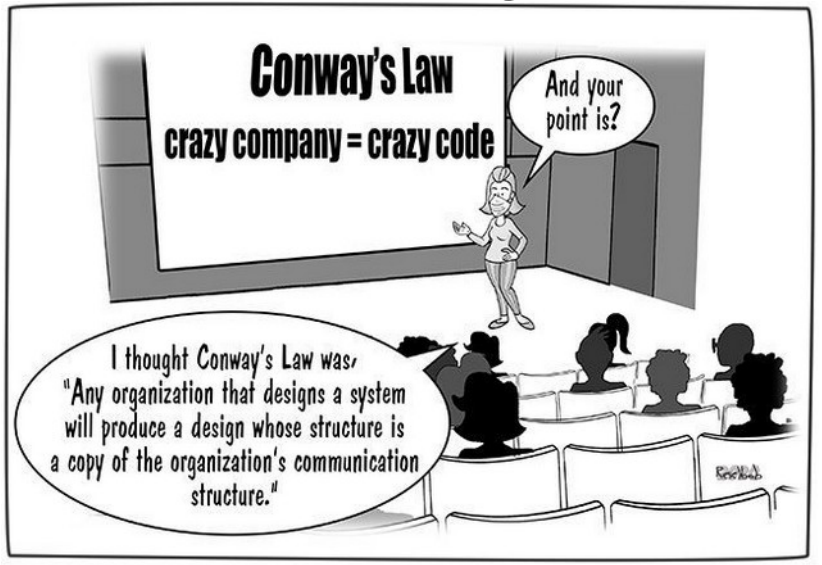
C) Document/proof that our hypothesis is correct, for example, via **Bayes' theorem**:
any scientifically supported research hypotheses require to publicly document implicit assumptions, tooling, and data in order to be replicable, verifiable, and to gain strong support for any level of confidence, notably we …

- … prefer simplicity → Ockham's Razor → limit/understand scope and 'degrees of freedom'
- … prefer universal and produce stable underlying laws
- … requires reproducibility and 'FAIR' principles (**F**indability, **A**ccessibility, **I**nteroperability, and **R**euse):
  - replicate data analysis → Open-Access & Open-Data
  - replicate experimental setup and measurements
    → **Open-Source** since most of our analyses are SW-driven

FAIR GSI

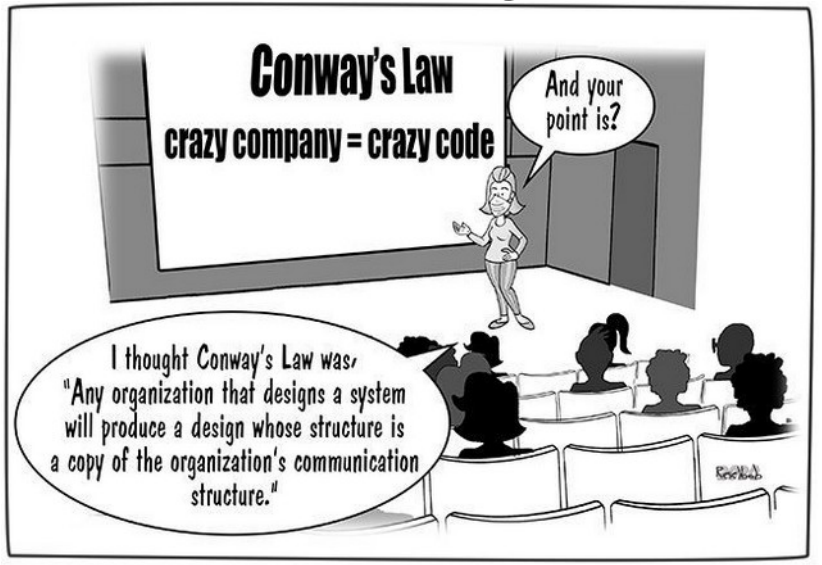
Why Free and Open-Source – The Organisational Challenge

