



# Designing a Licence for Open Collaboration: insights from the development and use of the CERN Open Hardware Licence

Andrew Katz

University of Skövde, Moorcrofts LLP

[www.moorcrofts.com](http://www.moorcrofts.com)

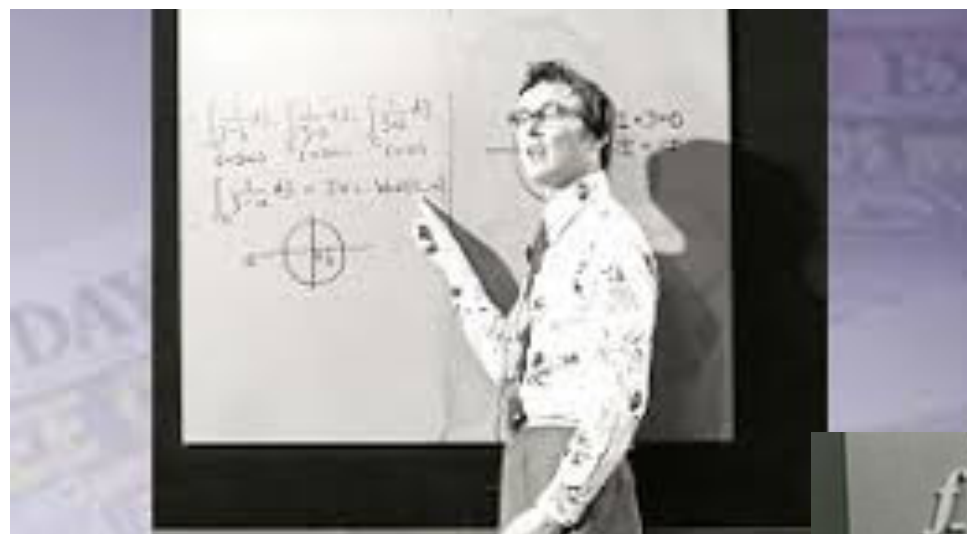
andrew.katz@moorcrofts.com



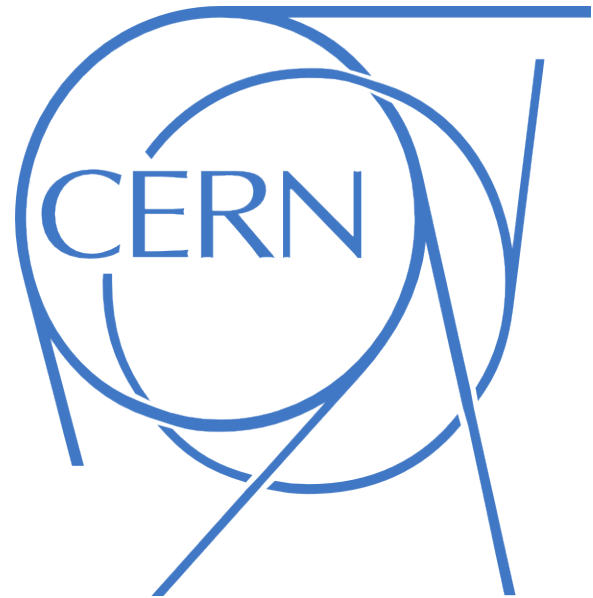
The Open  
University

---

**50**  
YEARS



