AMPEL

repeatable, `scalable`, modular
analysis of data streams

- **Use cases**
- **Motivation and methodology**
- **[How does it work]**

Jakob Nordin (HU Berlin) - NFDITalk Oct 21 2024

Particles, Universe, NuClei and Hadrons for the NFDI

PUNCH 4NFDI

42 Partners
Representing close to 10000 scientists in Germany
(KAT, KET, KHuK, RdS)

AMPEL sample use cases

Systematic execution of analysis units while taking care of provenance, data storage & logging.

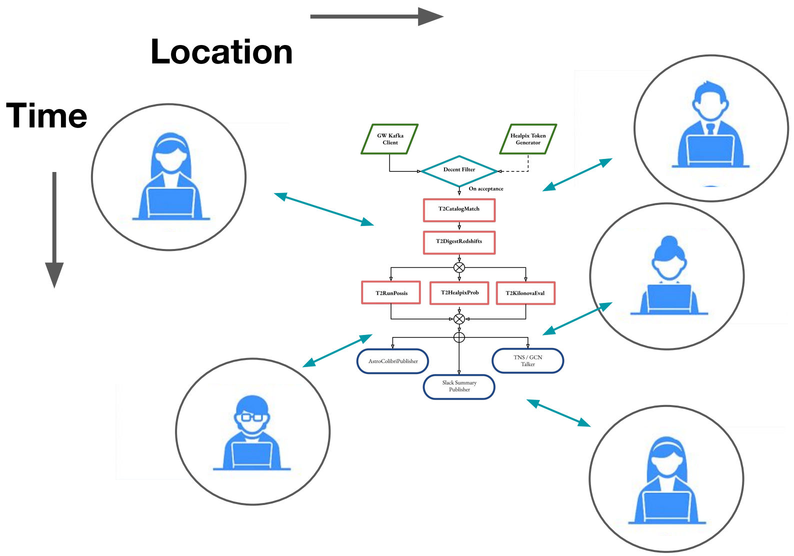
- **Real-time stream analysis**
- Modular software development
- Heterogeneous input sources
- Fine grained provenance
- Analysis variants of same data
- Code-to-data in science

Image: https://dbconvert.com/blog/data-stream-processing/

AMPEL sample use cases

Systematic execution of analysis units while taking care of provenance, data storage & logging.

- Real-time stream analysis
- **Modular software development**
- Heterogeneous input sources
- Fine grained provenance
- Analysis variants of same data
- Code-to-data in science



Icons: pluspng.com

AMPEL sample use cases

Systematic execution of analysis units while taking care of provenance, data storage & logging.

- Real-time stream analysis
- Modular software development
- **Heterogeneous input sources**
- Fine grained provenance
- Analysis variants of same data
- Code-to-data in science

?

Images: Wikipedia

AMPEL sample use cases

Systematic execution of analysis units while taking care of provenance, data storage & logging.

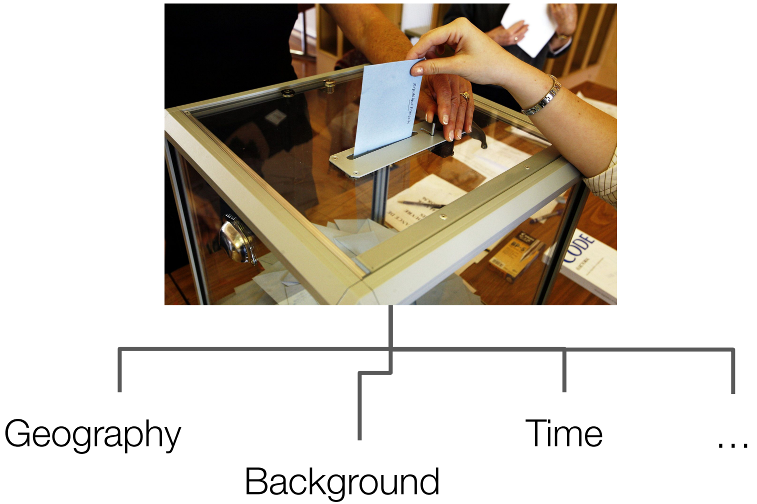
- Real-time stream analysis
- Modular software development
- Heterogeneous input sources
- **Fine grained provenance**
- Analysis variants of same data
- Code-to-data in science

Images: Wikipedia

AMPEL sample use cases

Systematic execution of analysis units while taking care of provenance, data storage & logging.

- Real-time stream analysis
- Modular software development
- Heterogeneous input sources
- Fine grained provenance
- **Analysis variants of same data**
- Code-to-data in science



Images: Wikipedia

AMPEL sample use cases

Systematic execution of analysis units while taking care of provenance, data storage & logging.

- Real-time stream analysis
- Modular software development
- Heterogeneous input sources
- Fine grained provenance
- Analysis variants of same data
- **Code-to-data in science**

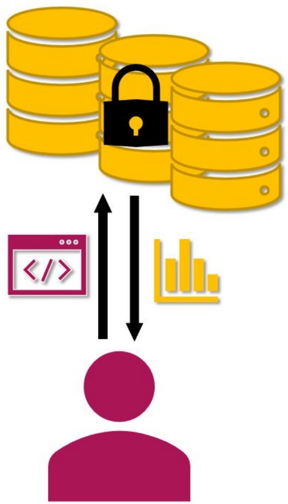
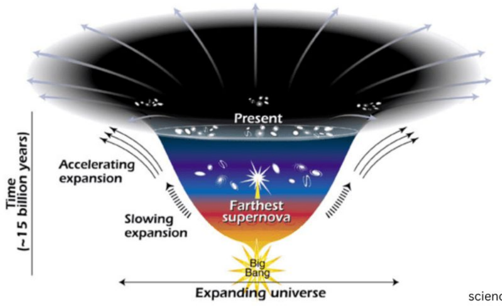


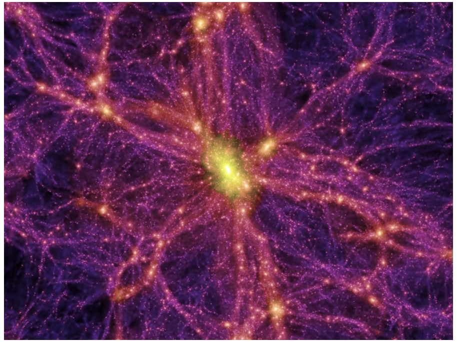
Image: Utrecht Uni.

Astrophysics



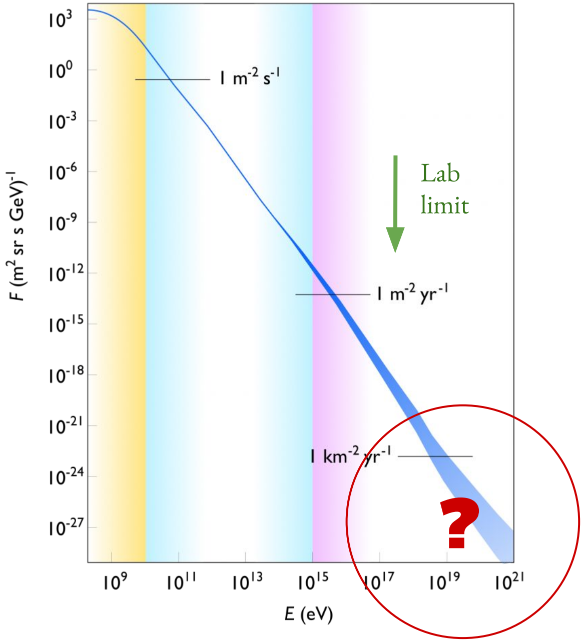
sciencenotes.org

Dark Energy - accelerated expansion



General Relativity - when does it break?

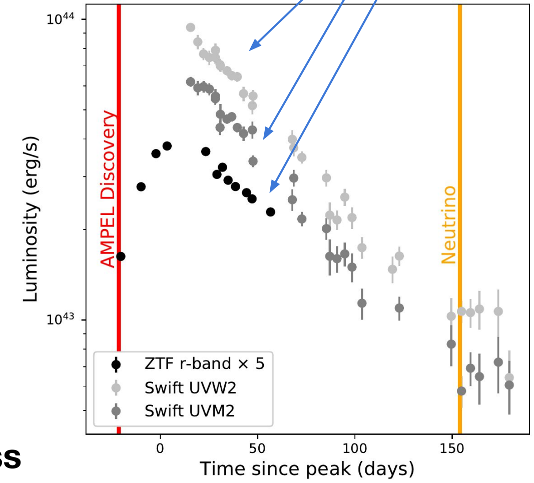
New energy regime



Particles hitting earth (energy)

