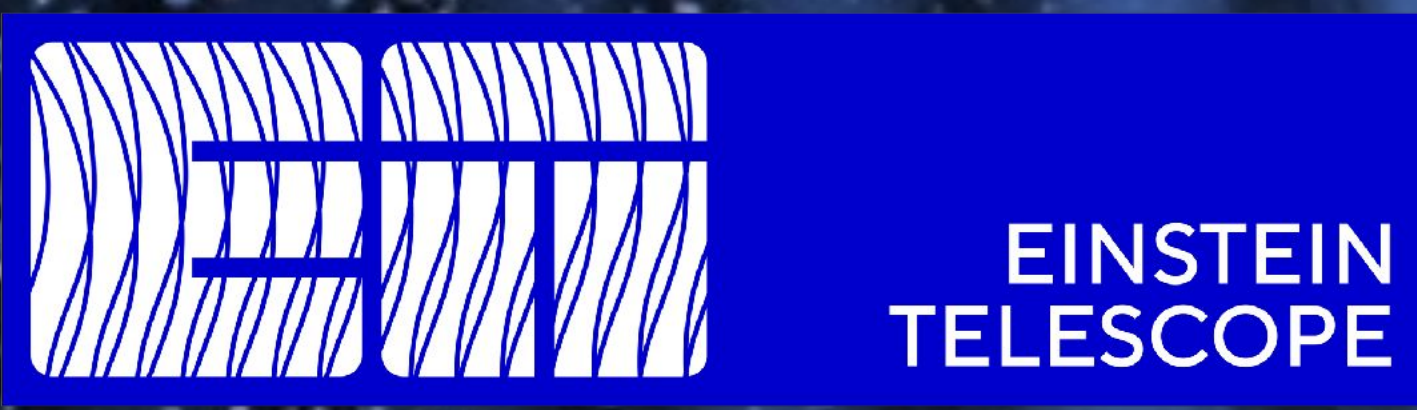


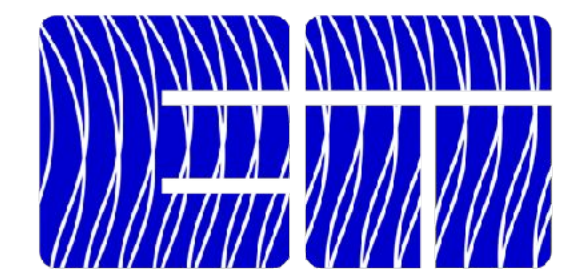
Einstein Telescope Computing and Software Challenges 101



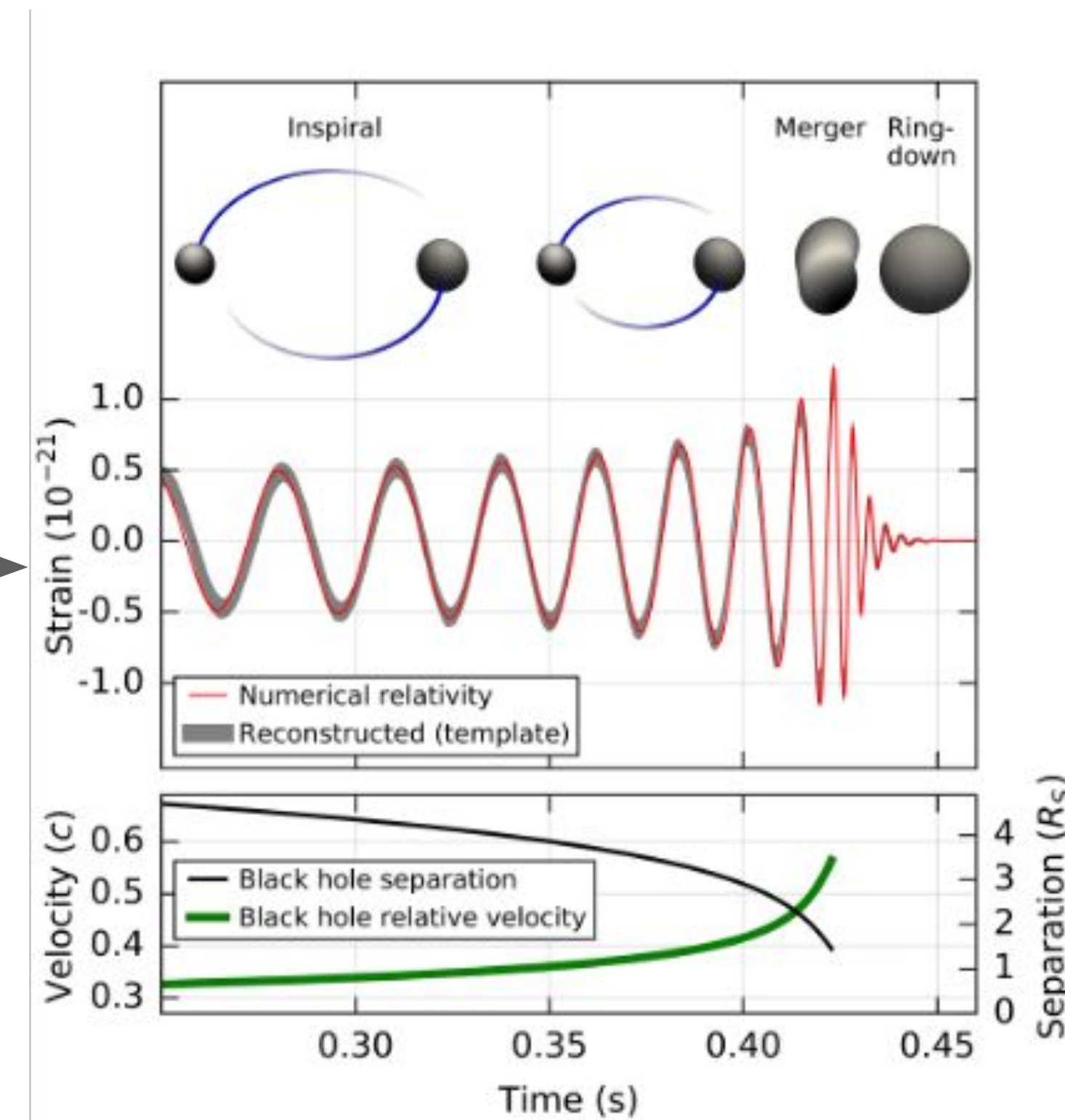
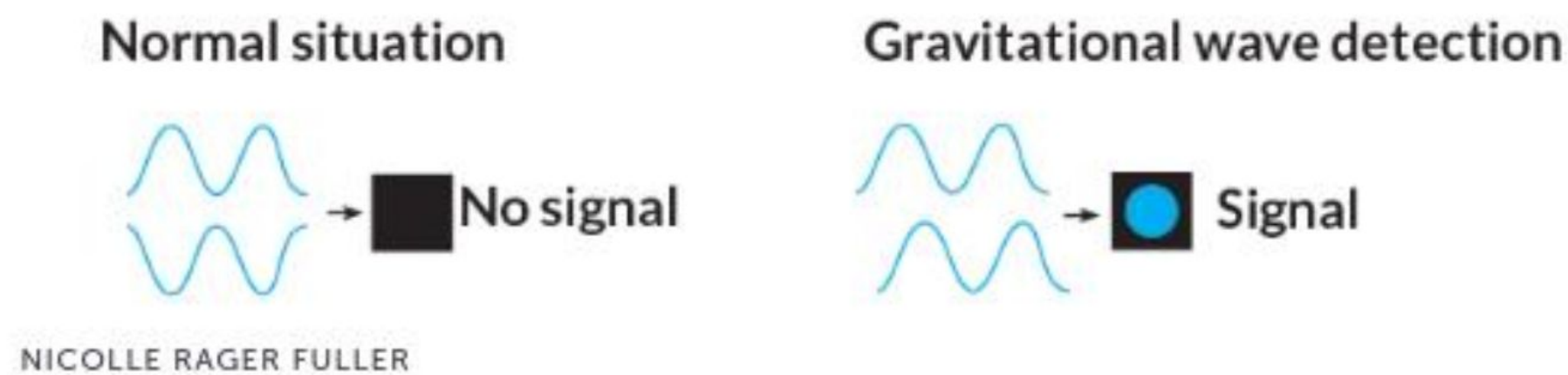
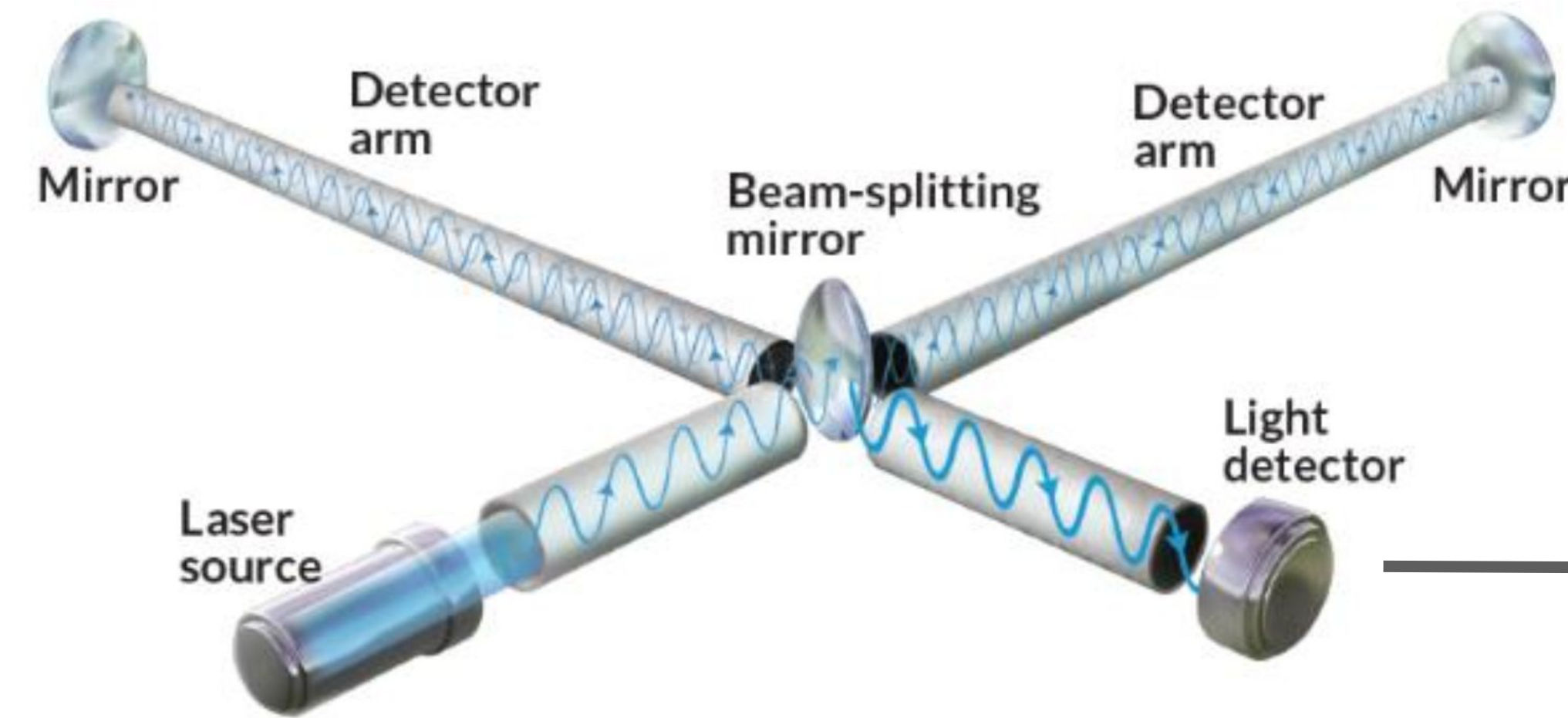
EINSTEIN
TELESCOPE