- our (maybe yours?) developments tended to be ...
  - immediate need, functionality, and limited resources driven
  - closed-source and w/ limited ‘works-as-is’ documentation
  - available only for specific platforms, dedicated machines, and for a selective pool of specially trained developers



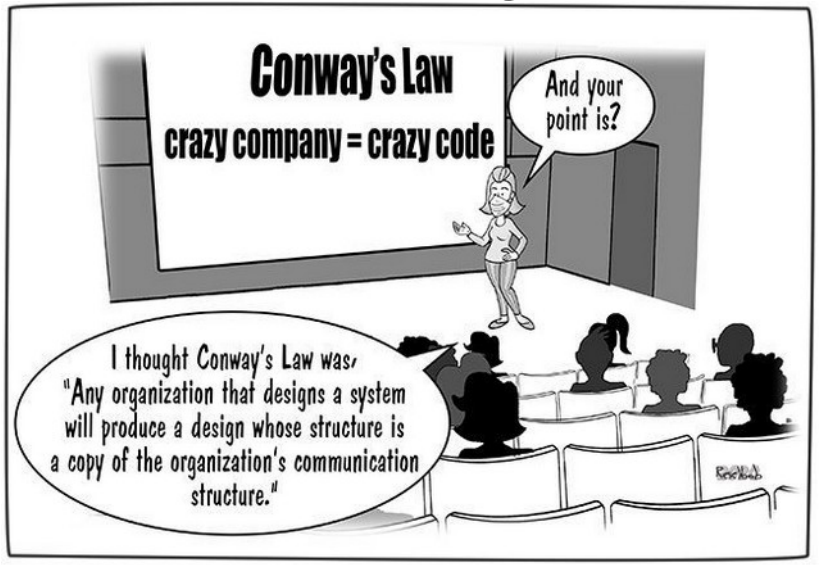
Why Free and Open-Source – The Organisational Challenge

- our (maybe yours?) developments tended to be ...
  - immediate need, functionality, and limited resources driven
  - closed-source and w/ limited ‘works-as-is’ documentation
  - available only for specific platforms, dedicated machines, and for a selective pool of specially trained developers



- fear of support & maintenance burden/risks lead to a slow creeping process that:
  - forced fixed long-term design choices
  - constrained continuous improvement and roll-out of new features
  - accumulated technical debt → bloated code and furthered high maintenance efforts
  - strong emphasise on control and minimising potential heterogeneity of expanding software eco-system rather than embracing an evolution of new ideas and concepts

Why Free and Open-Source – The Organisational Challenge



- our (maybe yours?) developments tended to be ...
  - immediate need, functionality, and limited resources driven
  - closed-source and w/ limited 'works-as-is' documentation
  - available only for specific platforms, dedicated machines, and for a selective pool of specially trained developers

- fear of support & maintenance burden/risks lead to a slow creeping process that:
  - forced fixed long-term design choices
  - constrained continuous improvement and roll-out of new features
  - accumulated technical debt → bloated code and furthered high maintenance efforts
  - strong emphasise on control and minimising potential heterogeneity of expanding software eco-system rather than embracing an evolution of new ideas and concepts

- relied on formal specific collaboration agreements (effectively NDAs)
  → high threshold for including 3rd parties
  → difficult to contract generic developments to external companies/universities

Why Free and Open-Source – Benefits II/III

… intentionally aimed at breaking these old bad habits & to open-up for new concepts, i.e. 'Inverse Conway's Law': *"purpose-led adoption of agile, lean & clean development principles that are inviting for other scientists/engineers and shape the organisation"*

FAIR GSI

Why Free and Open-Source – Benefits II/III

… intentionally aimed at breaking these old bad habits & to open-up for new concepts, i.e. 'Inverse Conway's Law': *"purpose-led adoption of agile, lean & clean development principles that are inviting for other scientists/engineers and shape the organisation"*