Astronomical transients

Different detectors



Brightness

Time

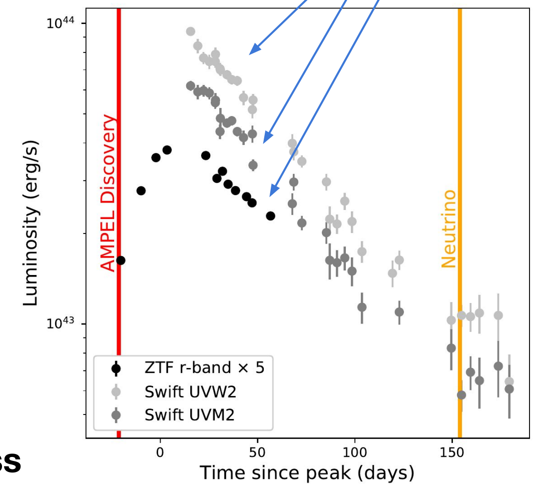
Astronomical transients

Tidal Disruption Event

A star disrupted by a nearby black hole

Different detectors

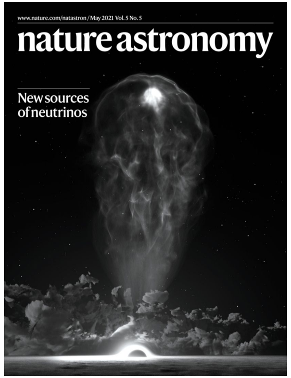
Brightness



Time

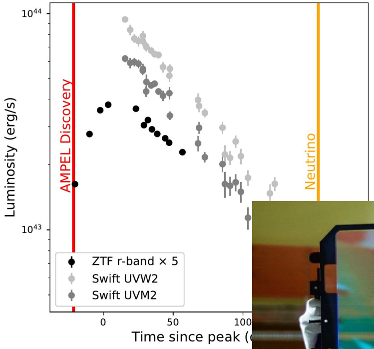
Test of:

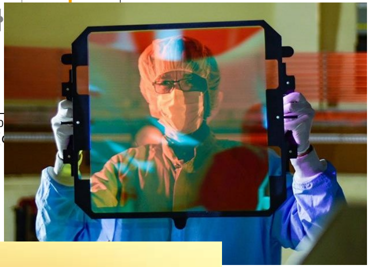
- General Relativity
- Structure formation
- Neutrino emission
- [Dark Matter / Energy]



https://doi.org/10.1038/s41550-020-01295-8
https://doi.org/10.1103/PhysRevLett.128.221101

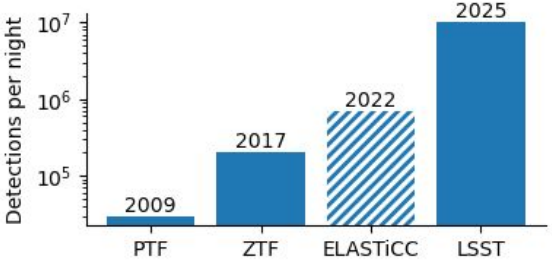
Astronomical transients



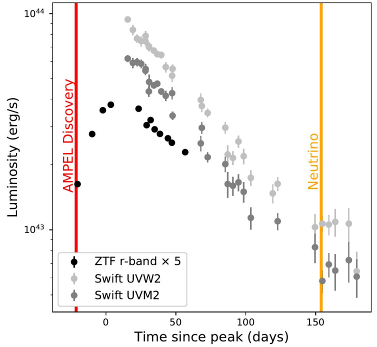


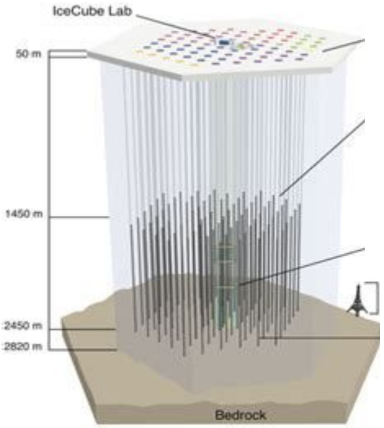
Needle in a haystack:

- Searching among billions of galaxies - every night.
- Multiple telescopes / streams.
- Yield millions of real but “boring” astronomical variables.
- Gravitational waves & neutrinos.



Astronomical transients

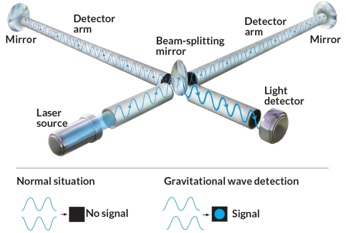




IceCube neutrino observatory.

Needle in a haystack:

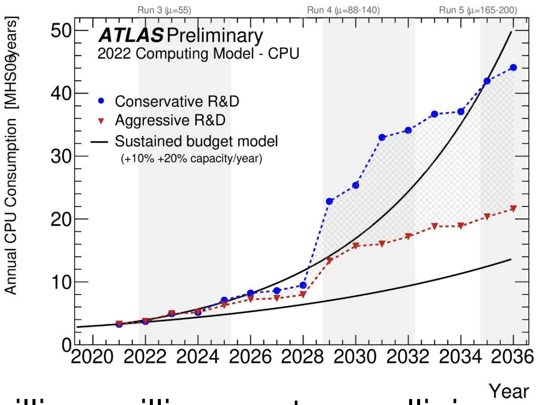
- Searching among billions of galaxies - every night.
- Multiple telescopes / streams.
- Yield millions of real but “boring” astronomical variables.
- Gravitational waves & neutrinos.



LIGO/Virgo Gravitational Wave telescopes.

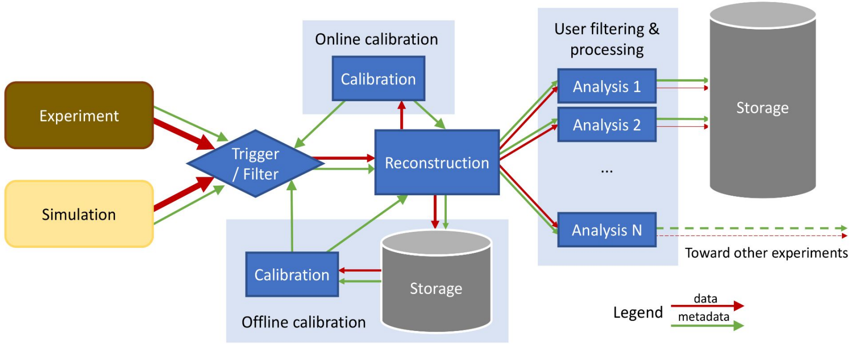
PUNCH4NFDI outlook

Future radio telescopes produce 157 TB/s



250 million million proton collisions / yr @ LHC

Methods for managing data irreversibility



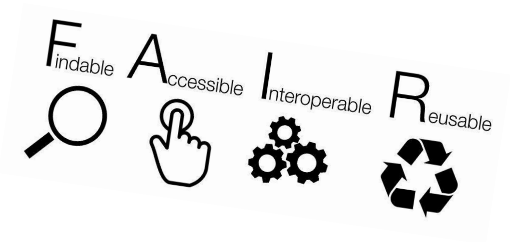
https://zenodo.org/records/10692169

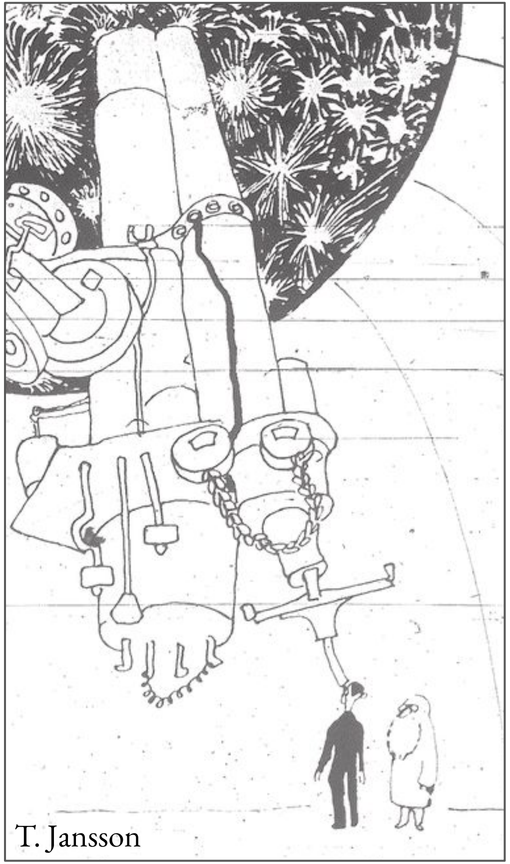
The promise:

Sensitive real-time detectors allow us to use the Universe to test fundamental physical laws.

The problem:

These create high throughput data streams that are too large for scientists to use for scientific exploration.





AMPEL

AMPEL is a

{methodology, workflow manager, real-time system}

where:

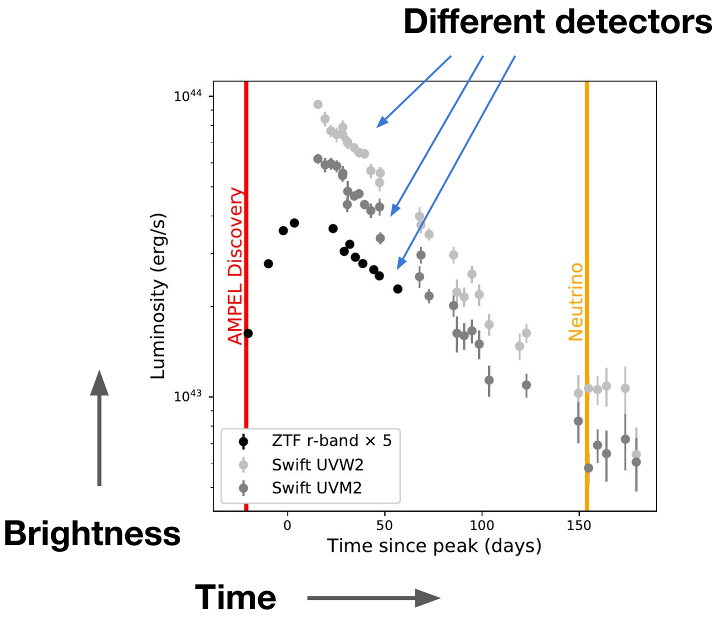
- **Scientists develop analysis algorithms.**
- **Instrument specialists design input feeds.**
- **Software designers create a scalable system.**

Provenance and reproducibility is built in from the start.