# Javier Serrano

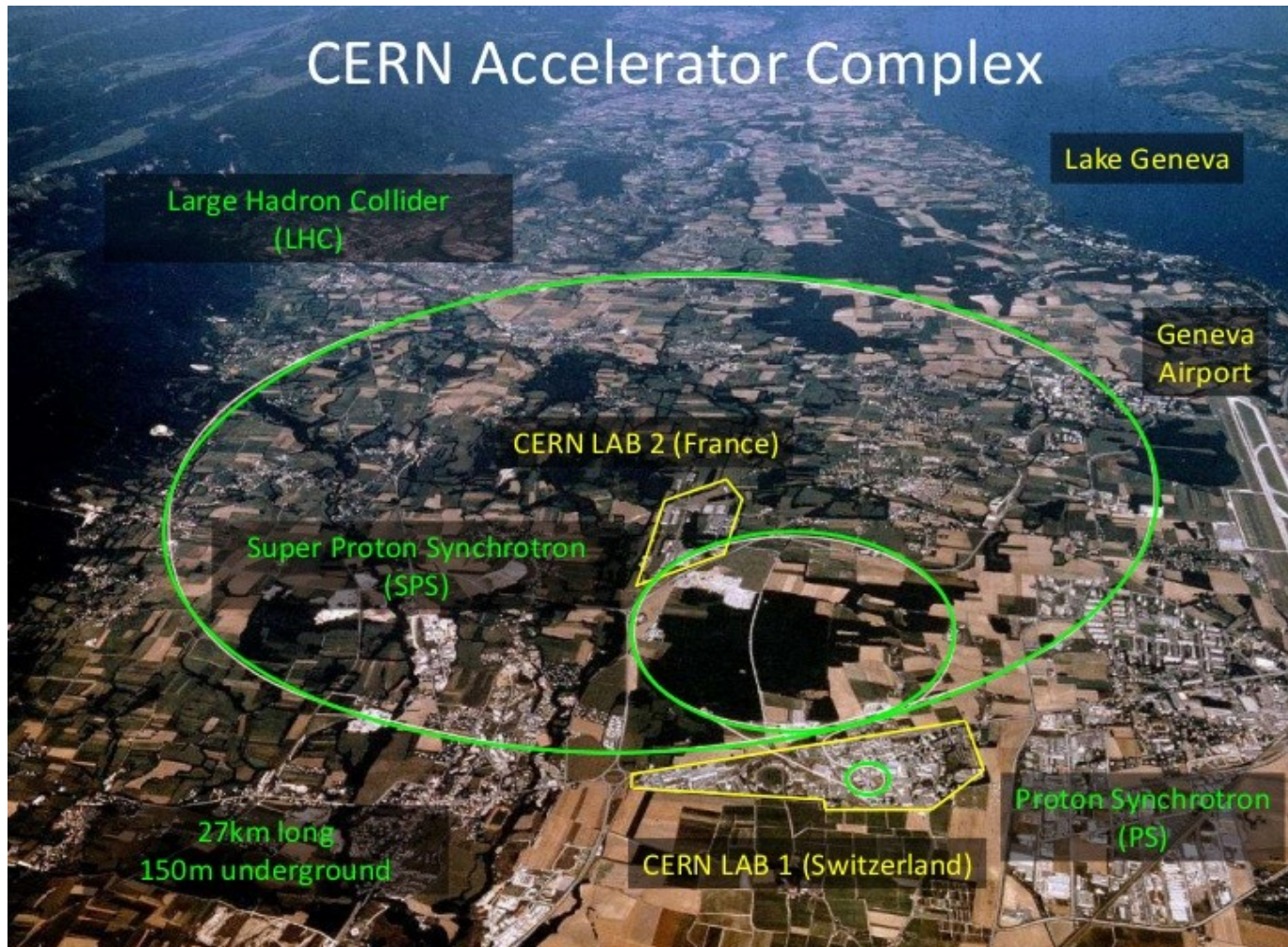




# CERN OHL History

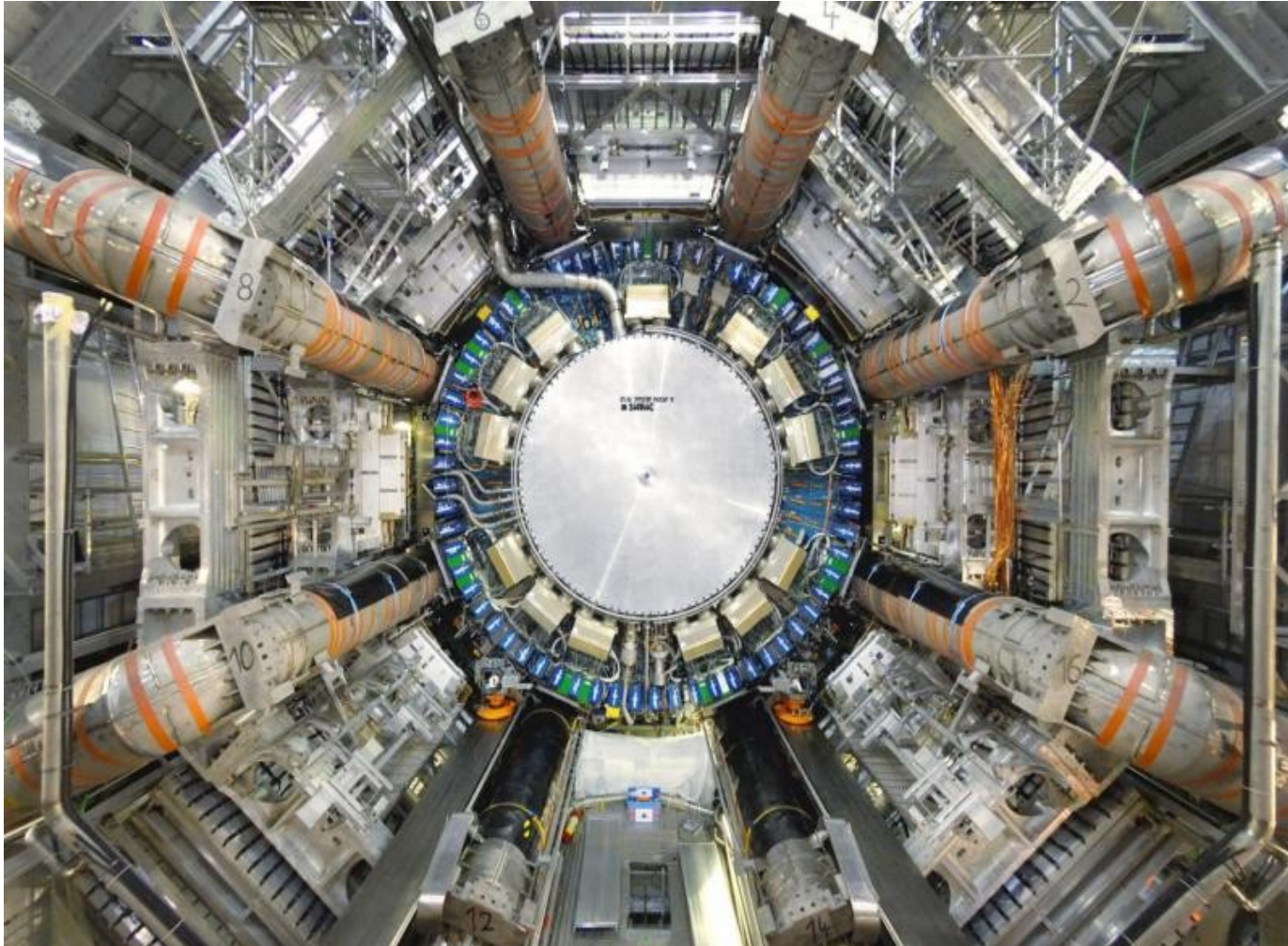
- March 2011: CERN OHL 1.0
- July 2011: CERN OHL 1.1
- September 2013: CERN OHL 1.2
- 2017: CERN OHL 2, beta 1
- 2019 : CERN OHL 2, beta 2
- Original drafting team: Myriam Ayass and Javier Serrano and the CERN Knowledge Transfer Group.
- AK became involved in 2012 with v1.2