Paul Laycock - *April 14th 2026*
ET Industry Computing workshop, Nikhef

GW detection and GW data



EINSTEIN
TELESCOPE



- Strain data, $h(t)$, is the time-series needed for analysis, recorded at **16 kHz**
- *This will be the same for **Einstein Telescope** as it is today for **LVK***
 - **Computing power needed scales ~number of events - a factor of 1000 using today's s/w**
 - **Assume a sister detector in the US - crucial to share data at low latency**

ET computing model

Contents

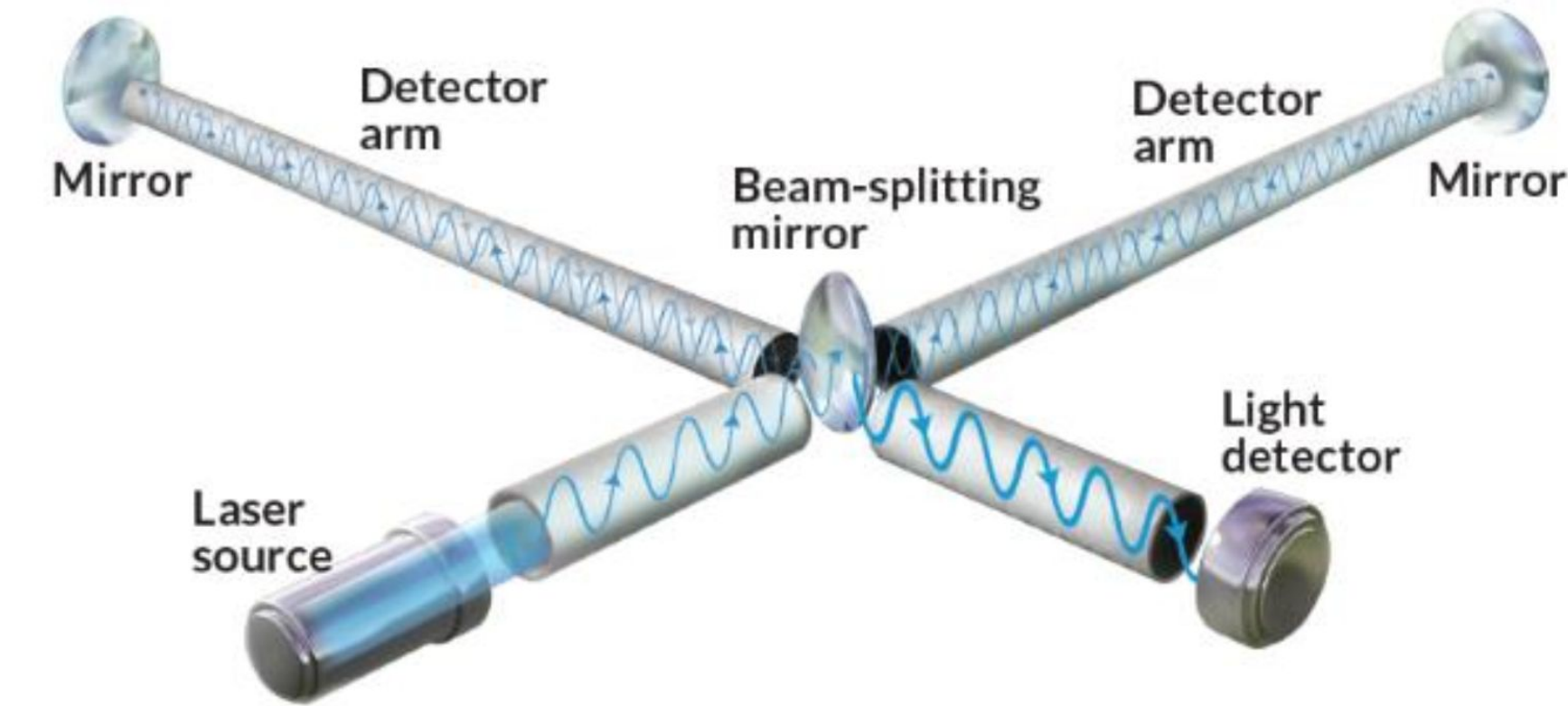
I Introduction	9
1 Scope	10
2 ET Science Goals	13
3 Gravitational Wave Detection	18
3.1 Basic principle of GW detectors	18
3.1.1 Interferometers	18
3.1.2 Data Output	20
3.2 Analysis of gravitational wave signals	21
3.2.1 Gravitational wave events	21
3.2.2 Analysis methods	24
3.3 Gravitational Wave Alerts	27
4 Computing Model Design Principles	30
4.1 FAIR Principles	31
4.1.1 A FAIR implementation for the Einstein Telescope	31
4.1.2 Beyond FAIR	33
4.2 Computing Tasks and Resource Classification	34
4.2.1 On-site Computing	34
4.2.2 Off-site Computing	35
4.3 Software management	36
4.3.1 Software Best Practices and Metrics for Success	36
4.3.2 Organisational Principles	37
4.3.3 Critical Engineering Practices	37
4.3.4 Metrics to Track (DORA)	38
4.3.5 Release management	38
II Core Mission	40
5 On-site Computing	41
5.1 DAQ	41
5.1.1 Main data streams	41
5.1.2 Supervisory control and data acquisition (SCADA) system	42

- **Main question:** Can we analyse data that contains 1000 times more events, including long-lasting and overlapping signals?
- **Main strategy:** Build on the experience of **LVK**, leverage expertise particularly from **CERN**
 - Much of **LVK** computing relies on the “Open Science Grid” ([OSG](#)), used to provision resources for the CERN experiments ([WLCG](#)) - tools like [HTCondor](#) and [Rucio](#)
 - For ET, there are also synergies with [EOSC](#) / [ESCAPE](#)
- **Main challenges:** There is no formal **LVK** Computing Model document