1. opted to make 'free and open-source' the default → aim at maximum transparency
   - free/libre use by any other public organisation, private, or commercial entity

FAIR GSI

Why Free and Open-Source – Benefits II/III

… intentionally aimed at breaking these old bad habits & to open-up for new concepts, i.e. 'Inverse Conway's Law': *"purpose-led adoption of agile, lean & clean development principles that are inviting for other scientists/engineers and shape the organisation"*

1. opted to make 'free and open-source' the default → aim at maximum transparency
   - free/libre use by any other public organisation, private, or commercial entity

2. enabled ad-hoc & required collaborations with ext. partners with low legal threshold
   - shared innovation – cross-pollination of ideas and designs
     - feedback from industry, best-practices, minimised risk for intellectual in-breeding
   - facilitates (in particular unexpected) 3rd party contributions (> 300 users)
     - positive feedback & external contributions outweigh our community support effort

FAIR GSI

Why Free and Open-Source – Benefits II/III

… intentionally aimed at breaking these old bad habits & to open-up for new concepts, i.e. 'Inverse Conway's Law': *"purpose-led adoption of agile, lean & clean development principles that are inviting for other scientists/engineers and shape the organisation"*

1. opted to make 'free and open-source' the default → aim at maximum transparency
   - free/libre use by any other public organisation, private, or commercial entity

2. enabled ad-hoc & required collaborations with ext. partners with low legal threshold
   - shared innovation – cross-pollination of ideas and designs
     - feedback from industry, best-practices, minimised risk for intellectual in-breeding
   - facilitates (in particular unexpected) 3rd party contributions (> 300 users)
     - positive feedback & external contributions outweigh our community support effort

3. fosters higher quality assurance and coding standards
   - higher SW quality standards: peer review, feedback, bug reports/fixes, …
   - static code-analysis, CI/CD and other otherwise inaccessible tooling

Why Free and Open-Source – Benefits III/III

4. premise for sustainably growing community and software eco-system,

FAIR GSI

Why Free and Open-Source – Benefits III/III

4. premise for sustainably growing community and software eco-system,
   - democratises development → transparency w.r.t. internal/external goals and processes
     → being predictable & long-term dependable (i.e. ‘playing with open cards’ and creating a public track record)

Why Free and Open-Source – Benefits III/III

4. premise for sustainably growing community and software eco-system,
   - democratises development → transparency w.r.t. internal/external goals and processes
     → being predictable & long-term dependable (i.e. ‘playing with open cards’ and creating a public track record)
   - keeping a low-entry threshold – being open for and value small contribution and changes
     - limit mandatory external technical dependencies
     - limit prior required knowledge to use for own development/research
     - better documented and open build processes → better auditable QA processes
       (e.g. controlled via public CI/CD-integrated agile development rules, styles and standards)
     - limit boiler-plate code → lean and ‘clean code’ principles

FAIR GSI

Why Free and Open-Source – Benefits III/III

4. premise for sustainably growing community and software eco-system,
   - democratises development → transparency w.r.t. internal/external goals and processes
     → being predictable & long-term dependable (i.e. 'playing with open cards' and creating a public track record)
   - keeping a low-entry threshold – being open for and value small contribution and changes
     - limit mandatory external technical dependencies
     - limit prior required knowledge to use for own development/research
     - better documented and open build processes → better auditable QA processes
       (e.g. controlled via public CI/CD-integrated agile development rules, styles and standards)
     - limit boiler-plate code → lean and 'clean code' principles
   - yields better design aiming at re-usability also in other domains
     (generic scalable vs. specific hard-to-extend/unmaintainable designs)
     - also: shared eco-system ↔ shared maintenance