1 Methodology: **Deconstructing scientific workflows.**

Deconstruct workflow into *types* of operations:

- **Add**: Select measurements from detectors.
- **Combine**: Aggregate knowledge regarding one object at some time.
- **Augment**: Derive values, e.g. TDE probability.
- **React**: Real-time request for detailed observations.

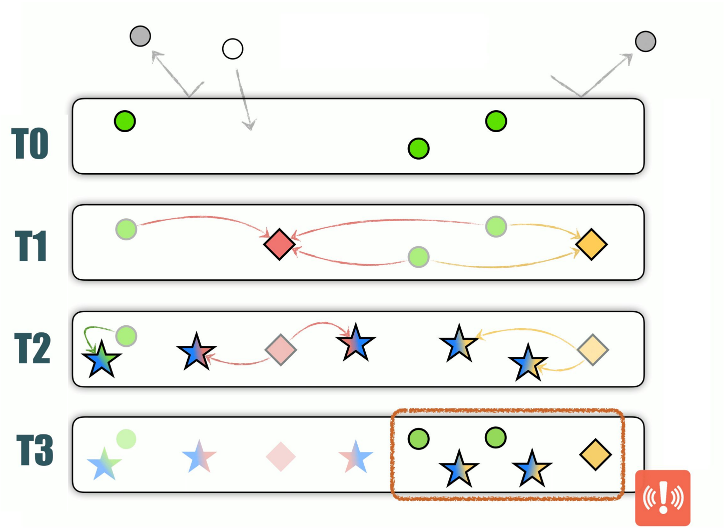


1

Methodology:
Deconstructing scientific workflows.

Deconstruct workflow into *types* of operations:

- **Add**: Select measurements from detectors.
- **Combine**: Aggregate knowledge regarding one object at some time.
- **Augment**: Derive values, e.g. TDE probability.
- **React**: Real-time request for detailed observations.



1 Methodology: **Deconstructing scientific workflows.**

Deconstruct workflow into *types* of operations:

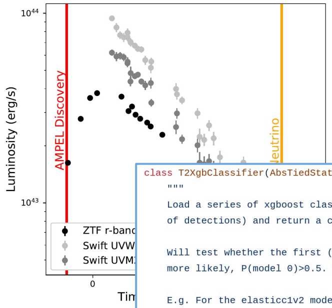
- **Add, Combine, Augment, React**.

Unique input/output, specified by abstract classes

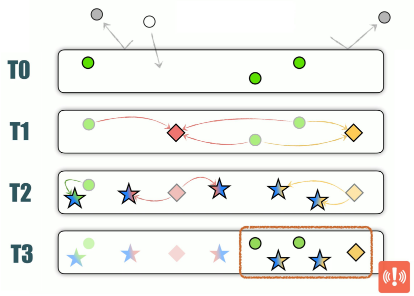
```
class AbsTiedStateT2Unit(Generic[T], AbsTiedT2Unit, abstract=True):
        """
        A T2 unit bound to a :class:`~ampel.content.T1Document.T1Document` (state of a stock),
        as well as the results of other T2 units
        """

        t2_dependency: Sequence[StateT2Dependency[T]]

        @abstractmethod
        def process(self,
                compound: T1Document,
                datapoints: Sequence[DataPoint],
                t2_views: Sequence[T2DocView]
        ) -> UBson | UnitResult:
                """
                Returned object should contain computed science results to be saved into the DB.

                .. note:: the returned dict must have only string keys and be BSON-encodable
                """
```

Scientist:
Domain interpretation

System engineer:
Task schedule interpretation



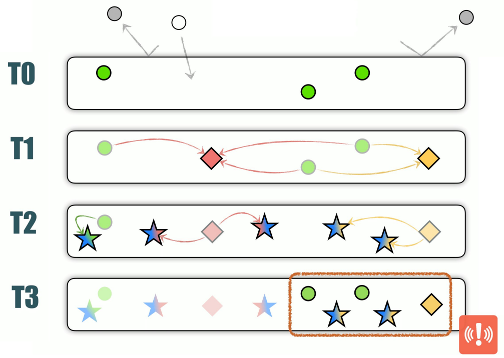

```
class T2XgbClassifier(AbsTiedStateT2Unit):
    """
    Load a series of xgboost classifier models (distinguished by number
    of detections) and return a classification.

    Will test whether the first (null) model of the loaded classifier is
    more likely, P(model 0)>0.5. What this means is determined by the training.

    E.g. For the elasticc1v2 model, the null model corresponds to Elasticc class 1
    (and model 1 to class 2.)

    """
```



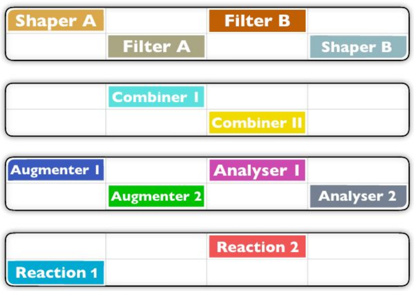
1 Methodology: **Deconstructing scientific workflows.**

A full science workflow constructed through:

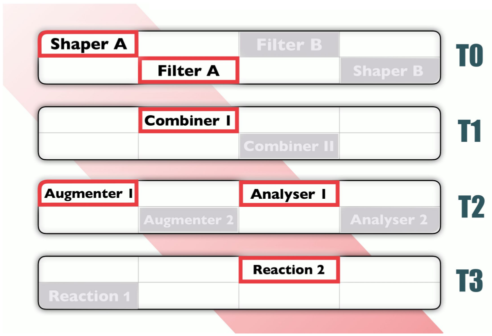
- → Unit libraries (e.g. github repositories)
- → Analysis schema constructed from units at all tiers (with configuration).



Tier definition



Unit library



Analysis schema

2

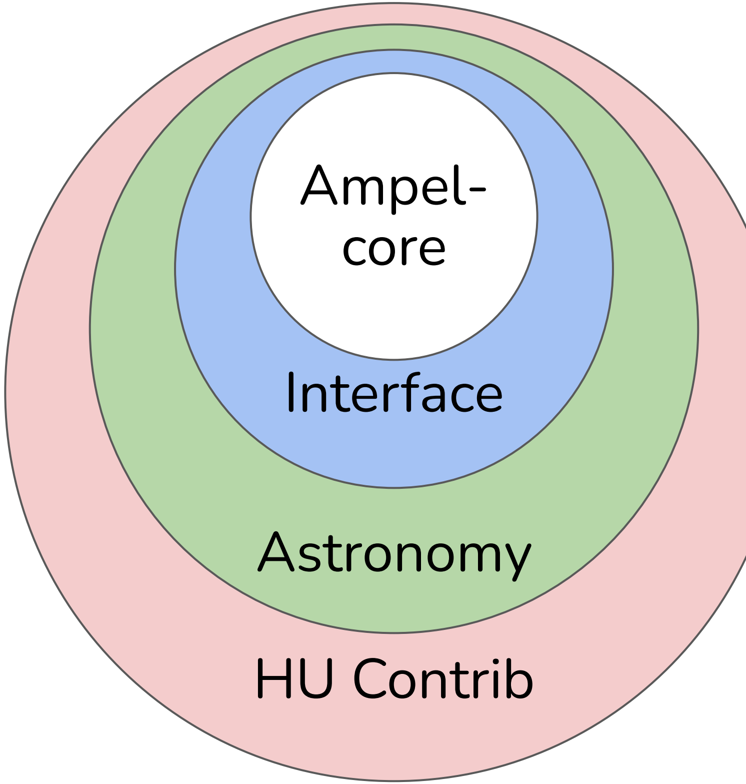
Workflow manager: **Execute workflow schema**

Application agnostic execution of workflows:
- Locally (development, reproduction)
- Parallel (real-time optimized)

AMPEL Core methods, optimized for database storage & provenance, manages input/output.

AMPEL Workers orchestrating unit execution.

Contributed analysis modules from different science groups are automatically detected.