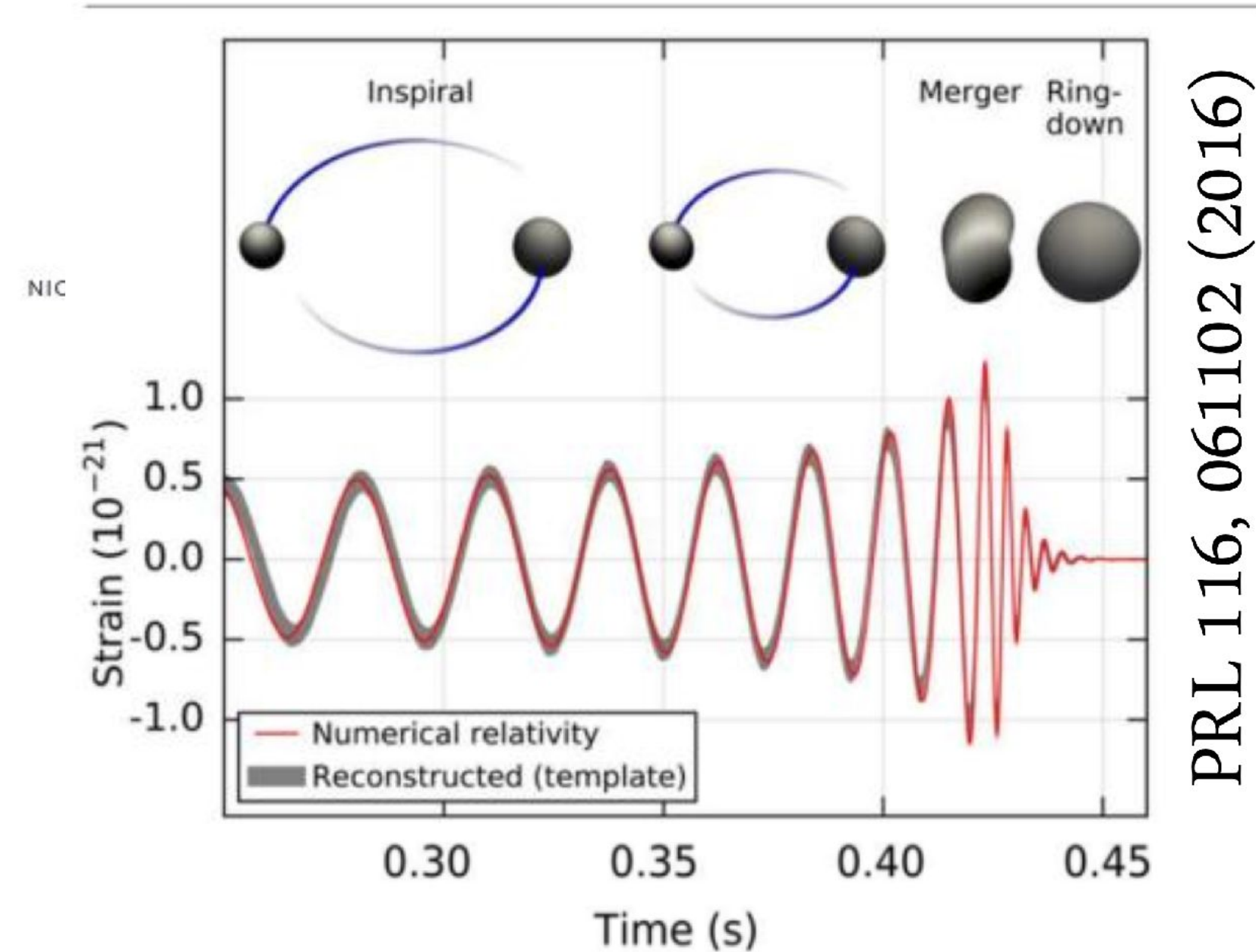
Data Challenges for Einstein Telescope



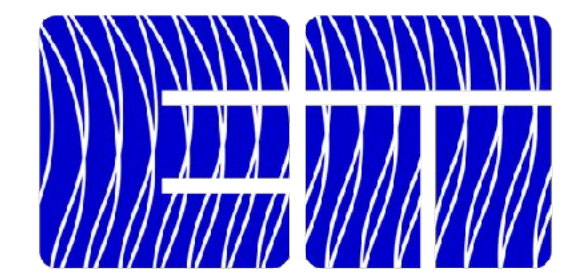
EINSTEIN
TELESCOPE



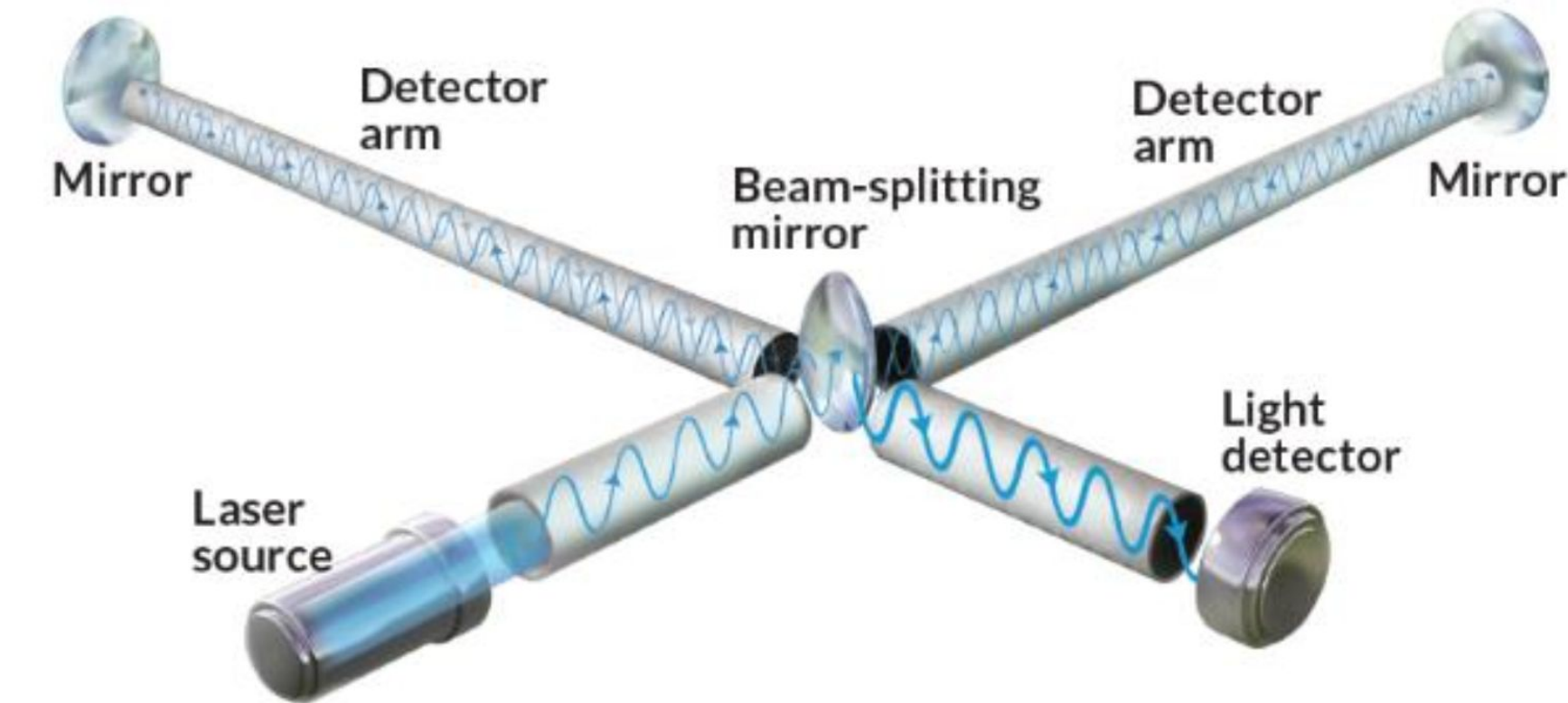
- Strain data, $h(t)$, is one *time series sampled at 16 kHz*, but ET will have more auxiliary channels than Virgo (cryogenics, advanced detectors)