# Accelerators





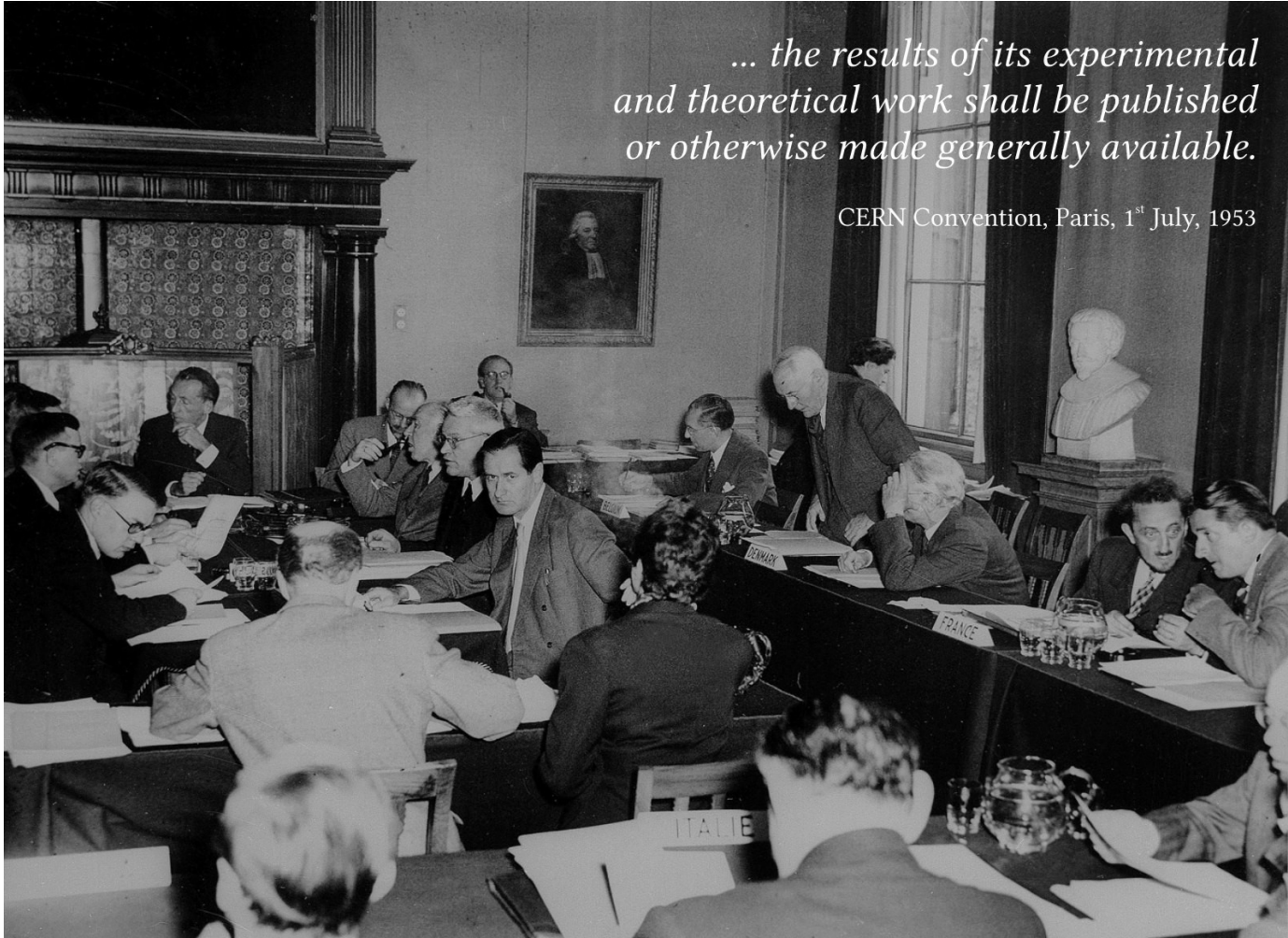
# Detectors



# Dissemination

*... the results of its experimental  
and theoretical work shall be published  
or otherwise made generally available.*

CERN Convention, Paris, 1<sup>st</sup> July, 1953





# How to interpret one's dissemination mandate in the 21st century

Free and Open Source Software

Open Hardware

Open Data

Open Access

World Wide Web

open source

Open Hardware Repository

INSPIRE

Information Management: A Proposal

Tim Berners-Lee, CERN/DD

March 1989

Abstract

This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext system.