2 Workflow manager: **Execute workflow schema**

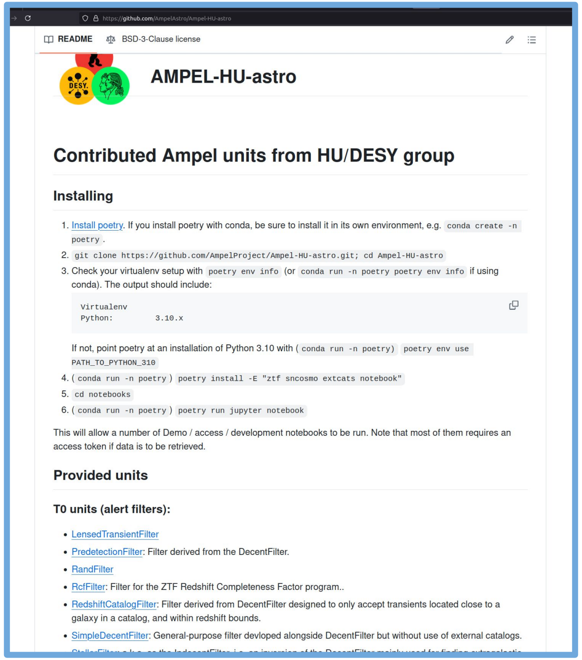
Application agnostic execution of workflows:

- Locally (development, reproduction)
- Parallel (real-time optimized)

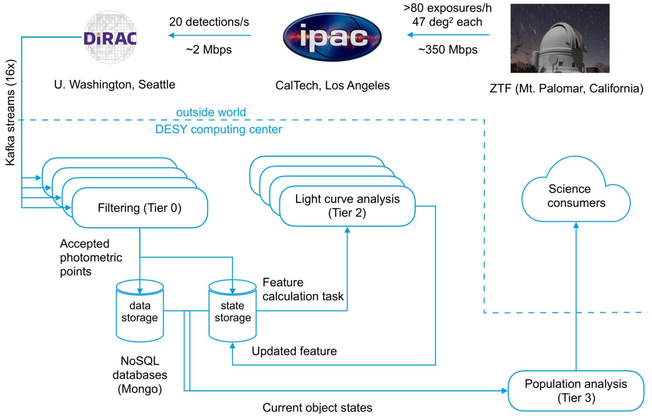
AMPEL Core methods, optimized for database storage & provenance, manages input/output.

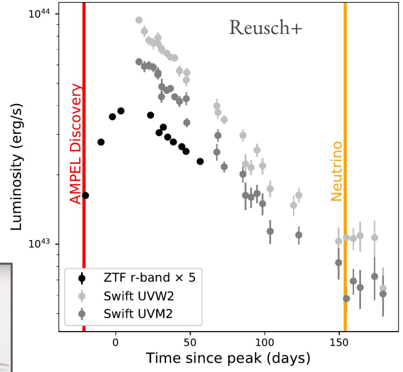
AMPEL Workers orchestrating unit execution.

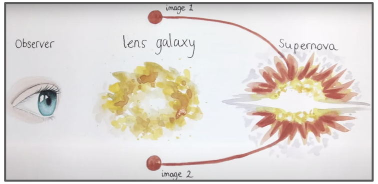
Contributed analysis modules from different science groups are automatically detected.

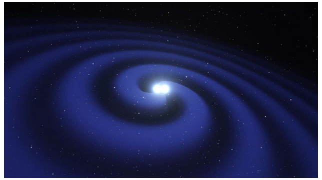


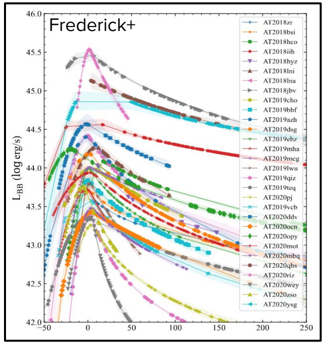
3 Real-time system: **The DESY Live AMPEL Instance**











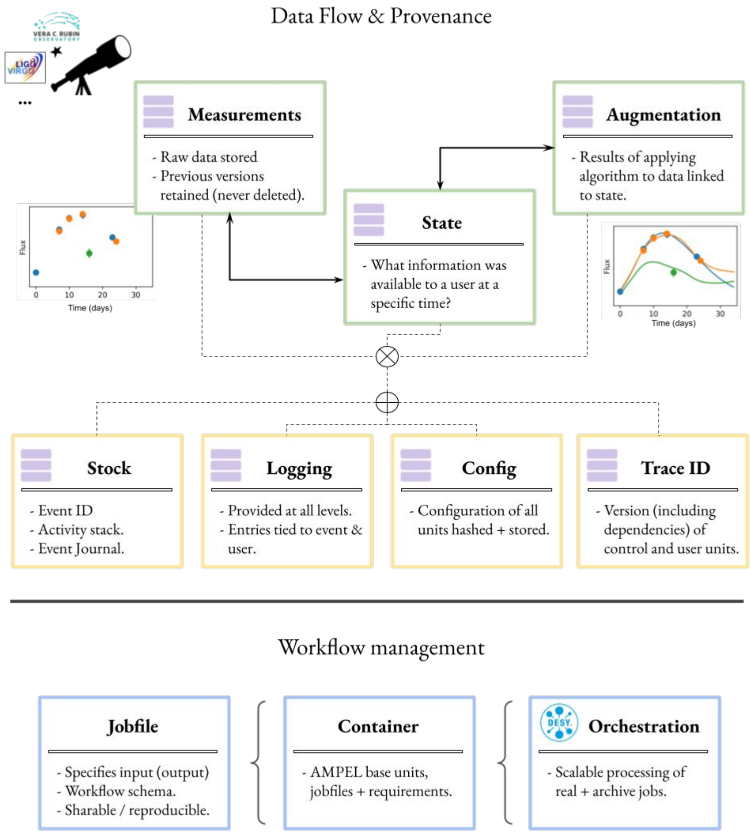
Continuously probing alert streams for exciting interest for different science programs, triggering real-time responses.
Hosted by DESY Zeuthen Computer Center.

Provenance & FAIR workflows

Records (data, states and results) are immutable objects.

The configuration of every operation (software versions, parameters) is recorded with the data.

A workflow (jobfile) can be published, distributed and executed at a HPC.

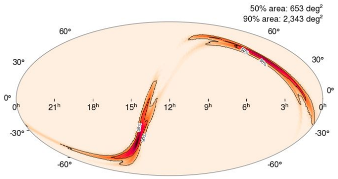


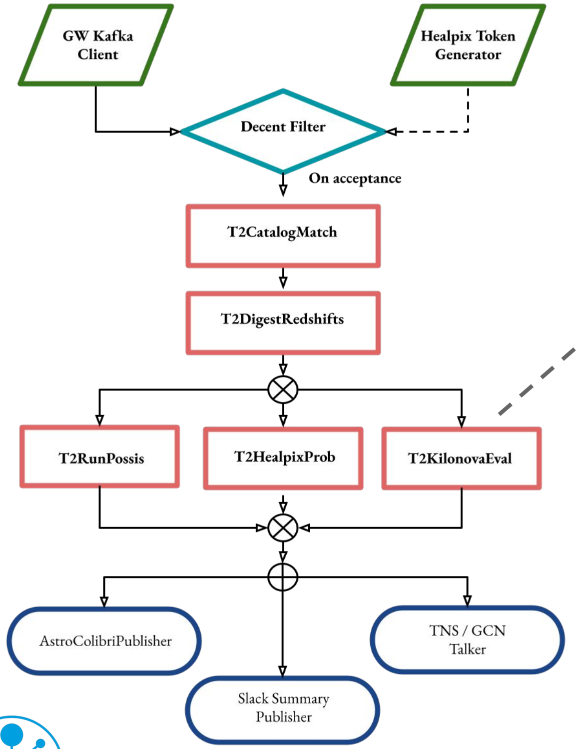
Gory details

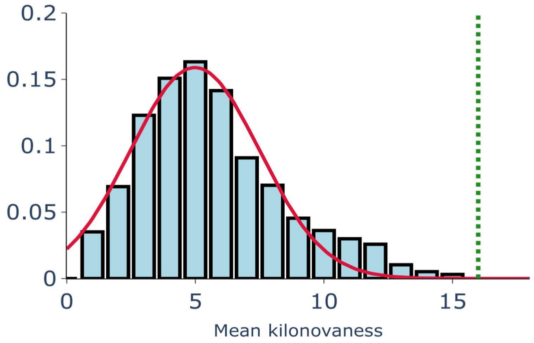
- Open source Python code:
  https://github.com/AmpelProject
- NoSQL (Mongo DB) database setup, including provenance-only collections.
- Two execution modes: sequential (local), parallel (at compute center).
- Modular integration through type hints, Pydantic, Renovate/continuous integration.
- Authorization via GitHub (orgs, teams, individual users).
- Rest API for remote access to results / catalogs.

AMPEL Workflow design process

1. **Design**
   - Install local environment
   - Formalize workflow
   - Identify existing units







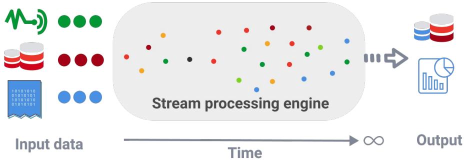
2. **Develop**
   - Create interface to science specific algorithm(s).
   - Test …

3. **Run**
   - Scale at computer center
   - Public live instance @ DESY
   - Publish code for others

AMPEL sample use cases

Systematic execution of analysis units while taking care of provenance, data storage & logging.

- Real-time stream analysis
- Modular software development
- Heterogeneous input sources
- Fine grained provenance
- Analysis variants of same data
- Code-to-data in science



Images: Wikipedia

Install AMPEL locally

Create a python 3.10 environment w. poetry and:

- git clone https://github.com/AmpelAstro/Ampel-HU-astro.git
- cd Ampel-HU-astro/
- poetry install -E "ztf sncosmo extcats notebook elasticc"

Allows to run demo notebooks

- cd notebooks
- poetry run jupyter notebook

Includes accessing archived or new ZTF alerts, running existing ML models and sample AMPEL units.

Summary

Astronomical transients allow us to use the Universe to test fundamental physics & cosmology.

Data streams are large and heterogeneous - AMPEL was constructed to allow FAIR, flexible and scalable analysis of real-time data streams.