- Over-estimate **~100 PB/year** of raw detector data (very many auxiliary channels)
- This is significant, but largely for archive
- To first order, most analysis only needs $h(t)$
 - about **10-100 TB/year**
 - Plus data quality and some metadata



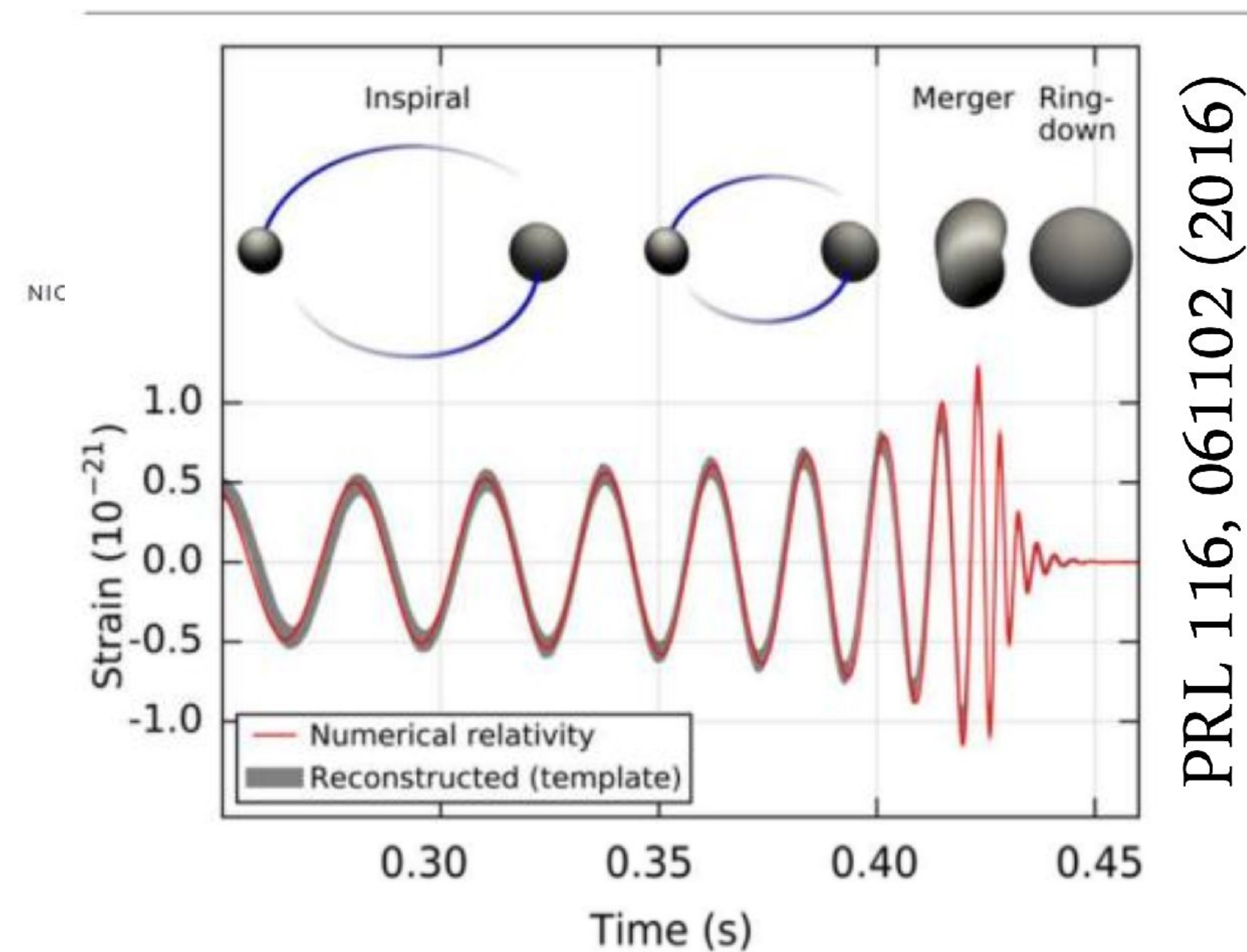
Compute Challenges for Einstein Telescope