Keywords: Hypertext, Computer conferencing, Document retrieval, Information management, Project control

An Updated Historical Profile of the Higgs Boson

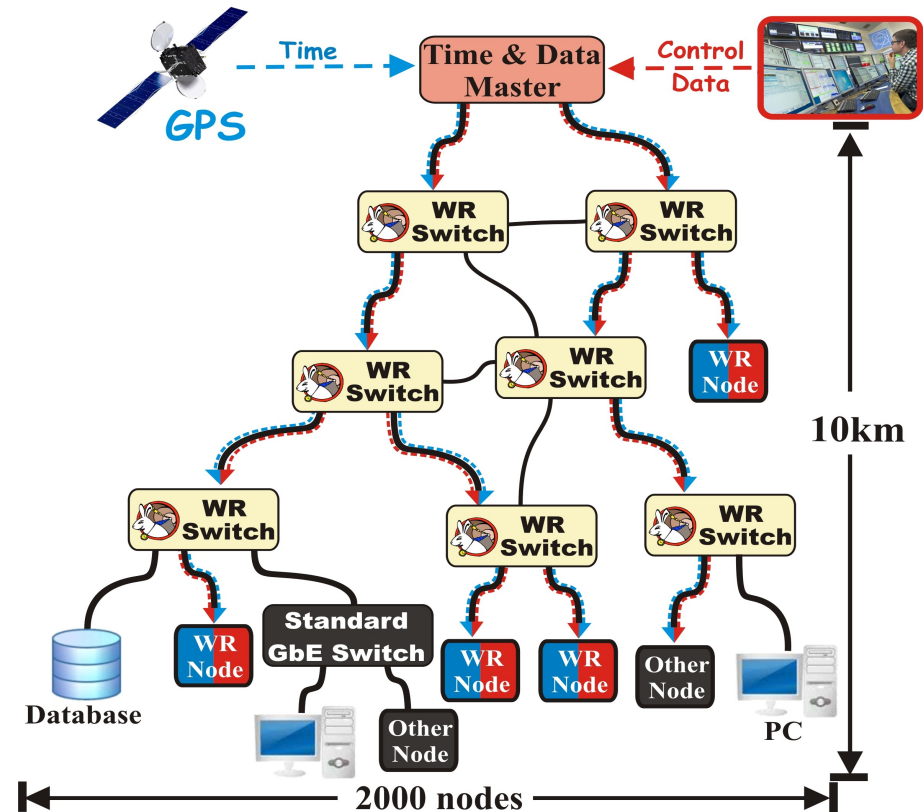
John Ellis (King's Coll. London & CERN), Mary K. Gaillard (LBL, Berkeley & UC, Berkeley), Dimitri V. Nanopoulos (Texas A-M & HARC, Woodlands & Athens Academy)

Apr 27, 2015 - 22 pages

KCL-PH-TH-2015-20, LCTS-2015-10, CERN-PH-TH-2015-098  
e-Print: [arXiv:1504.07217](https://arxiv.org/abs/1504.07217) [hep-ph] | PDF

# How to interpret one's dissemination mandate in the 21st century

- Standard Ethernet network
- Ethernet features (VLAN) & protocols (SNMP)
- Sub-nanosecond synchronisation
- **Guaranteed (by design) upper bound in frame latency**