FAIR GSI

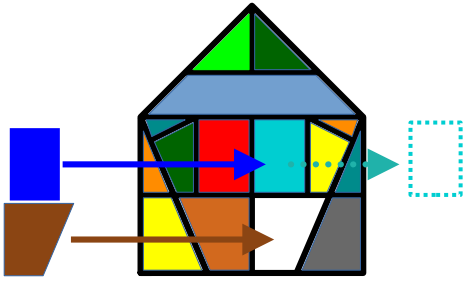
Why Free and Open-Source – Benefits III/III

4. premise for sustainably growing community and software eco-system,
   - democratises development → transparency w.r.t. internal/external goals and processes
     → being predictable & long-term dependable (i.e. 'playing with open cards' and creating a public track record)
   - keeping a low-entry threshold – being open for and value small contribution and changes
     - limit mandatory external technical dependencies
     - limit prior required knowledge to use for own development/research
     - better documented and open build processes → better auditable QA processes
       (e.g. controlled via public CI/CD-integrated agile development rules, styles and standards)
     - limit boiler-plate code → lean and 'clean code' principles
   - yields better design aiming at re-usability also in other domains
     (generic scalable vs. specific hard-to-extend/unmaintainable designs)
     - also: shared eco-system ↔ shared maintenance
   - core: 'prefer composition over inheritance'

FAIR GSI

Why Free and Open-Source – Benefits III/III

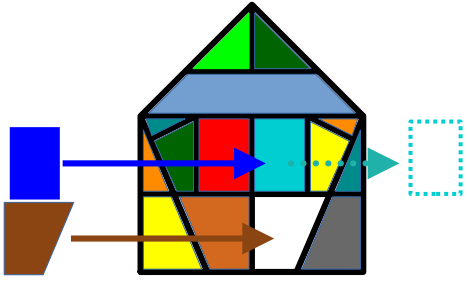
4. premise for sustainably growing community and software eco-system,
   - democratises development → transparency w.r.t. internal/external goals and processes
     → being predictable & long-term dependable (i.e. 'playing with open cards' and creating a public track record)
   - keeping a low-entry threshold – being open for and value small contribution and changes
     - limit mandatory external technical dependencies
     - limit prior required knowledge to use for own development/research
     - better documented and open build processes → better auditable QA processes
       (e.g. controlled via public CI/CD-integrated agile development rules, styles and standards)
     - limit boiler-plate code → lean and 'clean code' principles
   - yields better design aiming at re-usability also in other domains
     (generic scalable vs. specific hard-to-extend/unmaintainable designs)
     - also: shared eco-system ↔ shared maintenance
   - core: 'prefer composition over inheritance'



(prescriptive rigid) frameworks:
user implement stubs
limited options to exchange
or to extend
(often monolithic structures)

Why Free and Open-Source – Benefits III/III

4. premise for sustainably growing community and software eco-system,
   - democratises development → transparency w.r.t. internal/external goals and processes
     → being predictable & long-term dependable (i.e. 'playing with open cards' and creating a public track record)
   - keeping a low-entry threshold – being open for and value small contribution and changes
     - limit mandatory external technical dependencies
     - limit prior required knowledge to use for own development/research
     - better documented and open build processes → better auditable QA processes
       (e.g. controlled via public CI/CD-integrated agile development rules, styles and standards)
     - limit boiler-plate code → lean and 'clean code' principles
   - yields better design aiming at re-usability also in other domains
     (generic scalable vs. specific hard-to-extend/unmaintainable designs)
     - also: shared eco-system ↔ shared maintenance
   - core: 'prefer composition over inheritance'