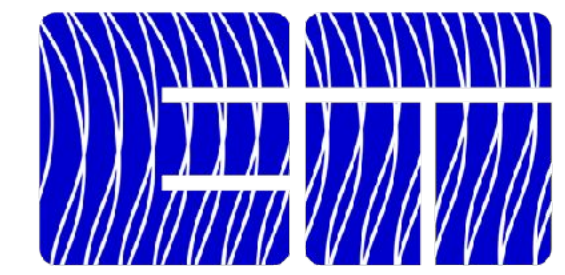


EINSTEIN
TELESCOPE



- **Low latency computing** will deal with 1000 times more events, some of which will overlap
- It currently takes ~30s to analyse data promptly and generate alerts using **~20k cores + ~500 GPUs**
- Increased signal sensitivity for **ET** means signals are also in-band for much longer (minutes, hours, days)
- A huge deal for multi-messenger science!
- *Using today's s/w, ET could need 40M cores just for low latency - this will NOT be what ET will use!*
- **AI/ML and GPUs are being aggressively pursued, see the [Science Blue Book](#)**





EINSTEIN
TELESCOPE

Paul Laycock



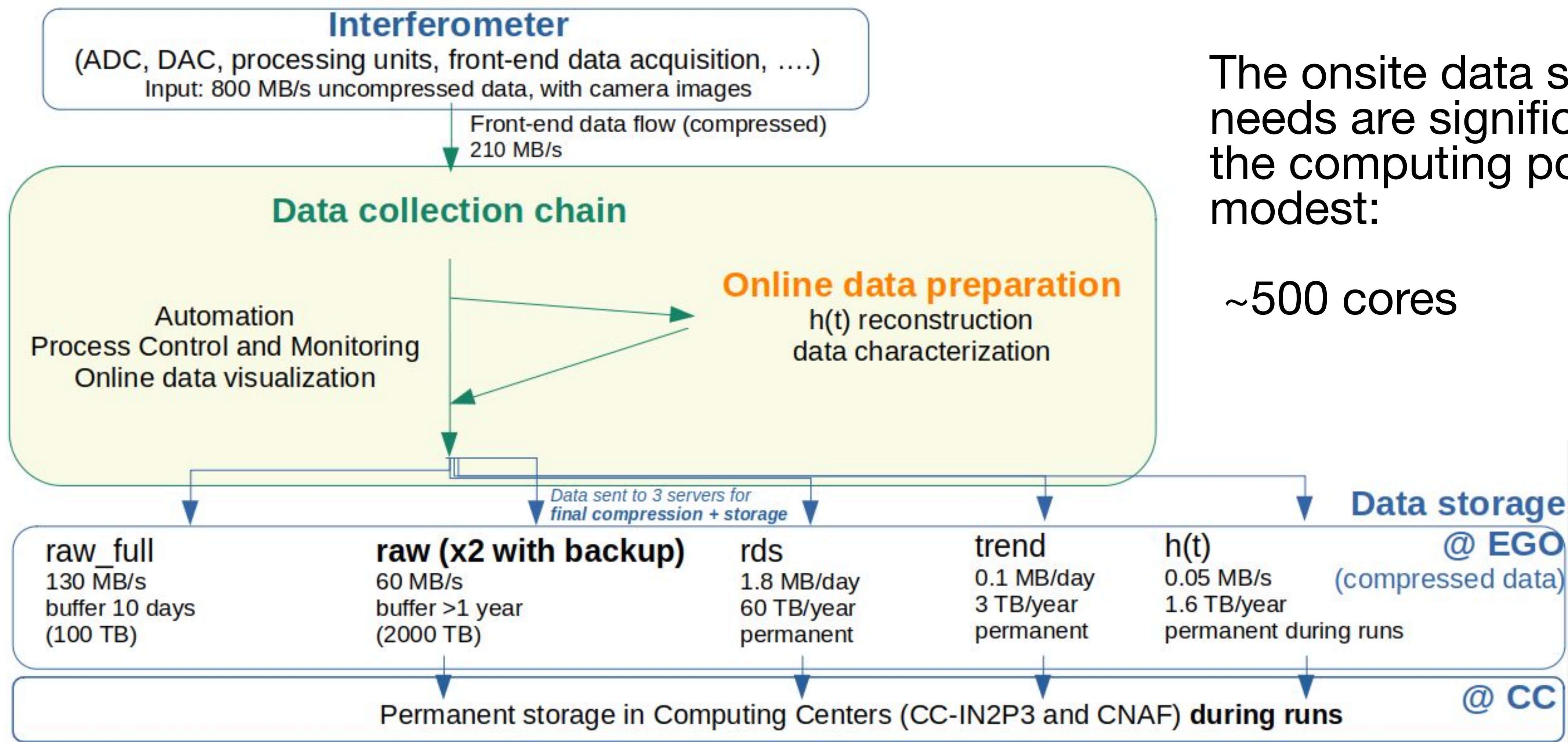
GWSC

GRAVITATIONAL
WAVE
SCIENCE
CENTER

Onsite computing for Virgo



EINSTEIN
TELESCOPE



The onsite data storage needs are significant, while the computing power is modest:

~500 cores

Scaling data rates to ET

Data rate	DAQ	Raw_full	Raw	RDS	Trend	Strain	DetChar
MB / s	250	130	2x60	1.8	0.2	0.05	0.2
TB / day	20.5	10.7	2x4.9	0.15	0.016	0.004	0.016
PB / year	7.4	3.8	2x1.8	0.05	0.006	0.001	0.006

Table 2: Data rates for the **Virgo** interferometer before the O4 run, “DAQ” is the bandwidth used by the entire DAQ system, while the subsequent columns correspond to data formats described in the text. Note that two full copies of the “Raw” data have been accounted for, while for Virgo the second copy currently only covers 6 months.

Data rate	DAQ	Raw_full	Raw	RDS	Trend	Strain	DetChar
MB / s	1250	650	2x300	9	1	0.25	1
TB / day	104	54	2x25	0.75	0.08	0.02	0.08
PB / year	37	19	2x9	0.26	0.029	0.007	0.029

Table 3: Data rates per **ET** interferometer.