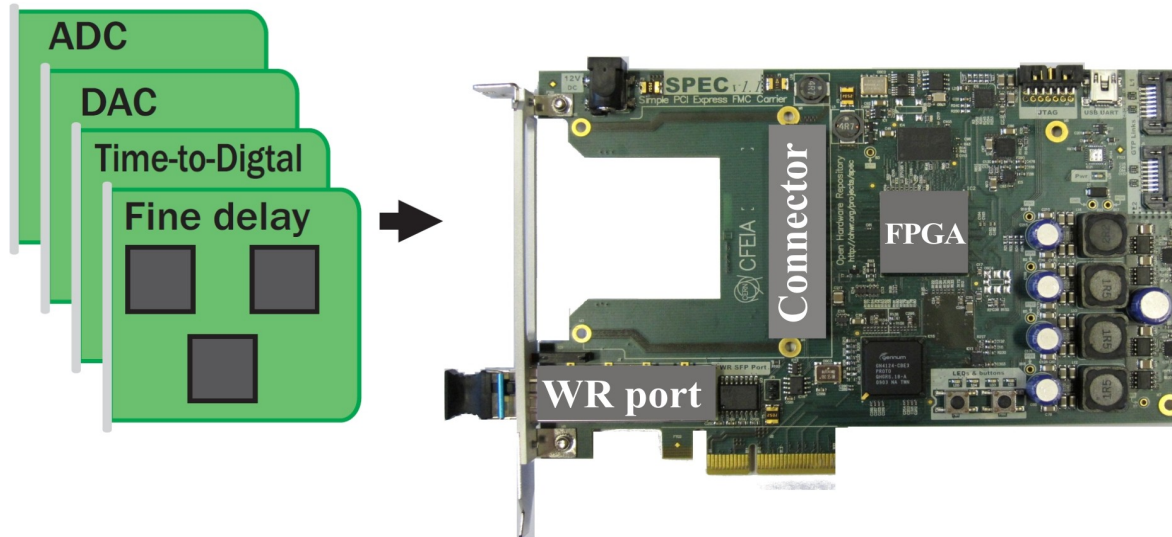


# White Rabbit Switch



- Central element of White Rabbit network
- 18 port gigabit Ethernet switch with WR features
- Optical transceivers: single-mode fibre, originally 10 km range
- Fully open design, commercially available

# WR Node: SPEC board



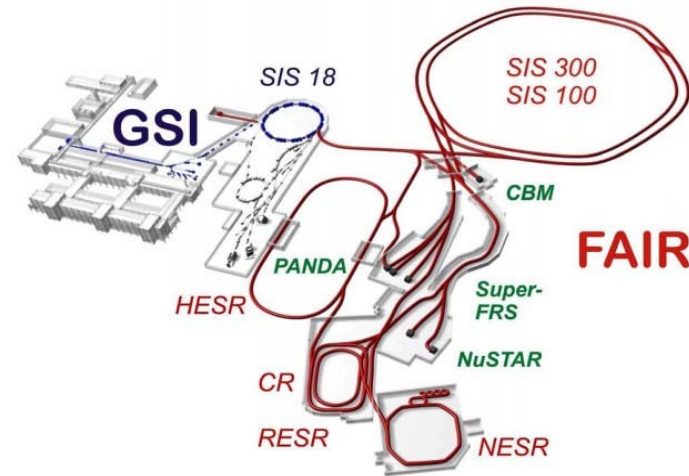
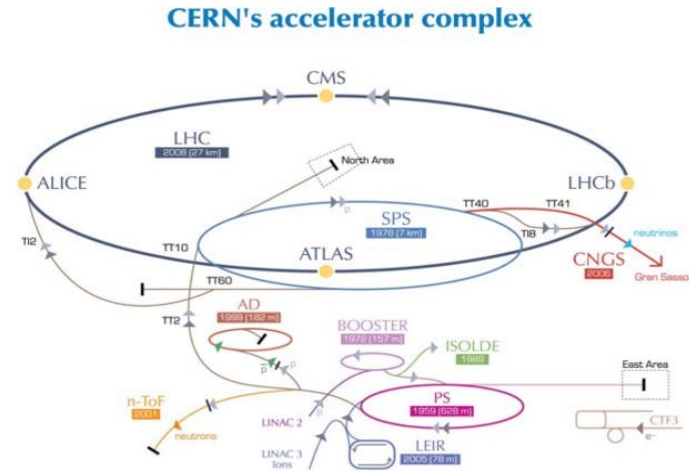
## FMC-based Hardware Kit:

- All carrier cards are equipped with a White Rabbit port.
- Mezzanines can use the accurate clock signal and “TAI” (synchronous sampling clock, trigger time tag, . . . ).