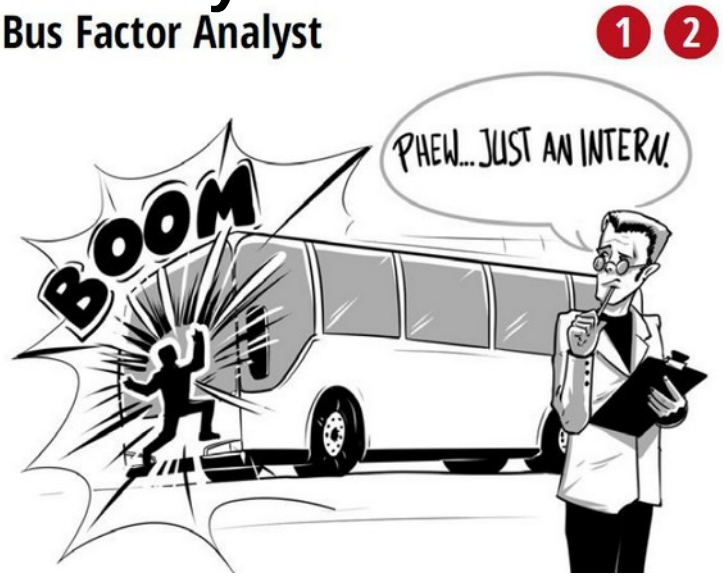
(prescriptive rigid) frameworks:
user implement stubs
limited options to exchange
or to extend
(often monolithic structures)

modular library:
user can opt-in what to use and
what is needed, free to extend,
modify, synthesise new ideas, …
(e.g. header-only libs)

→ fosters long-term maintainability, extendability, and real-world R&D impact FAIR GSI

A Better Impact of R&D – Bus Factor & Reusability

… *“number of team members that have to disappear from a project before the project stalls due to lack of knowledgeable or competent personnel"*, Wikipedia



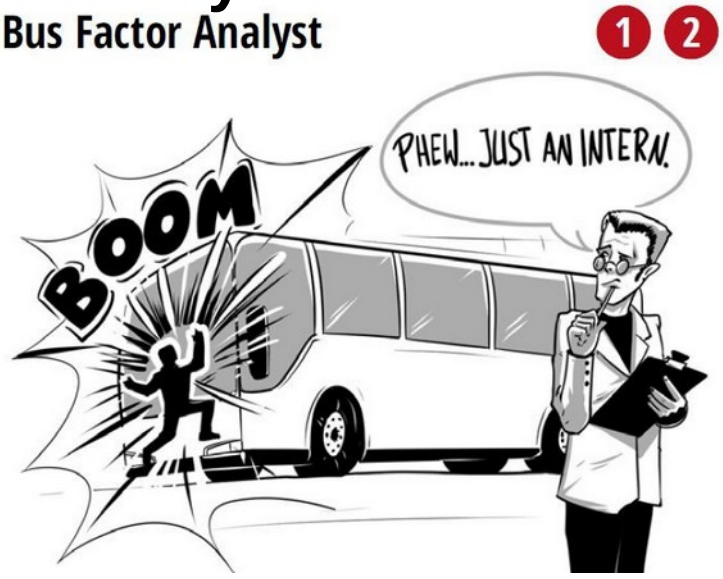
https://www.playitstartup.com/

A Better Impact of R&D – Bus Factor & Reusability

… *“number of team members that have to disappear from a project before the project stalls due to lack of knowledgeable or competent personnel"*, Wikipedia

example: bus-factor of

- ==1: e.g. one-of code-dump to meet student’s PhD requirements
  - limited use (1) and low return-on-investment (ROI)