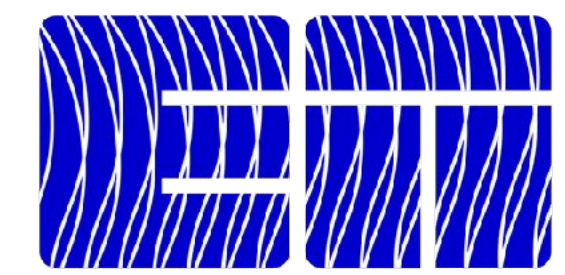
The scaling model *per interferometer* is

$$ET = \text{Virgo} * 5$$

- This accounts for more auxiliary channels and higher sampling freq.
- The final number will depend on geometry,

We think this is very conservative

Onsite Computing needs for ET



EINSTEIN
TELESCOPE

	Minimal scenario	Operational safety margin
Operations storage buffer (TB)	800	1600
Long-term storage (PB)	20	40
CPU cores	2150	6450
RAM (GB)	6300	18900
Network	100 Gb/s	2 * 100 Gb/s

Table 7: Baseline online computing requirements per ET interferometer for data storage capacity, processing power, RAM memory and network speed.

The scaling model per IF is

$$ET = \text{Virgo} * 5 \text{ (storage)}$$

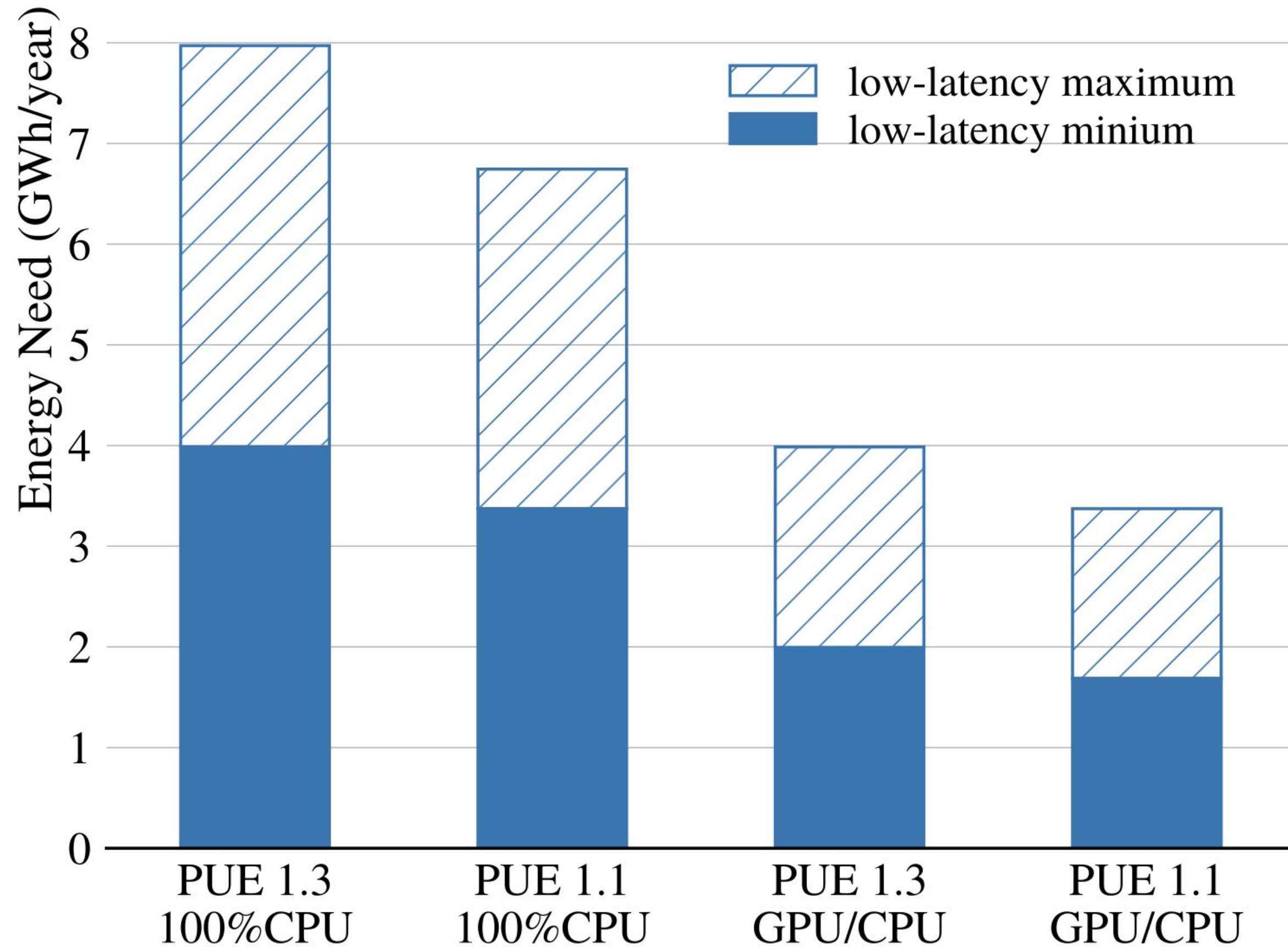
$$ET = \text{Virgo} * 3 \text{ (CPU and RAM)}$$

The number depends on geometry, and allows for more auxiliary channels and higher sampling freq.

We think this is very conservative

While the computing power needed is modest, there is still a relatively large amount of data needed onsite and, particularly during commissioning, an efficient data reduction strategy will be a priority

ET Computing - energy footprint



In the ET computing model, we present an estimate of the energy footprint of ET low latency computing, which is expected to dominate

This follows the methodology used by the WLCG for worldwide computing needs for CERN experiments (beyond the CERN sites)

Power Usage Efficiency (PUE) makes a big difference, as do strategies that leverage GPUs