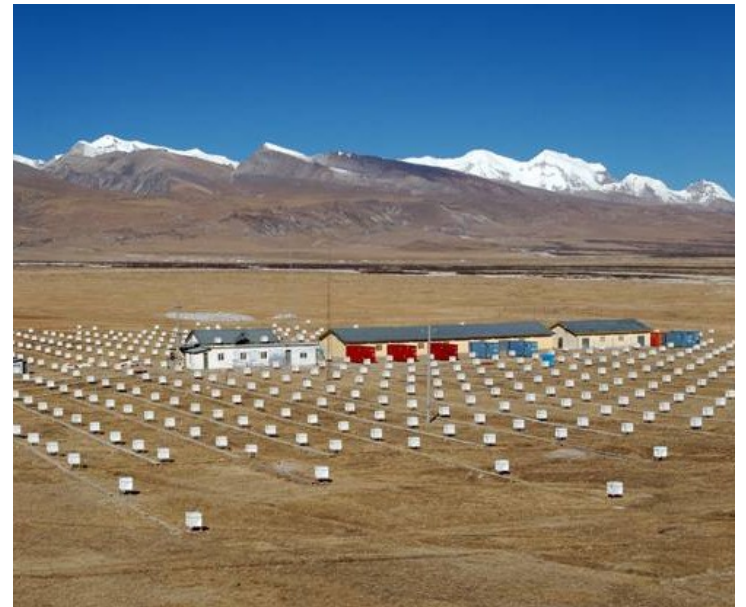
# White Rabbit application examples

- CERN and GSI (Germany) near Darmstadt



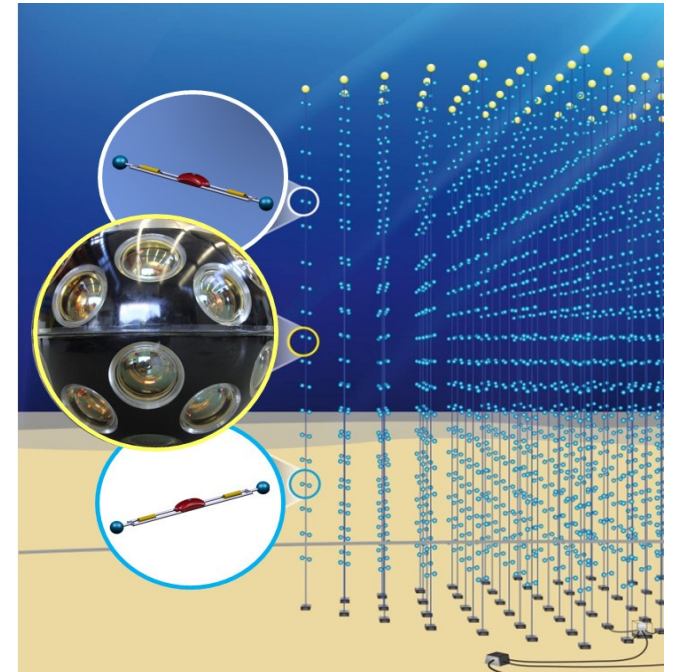
# White Rabbit application examples

- CERN and GSI
- The Large High Altitude Air Shower Observatory



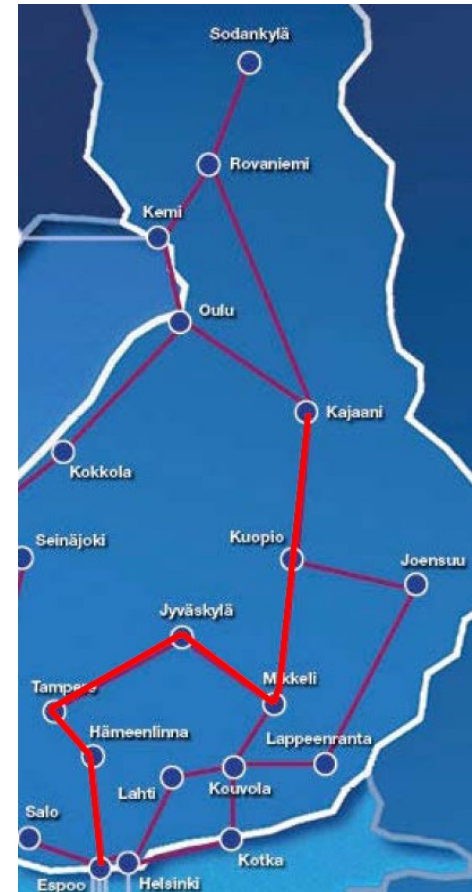
# White Rabbit application examples

- CERN and GSI
- The Large High Altitude Air Shower Observatory
- **KM3NET: European deep-sea neutrino telescope**



# White Rabbit application examples

- CERN and GSI
- The Large High Altitude Air Shower Observatory
- KM3NET: European deep-sea neutrino telescope
- MIKES: Centre for metrology and accreditation

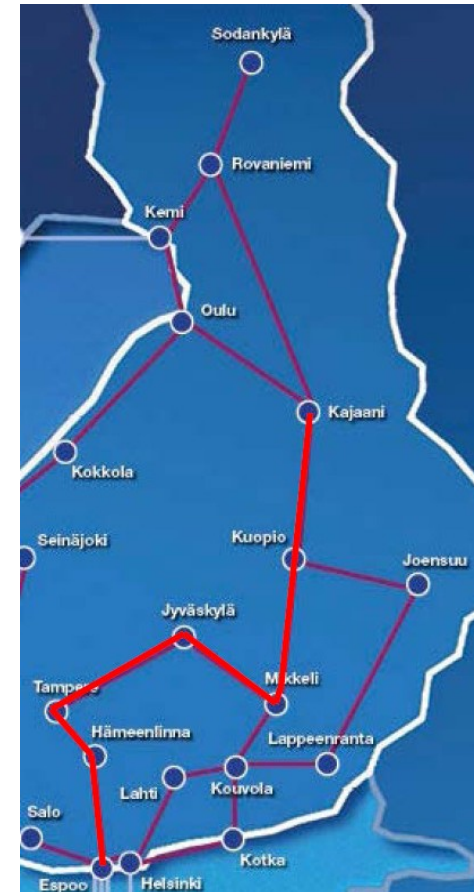


# White Rabbit application examples

- CERN and GSI
- The Large High Altitude Air Shower Observatory
- KM3NET: European deep-sea neutrino telescope
- MIKES: Centre for metrology and accreditation