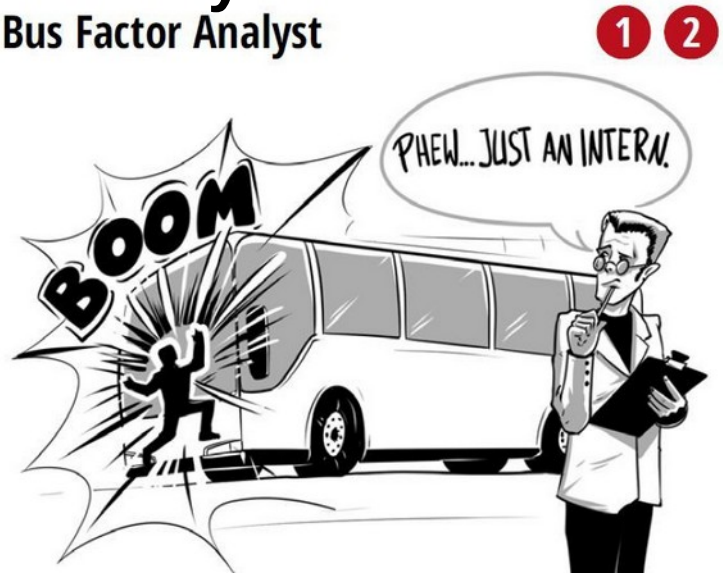
https://www.playitstartup.com/

A Better Impact of R&D – Bus Factor & Reusability

… *“number of team members that have to disappear from a project before the project stalls due to lack of knowledgeable or competent personnel"*, Wikipedia

example: bus-factor of

- ==1: e.g. one-of code-dump to meet student’s PhD requirements
  - limited use (1) and low return-on-investment (ROI)
- >>1: cross-functional team, minimum required complexity, better/clean code, sustainable development, well documented, creates shared value and opportunity also across initial domain
  → higher chance of being used, reproduced, and useful ROI



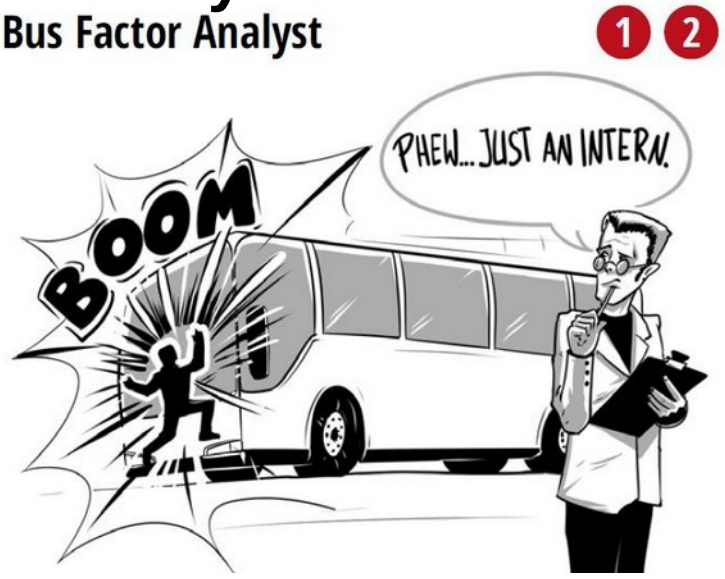
https://www.playitstartup.com/

A Better Impact of R&D – Bus Factor & Reusability

… *"number of team members that have to disappear from a project before the project stalls due to lack of knowledgeable or competent personnel"*, Wikipedia

example: bus-factor of

- ==1: e.g. one-of code-dump to meet student's PhD requirements
  - limited use (1) and low return-on-investment (ROI)
- >>1: cross-functional team, minimum required complexity, better/clean code, sustainable development, well documented, creates shared value and opportunity also across initial domain
  → higher chance of being used, reproduced, and useful ROI



https://www.playitstartup.com/

… IMO a better metric measuring the real academic/societal impact of R&D projects

- fosters building communities and software eco-system
- cross-pollination of ideas from different domains → universality of R&D
- significantly improves code- and project quality → public show-and-tell
- more practical real-impact for both academics and society in general

Open-Source platforms (GitHub, GitLab, …) are strong-enabler/continuous tracker of improving the 'bus factor' and overall health/impact of an R&D project

FAIR GSI

Why Free and Open-Source – Our Responsibilities

- ‘free and open source software’ (FOSS) principle
  → essentially based on honesty, ethics, and common scientific citation practices
  - acknowledge previous work your work builds- and extends upon
  - ‘copyleft’: grant the same rights to others that you have received (i.e. pay-it-forward principle)