AMPEL Core is domain agnostic and open source: Any analysis working with multiple data sources can create their AMPEL interface to allow repeatable, modular analysis schema.

More information:

- **Overview:** https://doi.org/10.1051/0004-6361/201935634
- **Contact:** **jnordin@physik.hu-berlin.de** or ampel-info@desy.de
- **PUNCH4NFDI tutorials**

1 Methodology: **Deconstructing scientific workflows.**

Deconstruct workflow into *types* of operations:
- **Add, Combine, Augment, React**.

Four separate execution layers, each defined by an input/output interface (in Python).

Domain experts develop fully-specified algorithms at each tier.

```python
class AbsTiedStateT2Unit(Generic[T], AbsTiedT2Unit, abstract=True):
        """
        A T2 unit bound to a :class:`~ampel.content.T1Document.T1Document` (state of a stock),
        as well as the results of other T2 units
        """

        t2_dependency: Sequence[StateT2Dependency[T]]

        @abstractmethod
        def process(self,
                compound: T1Document,
                datapoints: Sequence[DataPoint],
                t2_views: Sequence[T2DocView]
        ) -> UBson | UnitResult:
                """
                Returned object should contain computed science results to be saved into the DB.

                .. note:: the returned dict must have only string keys and be BSON-encodable
                """
```

```python
class T2XgbClassifier(AbsTiedStateT2Unit):
    """
    Load a series of xgboost classifier models (distinguished by number
    of detections) and return a classification.

    Will test whether the first (null) model of the loaded classifier is
    more likely, P(model 0)>0.5. What this means is determined by the training.

    E.g. For the elasticc1v2 model, the null model corresponds to Elasticc class 1
    (and model 1 to class 2.)

    """
```

1

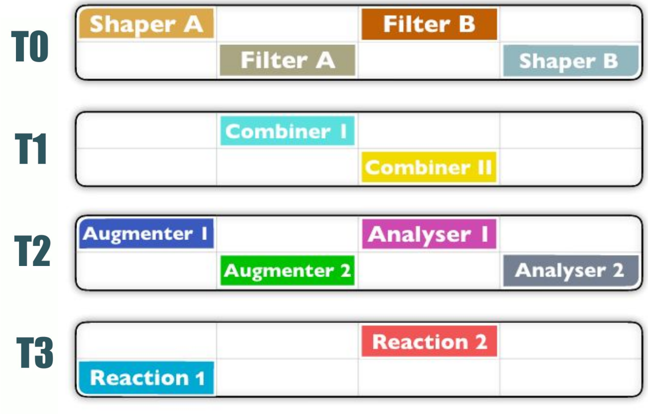
Methodology:
Deconstructing scientific workflows.

Deconstruct workflow into *types* of operations:

- **Add, Combine, Augment, React**.

Four separate execution layers, each defined by an input/output interface (in Python).

Domain experts develop fully-specified algorithms at each tier.



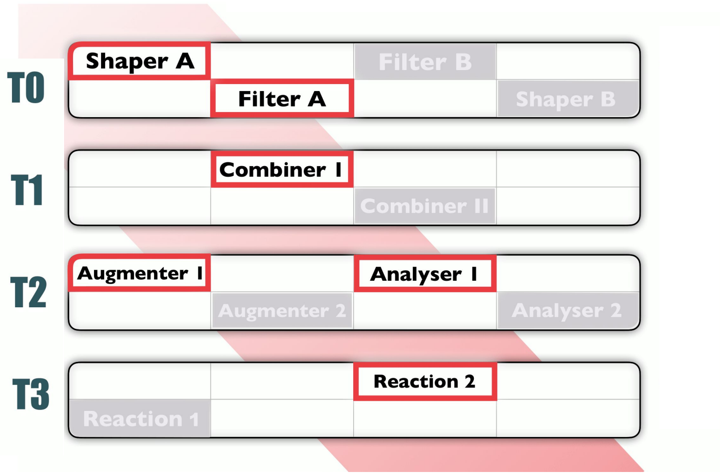
1 Methodology: **Deconstructing scientific workflows.**

Deconstruct workflow into *types* of operations:
- **Add, Combine, Augment, React**.

Four separate execution layers, each defined by an input/output interface (in Python).

Domain experts develop fully-specified algorithms at each tier.

Users construct analysis schema through combining units.



1 Methodology: **Deconstructing scientific workflows.**

Deconstruct workflow into *kinds* operations:
- **Add, Combine, Augment, React**.

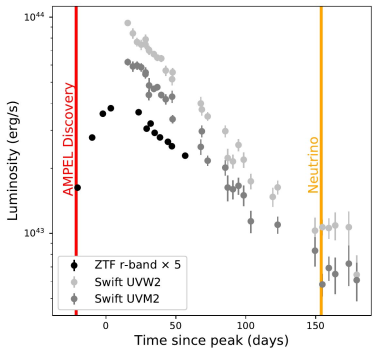
Four separate execution layers, each defined by [python] input/output method definitions.

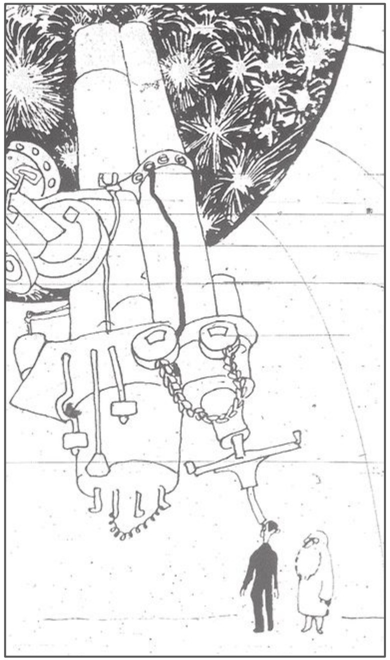
Domain experts develop fully specified algorithms at each tier.

```python
class AbsTiedStateT2Unit(Generic[T], AbsTiedT2Unit, abstract=True):
        """
        A T2 unit bound to a :class:`~ampel.content.T1Document.T1Document` (state of a stock),
        as well as the results of other T2 units
        """

        t2_dependency: Sequence[StateT2Dependency[T]]

        @abstractmethod
        def process(self,
                compound: T1Document,
                datapoints: Sequence[DataPoint],
                t2_views: Sequence[T2DocView]
        ) -> UBson | UnitResult:
                """
                Returned object should contain computed science results to be saved into the DB.

                .. note:: the returned dict must have only string keys and be BSON-encodable
                """
```

```python
class T2XgbClassifier(AbsTiedStateT2Unit):
    """
    Load a series of xgboost classifier models (distinguished by number
    of detections) and return a classification.

    Will test whether the first (null) model of the loaded classifier is
    more likely, P(model 0)>0.5. What this means is determined by the training.

    E.g. For the elasticc1v2 model, the null model corresponds to Elasticc class 1
    (and model 1 to class 2.)

    """
```

Astronomical transients





Needle in a haystack:

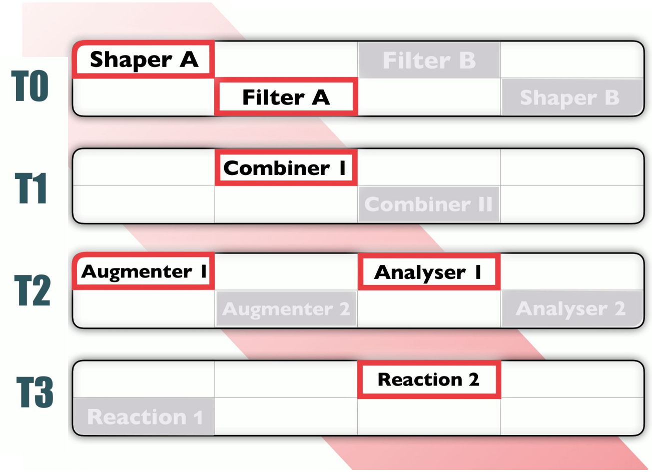
- Searching among billions of galaxies - every night.
- Yield millions of real but “boring” astronomical variables.
- Multiple telescopes + streams.
- Gravitational waves & neutrinos.
- Individual scientists.

2

Workflow manager:
Execute workflow schema

Software developers design how:
- Data is being provided and read from units.
- Operations are parallelized & scaled.
- Workers orchestrate unit execution.

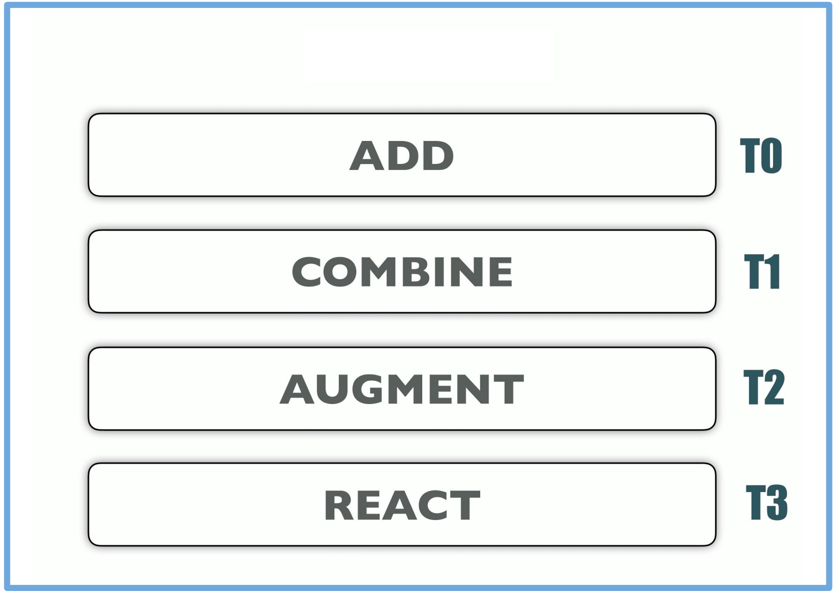
Provenance & reproducibility by default:
- Schema, configurations & software versions recorded at every run.
- State mechanism records knowledge develop with time.