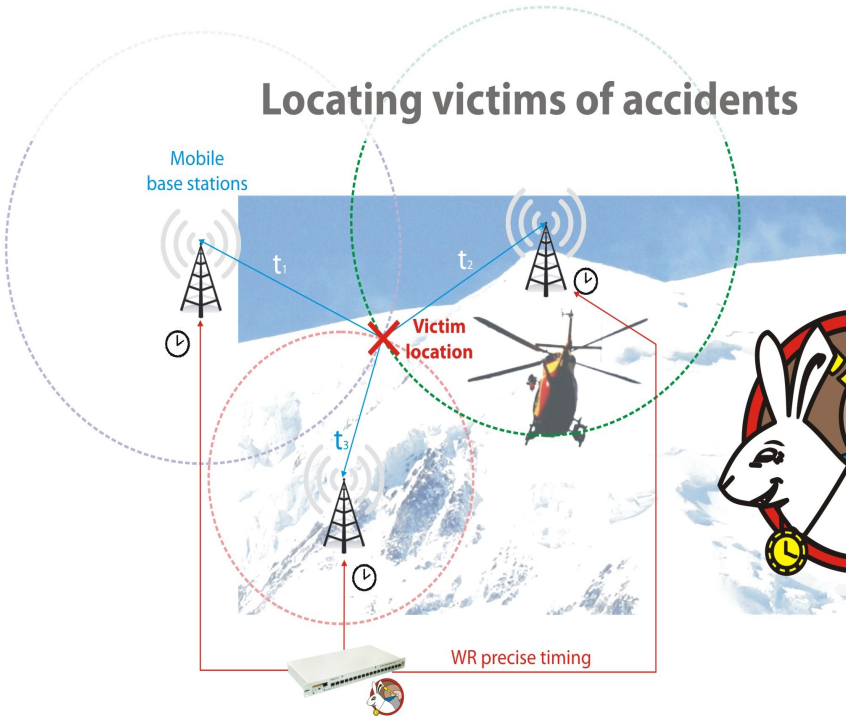
More WR collaborators/users:

<http://www.ohwr.org/projects/white-rabbit/wiki/WRUsers>



# Open source and the unexpected

## Locating victims of accidents



## Smart power grids

Power station synchronization

Phase & power measurements

The complex block contains several elements: a diagram of power station synchronization showing a network of power lines and stations; a photograph of a high-voltage power transmission tower; a map of Europe with a legend for different power lines; a cartoon rabbit wearing a yellow and blue striped hat and a gold medal; and a graph titled 'Phase & power measurements' showing two sine waves with a phase difference  $\theta$  and a power measurement  $P_{grid} - \xi_s$ .

# Why an open hardware licence?

Other licences don't work too well:

- Creative commons licences are maybe ok for simple documentation, but don't deal with patents.
- Software licences, like Apache and GPL don't use the right language, and don't necessarily cover the right IPRs (e.g. database right)
- There are domain-specific issues which open source software licences don't cover

# Why a reciprocal/copyleft licence?

- To ensure that designs remain free, and aren't able to be 'closed'
- To generate a pool – a 'commons' – of designs which can be freely shared and freely used
- But...we have now drafted a permissive version as well.



# “GPL for hardware”

Allows people to build on the design, but if they develop it, or combine it with other designs, and distribute it, or a product using it, they have to make the design materials available.

# EMACS General Public License

- (Apparently) the first copyleft licence
- A 'hack' on copyright.
- It's a condition of the licence, that if you distribute copyleft code, you need to make the source available, and license that code (and any modifications) under the *same* licence.
- Every time someone uses a bit of [E]GPL code in the code they distribute, that new code becomes [E]GPL as well.
- Developed into GPLv2, v3, LGPL, AGPL...

# Successful?

- The Linux kernel is licensed under GPLv2
- Contained in computers and devices everywhere, from every Android phone, to all of the top 500 supercomputers in the world.
- Red Hat sold for \$34Bn
- And that's only Linux

# Too successful?

## Attempts to use copyright to:

- Limit the use of software for 'good, not evil' (JSON)
- Limit working hours in Chinese factories (anti-996)
- Get free beer (Beerware licence)
- Extreme copyleft
- Prevent use of the software in a cloud context
- Careware
- Prevent the software from being TiVoised
- Prevent the software from being subject to DRM

# What people wanted from CERN OHL

Any hardware it touches also has to be licensed under CERN-OHL

As does any software included in it

To force people to print copyright notices on the hardware

To ensure that the designs of the distributed hardware are made public

To be compatible with GPLv3

To cover every sort of hardware from cars to planes to electronics to silicon  
chips to statues to beer

To guarantee that the hardware is easy to repair

To guarantee that the hardware is sustainable and green

To guarantee that the hardware is safe

To require distributors of the hardware to provide information about the  
success of the hardware

# A diversion – the MPL

- MPL version 1.1:

## **3.3. Description of Modifications.**

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

# A diversion – the MPL

- MPL v2.0



# Diversion – the MPL

“We didn’t include that clause in 2.0, because although it seemed like a good idea, almost no one actually complied with it, and it didn’t seem to add any value”



# In the CERN OHL

- V1.1 contained an obligation to provide information about distribution stats. That has now gone.
- V1.1 required a distributed design to be made public. That has now gone.

Golden Rule 1: don't include anything "just because it seems like a good idea"

# Make the licence understandable

- Short sentences
- “We” and “You” rather than “Licensee” and “Licensor”
- Steal concepts and definitions which are already well understood in the industry
- Make the licence tell a story – follow the narrative flow
- Avoid awkward and gendered language

Golden Rule 2: The licence tells a story. Make it clear, understandable, and as uncontroversial as possible.