FAIR GSI

Why Free and Open-Source – Our Responsibilities

- ‘free and open source software’ (FOSS) principle
  → essentially based on honesty, ethics, and common scientific citation practices
  - acknowledge previous work your work builds- and extends upon
  - ‘copyleft’: grant the same rights to others that you have received (i.e. pay-it-forward principle)
    - *N.B. imagine ‘applying for a licence’ to be allowed to build-upon Newton’s or Einstein’s theories of gravity, QCD, …*
    - core-issue w.r.t. OSI: power to the user/next developer or the original author (i.e. Tesla/Edison debate/debacle)

FAIR GSI

Why Free and Open-Source – Our Responsibilities

- ‘free and open source software’ (FOSS) principle
  → essentially based on honesty, ethics, and common scientific citation practices
  - acknowledge previous work your work builds- and extends upon
  - ‘copyleft’: grant the same rights to others that you have received (i.e. pay-it-forward principle)
    - *N.B. imagine ‘applying for a licence’ to be allowed to build-upon Newton’s or Einstein’s theories of gravity, QCD, …*
    - core-issue w.r.t. OSI: power to the user/next developer or the original author (i.e. Tesla/Edison debate/debacle)

- *‘dissemination before monetarisation’*: generating ‘good will’ for fundamental R&D is more valuable than monetary exploitation via patents stifling innovation & collaborations

FAIR GSI

Why Free and Open-Source – Our Responsibilities

- ‘free and open source software’ (FOSS) principle
  → essentially based on honesty, ethics, and common scientific citation practices
  - acknowledge previous work your work builds- and extends upon
  - ‘copyleft’: grant the same rights to others that you have received (i.e. pay-it-forward principle)
    - *N.B. imagine ‘applying for a licence’ to be allowed to build-upon Newton’s or Einstein’s theories of gravity, QCD, …*
    - core-issue w.r.t. OSI: power to the user/next developer or the original author (i.e. Tesla/Edison debate/debacle)

- *‘dissemination before monetarisation’*: generating ‘good will’ for fundamental R&D is more valuable than monetary exploitation via patents stifling innovation & collaborations

- intrinsic democratic responsibility being publicly funded (no double taxation)
  → FSFE’s ‘Public Money Public Code’ campaign
  - min. ‘reinventing the wheel’ and wasting of public resources → ‘bus-factor’ metric
  - Have you seen/already signed this?
    (N..B. it’s also in-line with our government coalition policy)

FAIR GSI

Why Free and Open-Source – Our Responsibilities

- ‘free and open source software’ (FOSS) principle
  → essentially based on honesty, ethics, and common scientific citation practices
  - acknowledge previous work your work builds- and extends upon
  - ‘copyleft’: grant the same rights to others that you have received (i.e. pay-it-forward principle)
    - *N.B. imagine ‘applying for a licence’ to be allowed to build-upon Newton’s or Einstein’s theories of gravity, QCD, …*
    - core-issue w.r.t. OSI: power to the user/next developer or the original author (i.e. Tesla/Edison debate/debacle)

- *‘dissemination before monetarisation’*: generating ‘good will’ for fundamental R&D is more valuable than monetary exploitation via patents stifling innovation & collaborations