Software development best practice



EINSTEIN
TELESCOPE

Continuous Integration Example – *Integration Test*

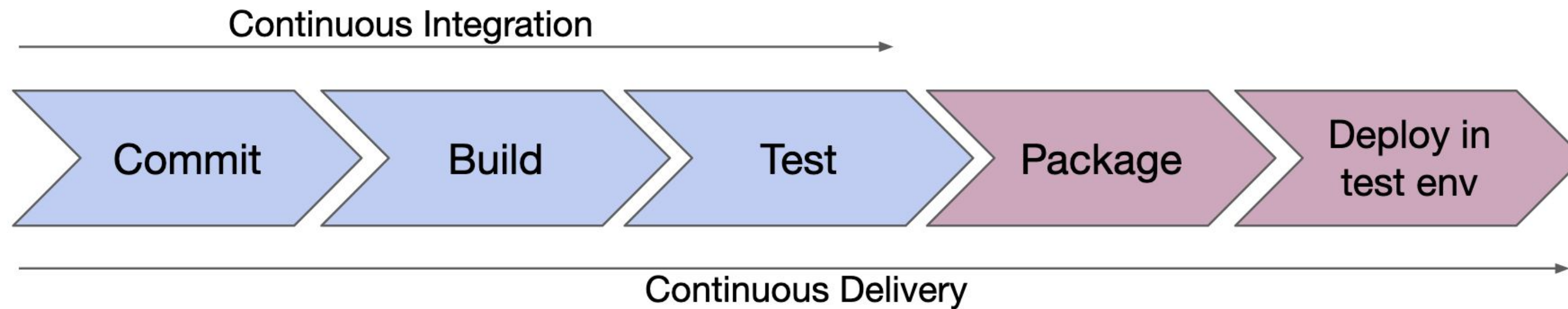
```

Integration Test

def test_upload_to_local_gracedb(self):
    # Check how many events are in GraceDB before upload
    events_before_upload = get_number_of_events()

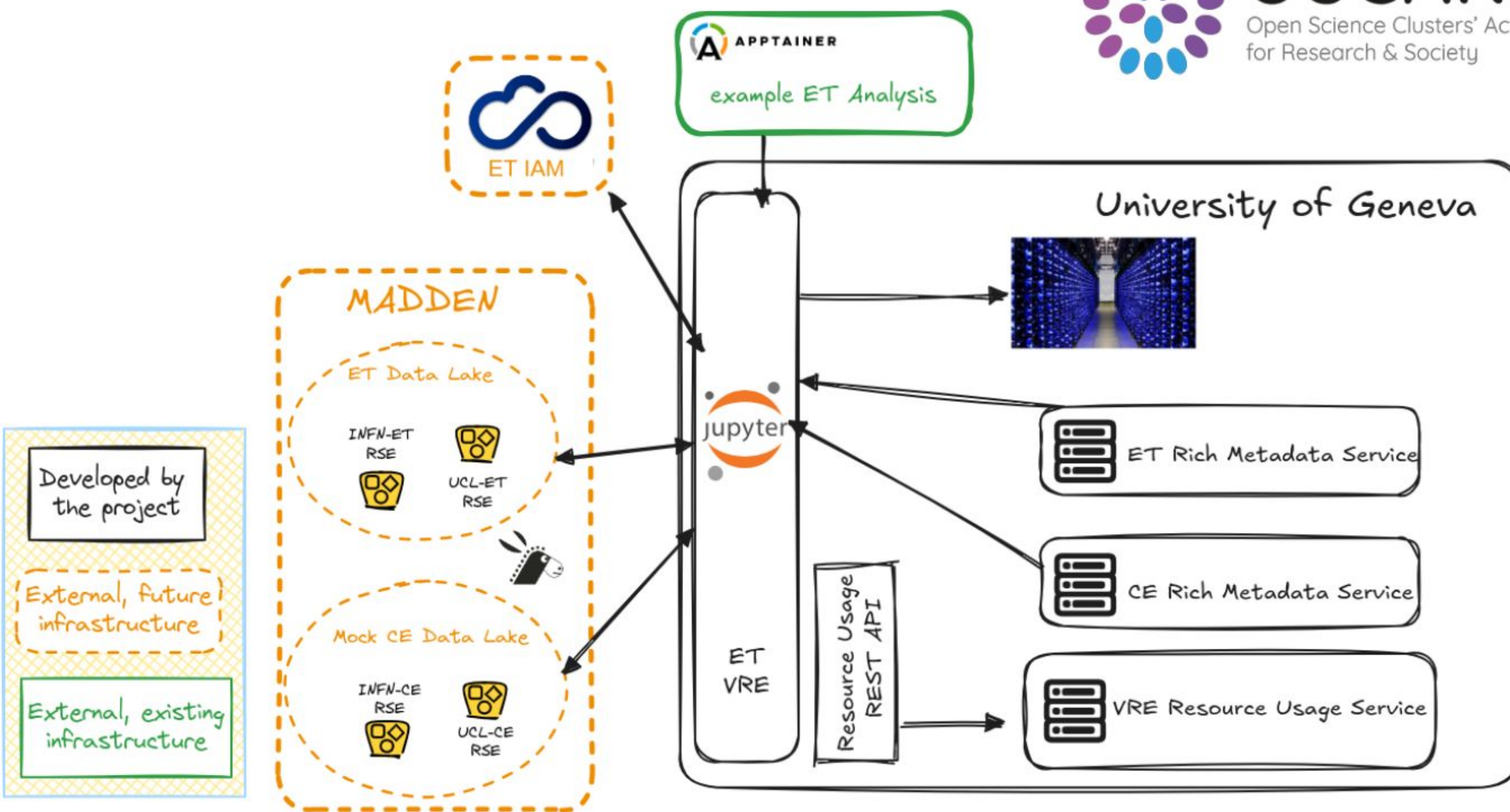
    # Generate and upload event
    event = generate_candidate_event()
    upload_candidate_event(event)

    # Check & assert that there is a new event in GraceDB after upload
    events_after_upload = get_number_of_events()
    assert events_after_upload == events_before_upload + 1
    
```



ET software must be made to follow software development best practices

ET must provide intelligent, high-level interfaces like the [Einstein Telescope Analysis Portal](#), being developed in collaboration with CERN (funded by [EOSC/OSCARS](#) cascading grants)



Latency and computing resources



EINSTEIN
TELESCOPE

Latency

On-site dedicated resources

- Lowest latency
- Full control of hardware specification
- Operation is our responsibility

We require some resources that are completely dedicated to ET, e.g. DAQ

Off-site dedicated resources

- <1s latency of network to centre(s)
- 1s - 1min latency to start ET jobs
- ET defines hardware requirements
- Operation is centre(s) responsibility

LVK strain data is distributed (< 1s) to different sites to look for GW event candidates, we will do the same for ET
Jobs must start 1s < 1 min of receiving strain data

Off-site fixed allocations

- <1s latency of network to centre(s)
- 1min - 10min latency to start ET jobs
- ET defines hardware requirements
- Operation is centre(s) responsibility

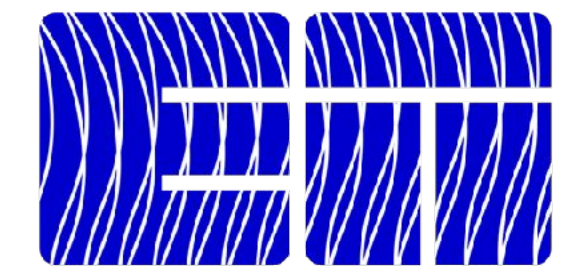
Some post-processing (enrichment) for GW event candidates takes hours, added value but not urgent
Jobs should start 1min-10min of receiving strain data

Shared resources

- Latency depends on current load
- Pledged resources on an annual basis
- ET suggests hardware requirements
- Operation spread across providers