# Shamelessly Steal

- Patent clauses from Apache 2.0
- The concept of a NOTICES file from Apache 2.0
- Termination and “Convey” and “Complete Source” from GPLv3
- Structure from MPL v2

Golden Rule 3: Good drafters  
borrow, great drafters steal\*

\*shamelessly stolen from Pablo Picasso\*\*

\*\*which he stole from TS Eliot

# Don't try to be too clever

- An earlier draft of V2 contained an ‘troll dissuasion clause” intended to force trolls to negotiate for compliance before suing for damages.
- Eben Moglen, rightly, told us this wouldn't work, so we removed it.

# Golden Rule 4: Don't try to be too clever.

\*Corollary: Eben Moglen is [almost] always right



# Some things are impossible

Common request: make the licence GPLv3 compatible:

- Which means a race to the bottom of “least restrictive”
- CERN OHL has extensive attribution requirements geared to hardware unlike GPLv3
- An original licensor would not necessarily be happy to lose this.
- If they are, they can dual license.

Golden Rule 5: Don't try to do the  
impossible

# Don't Be Too Domain Specific

- We have had a lot of input from FPGA and ASIC designers who have domain specific requests about coexistence with proprietary libraries.
- An earlier draft of V2 contained a great deal of HDL/FPGA/ASIC-specific language and even some optional add-on clauses. It became far too complicated.

Golden Rule 6: Don't get bogged  
down on a specific use case

# We ripped up an earlier draft of V2 and started from scratch

- The earlier version was too long, too complex, used unwieldy language.
- We did the bulk of the redraft in an intensive 2 day session.

Golden Rule 7: If you need to, rip it up and start again.

# The Boundary Problem

- With copyleft software licences, it can be difficult to determine how far the copyleft effect extends.
- Does it affect plugins? Dynamic linking? Static Linking? System libraries? Subclassing?
- With hardware, it's even more complex.

# Software...

...is made out of two raw materials. 0s and 1s.  
And you have an infinite supply of both, for  
free.



# Hardware...

...is made out of a periodic table full of elements, plus information (0s and 1s). The raw material 0s and 1s may be free and infinite in supply, but elements aren't.

..and we don't yet have a "compiler for hardware" which is cheaply and readily available

Atomic force microscopes don't count. Star Trek replicators don't yet exist.

# Hardware

- Is made out of components. (Even 3d printed objects need feedstock).
- So how far down the rabbit hole do we go?
- Do we need the complete instructions to make resistors, capacitors, bearings, screws, nuts, bolts..... etc. out of atoms?

# Introducing Available Components

Where a design consists of 'Available Components' then the full design files ("Source") of those components does not need to be provided, so long as they are:

- Readily available (including at a cost); and
- You have (or can readily obtain) all the rights specifications and interfacing information you need.

# Available components...

...can be nested at different levels.

- Datacentre
- Rack - network, power cabling
- Enclosure/PSU
- Circuit board
- Component - resistor, capacitor, FPGA
- Bitstream
- Custom code, standard libraries, third party libraries

# Available components...

- ..are possibly a hack that's too clever for its own good. So we stress tested the concept with
- experts in the field of open source software and hardware
  - Silicon/HDL/ASIC/FPGA designers
  - Mechanical hardware designers
  - Software architecture experts (yes, it seems to work well for containers: better than many OSS licences...).

Golden Rule 8: If you think you're  
being too clever, stress test your  
idea with real world scenarios, and  
experts\*

\*Corollary: Eben Moglen is [almost] always right

# Nice to haves..

...may make more sense in the FAQs

- Don't underestimate power of community norms.  
We can use them to:
  - Help clarify the scope of what counts as Complete Source
  - Encourage people to use standardised component IDs for Available Components (e.g. DOIs, although Software Heritage may have a better idea)

Golden Rule 9: Community norms are powerful. Consider using FAQs instead of the licence text.



# Remember your existing users

- There is already a significant user base of CERN OHL 1.x
- Users may have chosen the ‘or later’ option

*“CERN may publish updated versions and variants of this Licence which it considers to be in the spirit of this version, but may differ in detail to address new problems or concerns.”*

- Version 2.0 needs to respect this.
- CERN-OHL 1.x or later adopts CERN-OHL-S (strong). We also have CERN-OHL-L (lesser) and CERN-OHL-P (permissive)

Golden Rule 10: Remember your  
existing users

# Golden Rules

- 1: Don't include anything "just because it seems like a good idea"
- 2: The licence tells a story. Make it clear, understandable, and as uncontroversial as possible
- 3: Good drafters borrow, great drafters steal
- 4: Don't try to be too clever
- 5: Don't try to do the impossible
- 6: Don't get bogged down on a specific use case
- 7: If you need to, rip it up and start again
8. If you think you're being too clever, stress test your idea with real world scenarios, and experts
9. Community norms are powerful. Consider using FAQs instead of the licence text
10. Remember your existing users

Thank You.

Any questions?

# Thank You.

For more information see Licences, FAQ  
and commentary available on [ohwr.org/  
projects/cernohl/wiki/cern-ohl-v2-draft](http://ohwr.org/projects/cernohl/wiki/cern-ohl-v2-draft)

Thanks to CERN and Javier Serrano for some of the content and  
images in this presentation.