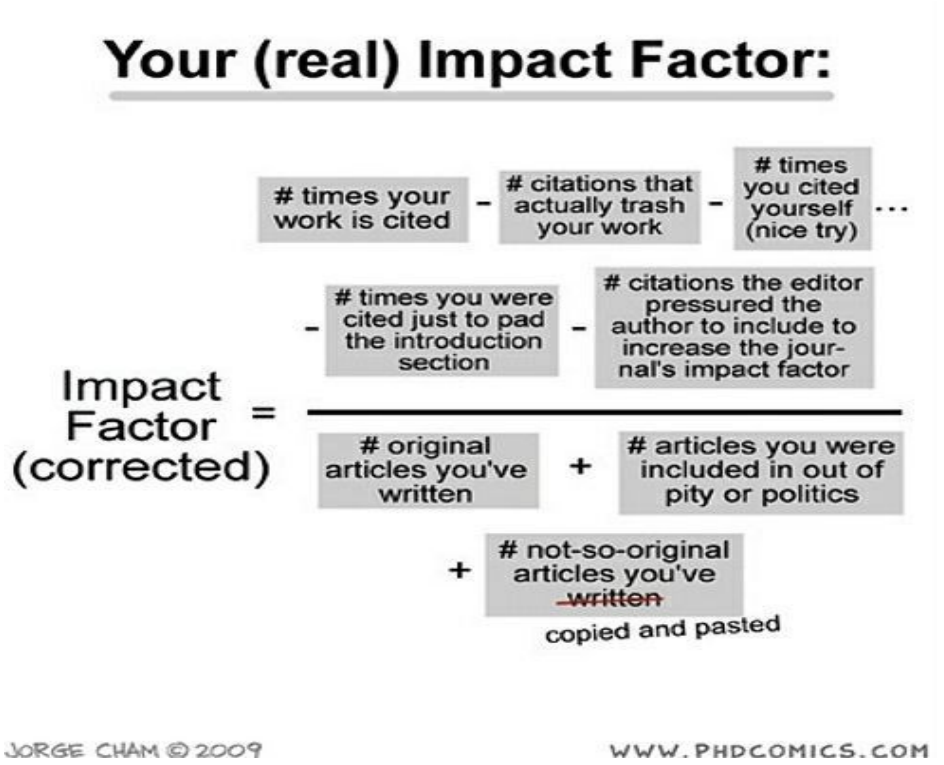
- intrinsic democratic responsibility being publicly funded (no double taxation)
  → FSFE’s ‘Public Money Public Code’ campaign
  - min. ‘reinventing the wheel’ and wasting of public resources → ‘bus-factor’ metric
  - Have you seen/already signed this?
    (N..B. it’s also in-line with our government coalition policy)

- More general open question:
  ”Should GSI/FAIR at large publicly commit to the ‘Public Money Public Code’ statement and adopt a policy where FOSS is the mandatory default* for R&D?”
  - *with exceptions to this being explicitly justified, cost-benefit analysed, and be peer-reviewed?

FAIR GSI

APPENDIX

Historical Measurements of R&D Impact – h-index et al.



We know that most of such simple metrics are wrong *… but often seem to not accept this*

- social media info war, R&D funding processes, 'reward function gaming' (especially in AI)
- broken peer-review processes, under-powered studies, ...
  e.g. John P. A. Ioannidis, "Why Most Published Research Findings Are False", PLOS Medicine, 2005, https://doi.org/10.1371/journal.pmed.0020124

Ockham's Razor

- *… There are always infinitely many potential explanations for any observation*

→ *Assume explanation with the least number of extra or unproven assumption*

William of Ockham
1288-1348

- More “modern” version:
  - “Rule I. We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances.” (Isaac Newton, Principia mathematica, 1687)
  - “Whenever possible, substitute constructions out of known entities for inferences to unknown entities.” (Bertrand Russell, 1924)

- Without, no useful knowledge can be gained about anything, ever!

Bayes' Theorem

- Defines level of confidence P(H|E) in any Hypothesis, depending the level of support by the Evidence:

Thomas Bayes
1701-1761

$$P(H|E) = \frac{P(E|H)}{P(E)} \cdot P(H)$$

Posterior | Support | Prior

- shows what effect new evidence should have on our perception

"Extraordinary claims require extraordinary evidence."

Carl Sagan (paraphrasing Hume & Laplace)

N.B. alternate writing: $P_E(H) = P(H|E)$

FAIR GSI