1 Methodology: **Deconstructing scientific workflows.**

Any analysis constructed from operations of four different *kinds*.

Four separate execution layers, each distinguished by [`python`] input/output method definitions.

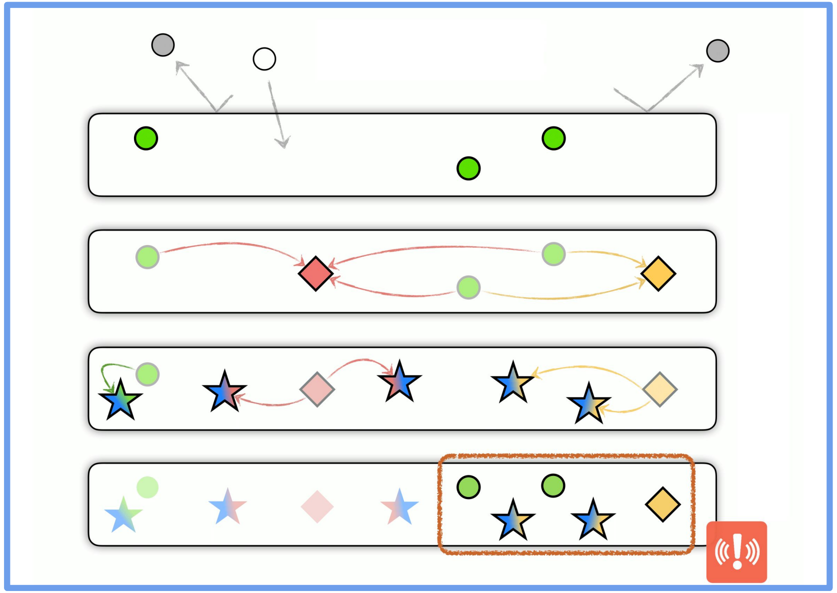


1

Methodology:
Deconstructing scientific workflows.

Any analysis constructed from operations of four different *kinds*.

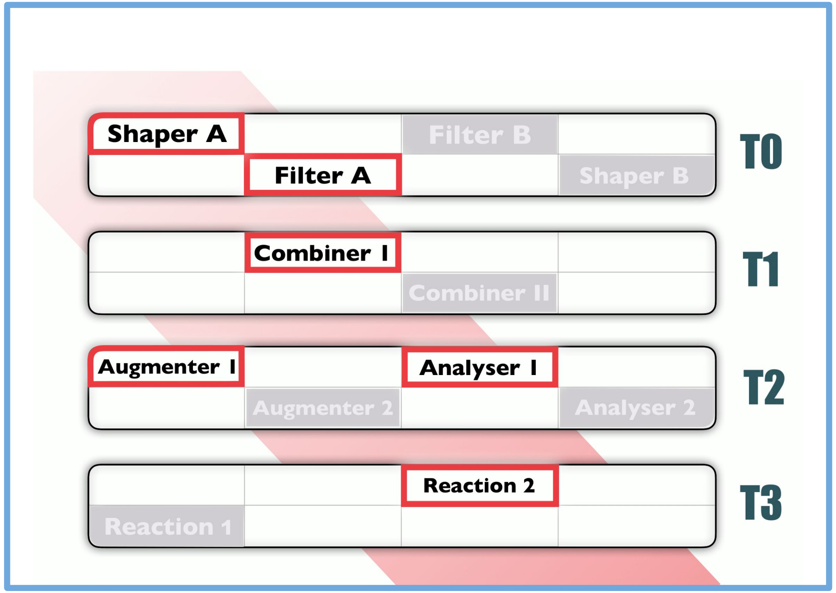
Four separate execution layers, each distinguished by [python] input/output method definitions.



1 Methodology: **Deconstructing scientific workflows.**

Any analysis constructed from operations of four different *kinds*.

Four separate execution layers, each distinguished by [python] input/output method definitions.



1 Methodology: **Deconstructing scientific workflows.**

Any analysis constructed from operations of four different *kinds*.

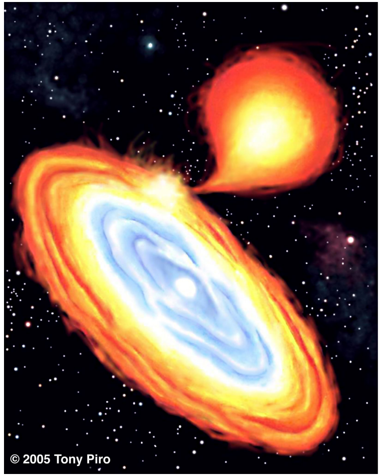
Four separate execution layers, each distinguished by [python] input/output method definitions.

```
class AbsTiedStateT2Unit(Generic[T], AbsTiedT2Unit, abstract=True):
        """
        A T2 unit bound to a :class:`~ampel.content.T1Document.T1Document` (state of a stock),
        as well as the results of other T2 units
        """

        t2_dependency: Sequence[StateT2Dependency[T]]

        @abstractmethod
        def process(self,
                compound: T1Document,
                datapoints: Sequence[DataPoint],
                t2_views: Sequence[T2DocView]
        ) -> UBson | UnitResult:
                """
                Returned object should contain computed science results to be saved into the DB.

                .. note:: the returned dict must have only string keys and be BSON-encodable
                """
```

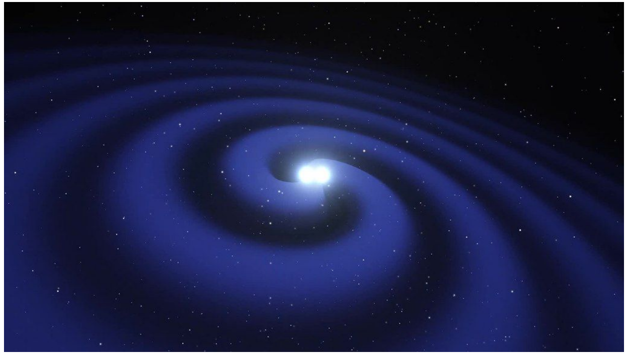
```
class T2XgbClassifier(AbsTiedStateT2Unit):
    """
    Load a series of xgboost classifier models (distinguished by number
    of detections) and return a classification.

    Will test whether the first (null) model of the loaded classifier is
    more likely, P(model 0)>0.5. What this means is determined by the training.

    E.g. For the elasticc1v2 model, the null model corresponds to Elasticc class 1
    (and model 1 to class 2.)

    """
```

Astronomical transients

Certain explosive, transient events carry distinct information regarding distances, energies & conditions across space.



Type Ia supernovae
Standardizable thermal detonations

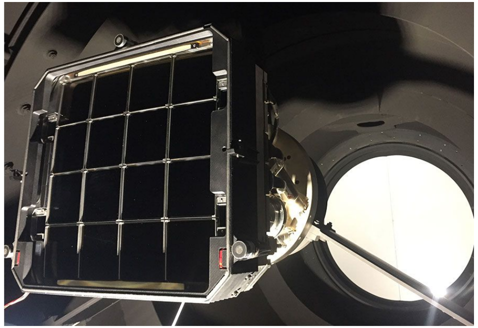


Kilonovae
Optical counterparts to GWs

Tidal Disruption Events
Stars ripped by black holes

Optical all-sky surveys

Detector development allows the full sky to be monitored to greater cosmological distances.



The Zwicky Transient Facility detects >100 000 transients / night.

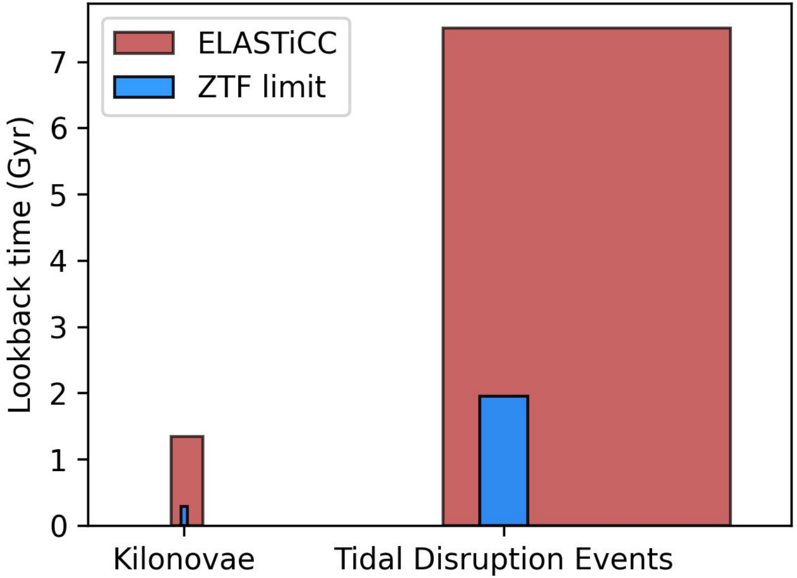
Optical all-sky surveys

Si detector development allows the full sky to be monitored, to ever greater depth.