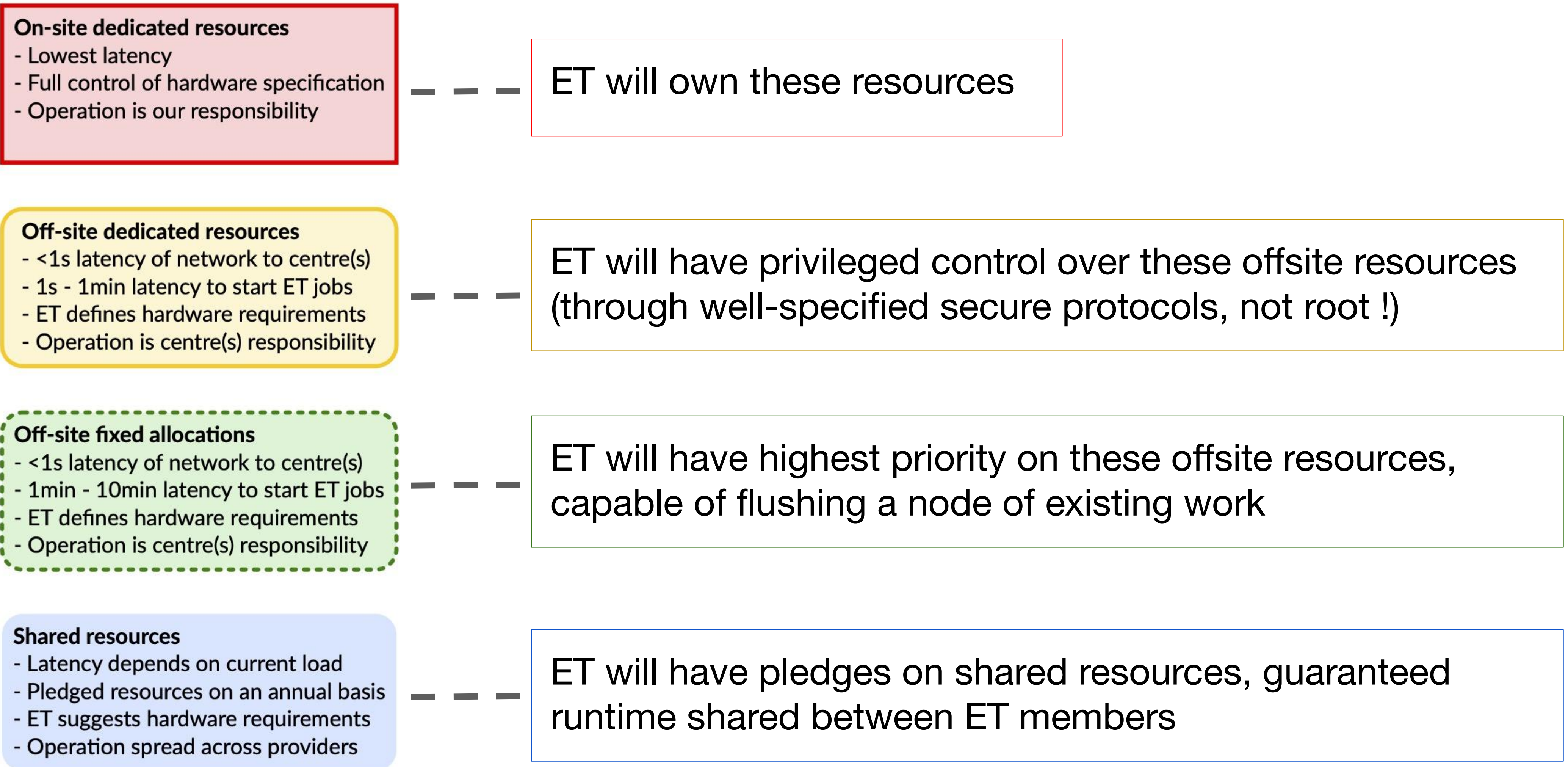
Data analysis of strain data, by collaboration members and later by the whole astro community
~asynchronous wrt availability of strain data

Latency and computing resources

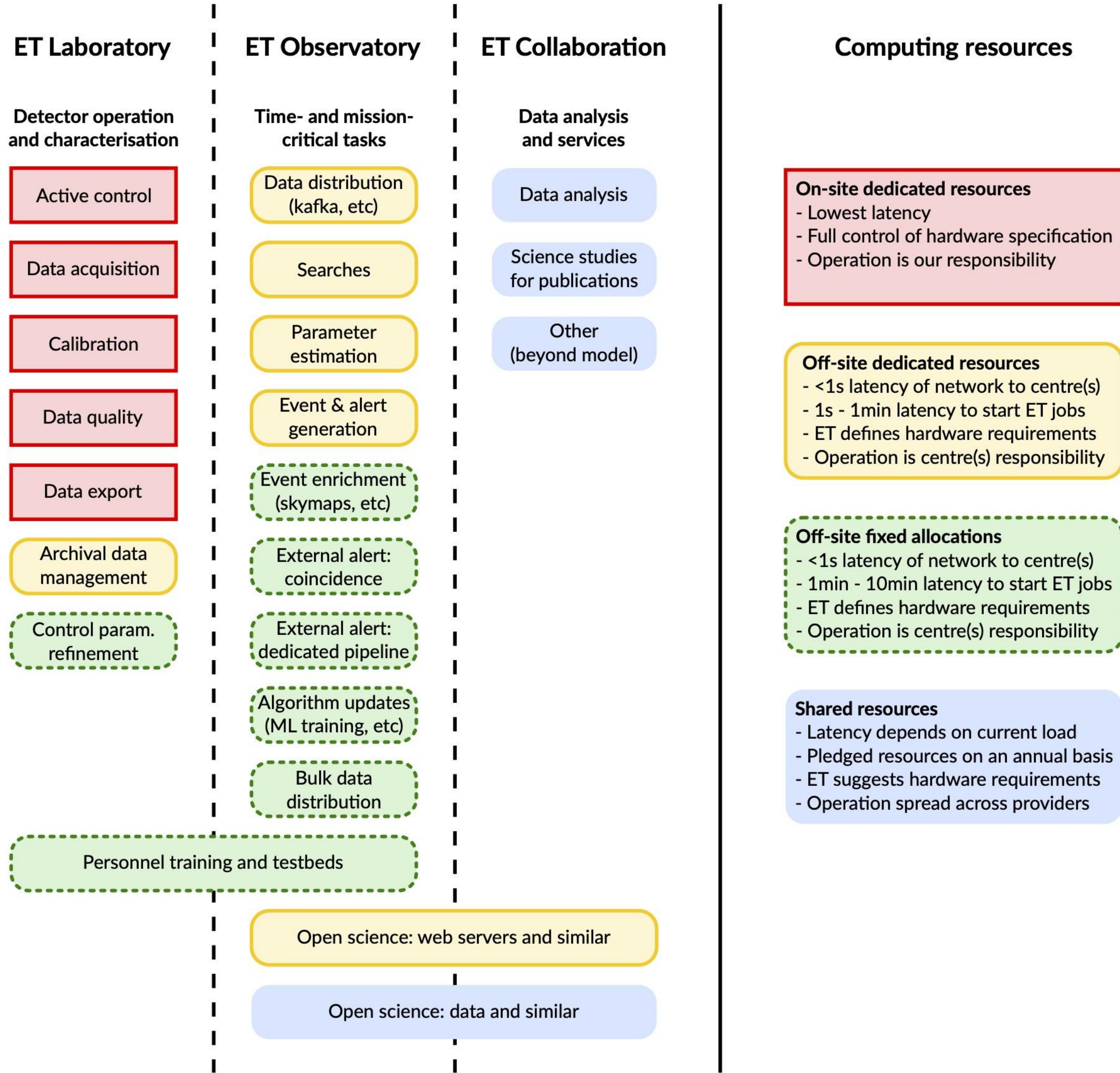


EINSTEIN
TELESCOPE

Cost



The ET Computing tasks



An important caveat, the ET Computing Model implementation will depend on the site decision

- The ET Computing Model concept defines the requirements
- There can be good reasons to minimise the number of computing sites, particularly for high priority tasks

In this talk, and the document, “Onsite computing” refers to things that are strictly ***required to be at detector site(s)***