LSST will starting 2025 deliver >10x more.

Optical all-sky surveys

Si detector development allows the full sky to be monitored, to ever greater depth.

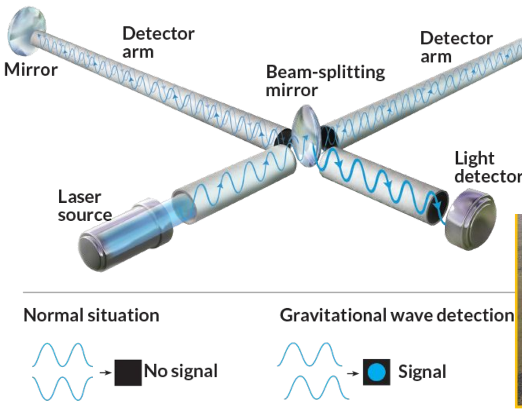


LSST will discover transients from throughout most of our Universe's history.

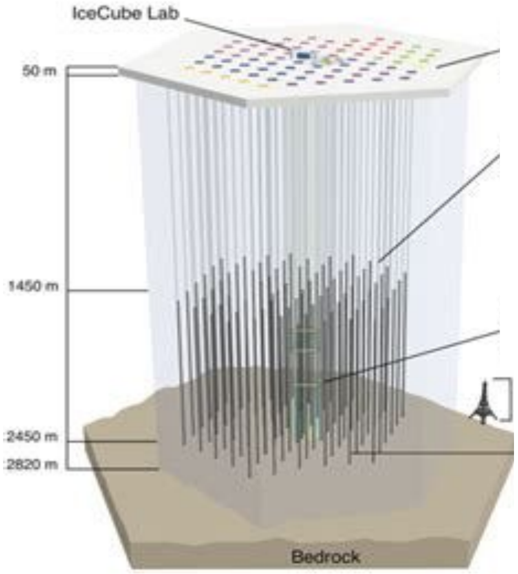
Gravitational waves & ghost particles

Non-photon based telescopes:

IceCube neutrino observatory.

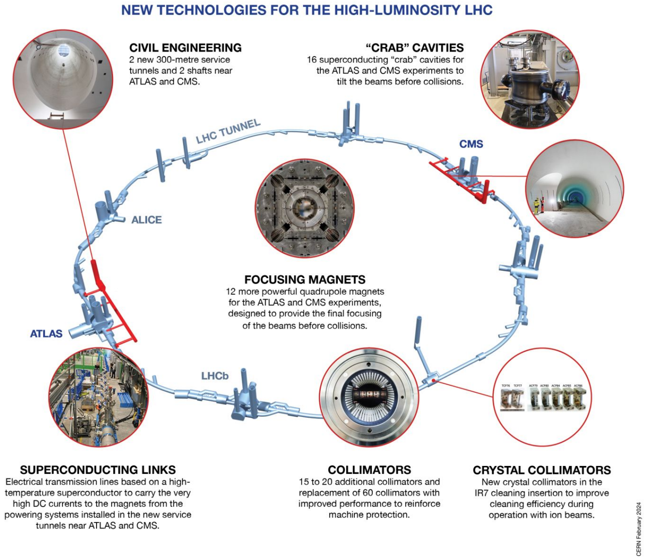


LIGO/Virgo Gravitational Wave telescopes.

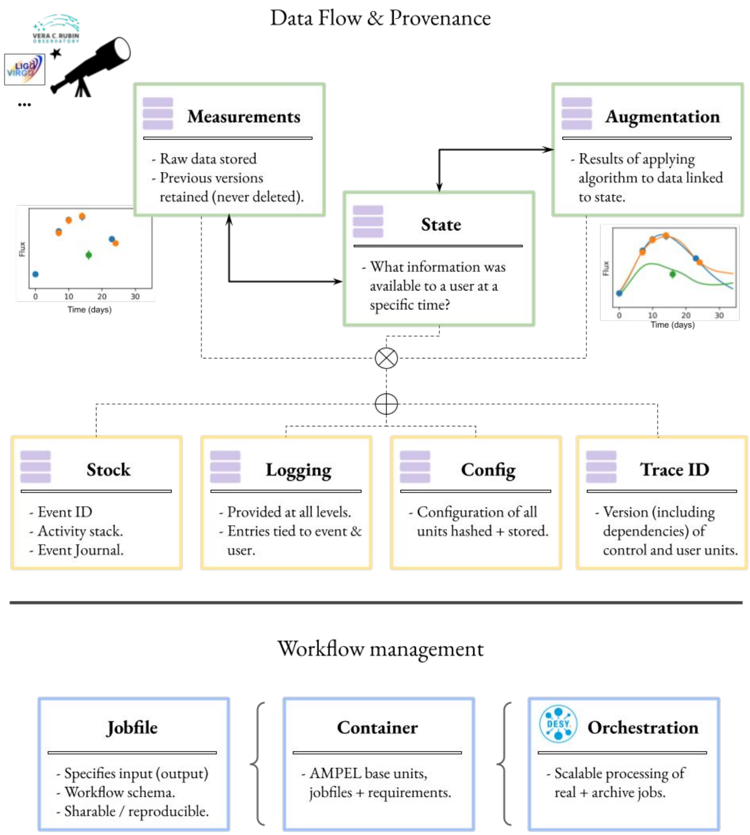


Development continues

Future radio telescopes produce 157 TB/s



250 million million proton collisions / yr @ LHC



Bottom-Up Provenance

Modularity / Flexibility

- Model plug-in
- Same setup for offline / online
- Core modules astro "agnostic"

Provenance

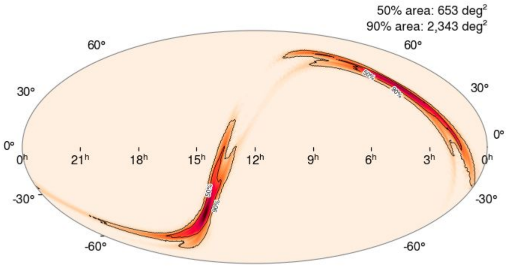
- Automatic version/config store
- Built in log classes
- Reproducible workflows
- Deduplication

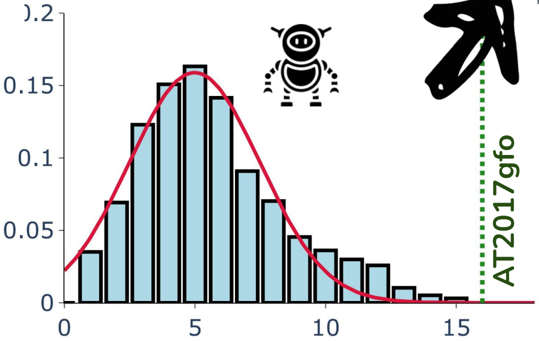
Operational

- Data archives
- LSST / ZTF / Ultrasat broker
- Dev + cluster workflow environments

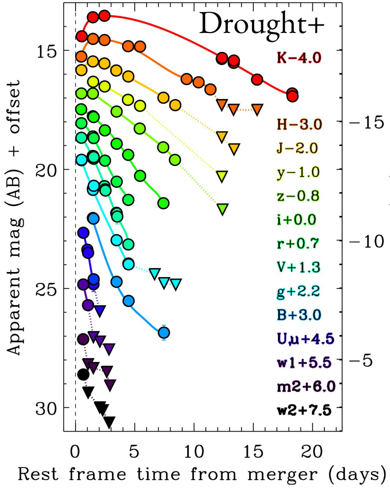
S1: **MM 101 - finding Kilonovae**

From GW to follow-up with autonomous “kilonovaness” calculation



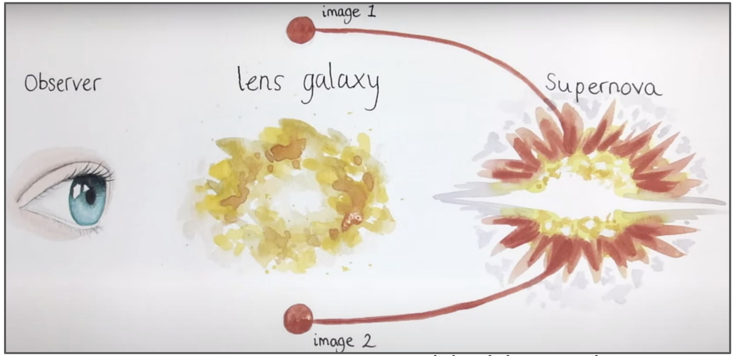


A. Ernst



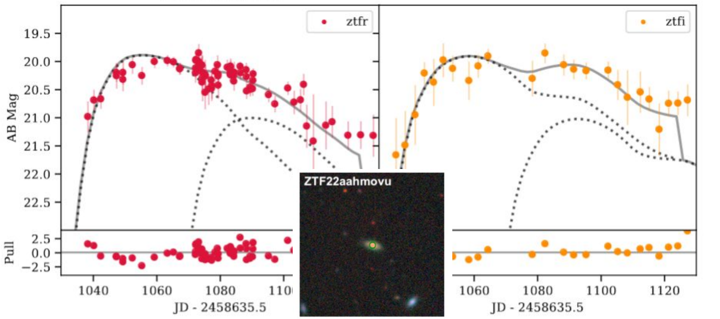
Any day now...

S2: **Searching for Gravitationally Lensed Supernovae**



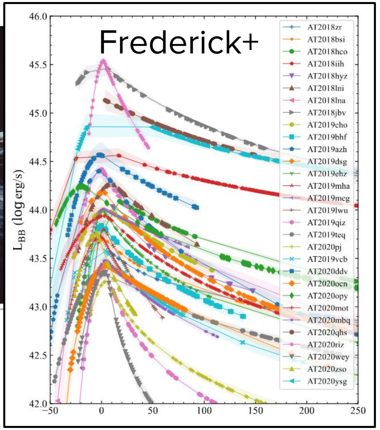
N. Ahrendse

- Potential key to the Hubble tension.
- Sneak peek into the early Universe
- Now found in *systematic*, large-volume searches.



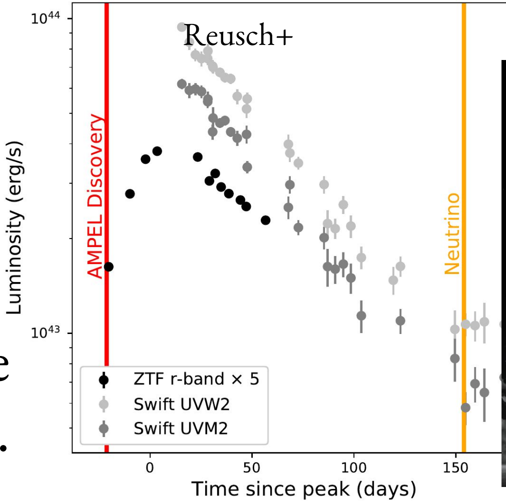
A. Townsend

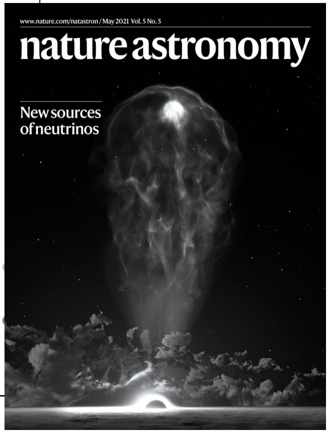
S3: **Tidal Disruption Events and neutrino emission**



Dramatic increase in TDE detection rate.

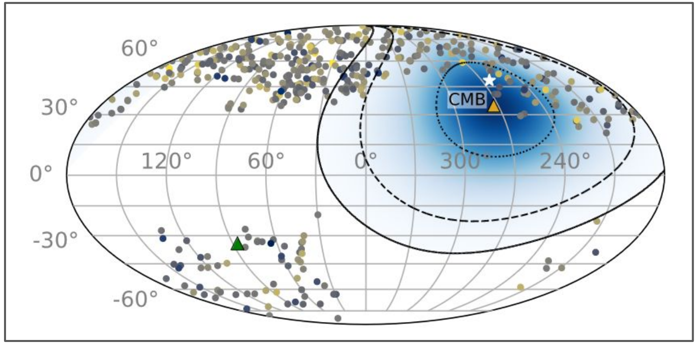
Coincident with IceCube neutrino alerts.

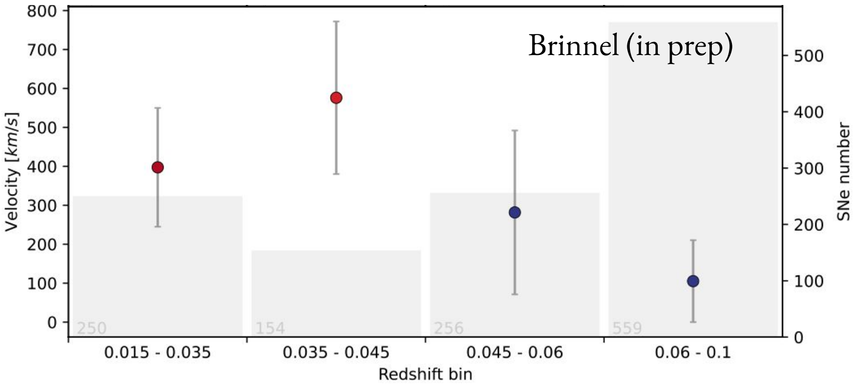




S4: Bulk flows - where are we going?

Large samples of Type Ia supernovae map the dynamic Universe.





Significant local flows, decreasing at larger distances - no signs of dark flows.

How does it work:

- An analysis divided into module, each living in a separate tier.
- Instrument specialists design input feeds.
- Software designers create a scalable systems.

Throughout provenance and reproducibility is built in from the ground.

You do

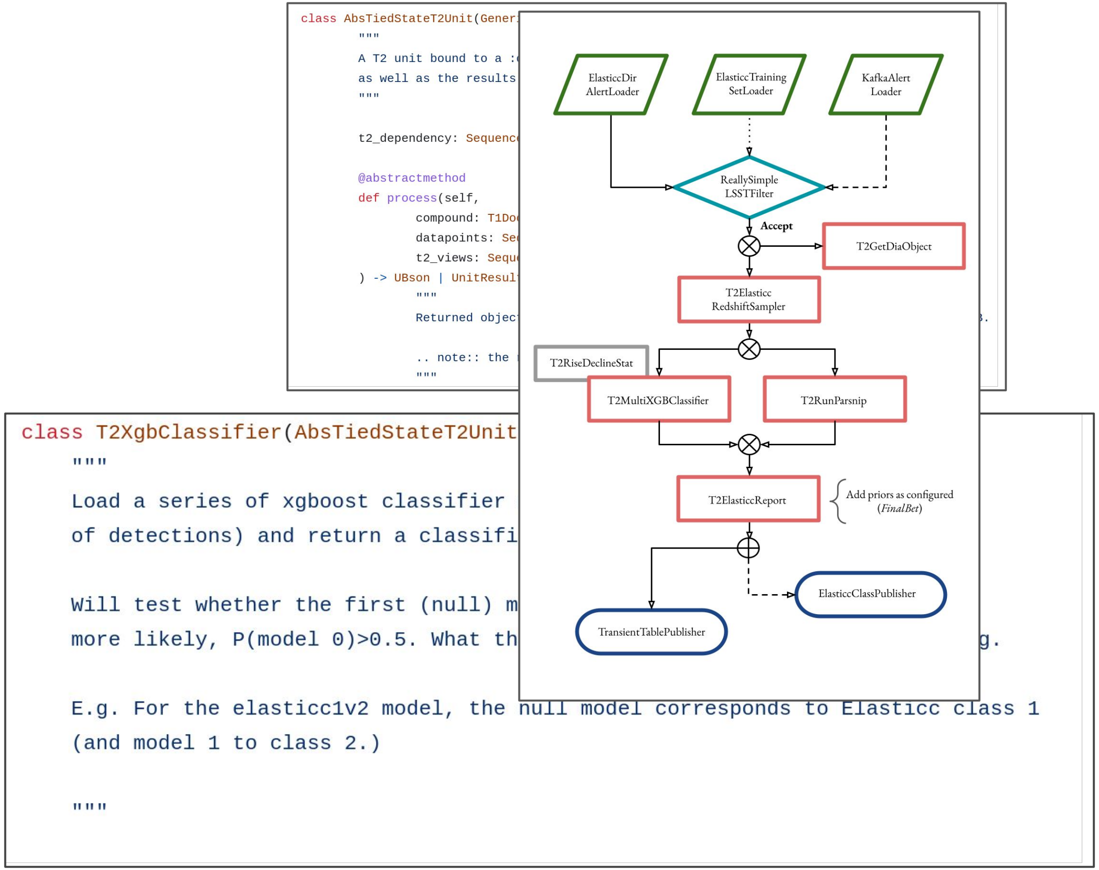
- Encode “your” algorithm as python module, implementing a set of I/O methods.
- Create workflow through a chain of modules.
- Run on test data locally.

```python
class AbsTiedStateT2Unit(Generic[T], AbsTiedT2Unit, abstract=True):
        """
        A T2 unit bound to a :class:`~ampel.content.T1Document.T1Document` (state of a stock),
        as well as the results of other T2 units
        """

        t2_dependency: Sequence[StateT2Dependency[T]]

        @abstractmethod
        def process(self,
                compound: T1Document,
                datapoints: Sequence[DataPoint],
                t2_views: Sequence[T2DocView]
        ) -> UBson | UnitResult:
                """
                Returned object should contain computed science results to be saved into the DB.

                .. note:: the returned dict must have only string keys and be BSON-encodable
                """
```

```python
class T2XgbClassifier(AbsTiedStateT2Unit):
    """
    Load a series of xgboost classifier models (distinguished by number
    of detections) and return a classification.

    Will test whether the first (null) model of the loaded classifier is
    more likely, P(model 0)>0.5. What this means is determined by the training.

    E.g. For the elasticc1v2 model, the null model corresponds to Elasticc class 1
    (and model 1 to class 2.)

    """
```

You do

- Encode “your” algorithm as python module, implementing a set of I/O methods.
- Create workflow through a chain of modules.
- Run on test data locally.



You do

- Encode “your” algorithm as python module, implementing a set of I/O methods.
- Create workflow through a chain of modules.
- Run on test data locally.

AMPEL does

- Access to (field standard) data sources.
- Scaling (local or remote).
- Guaranteed provenance.
- Reproducibility / code sharing.

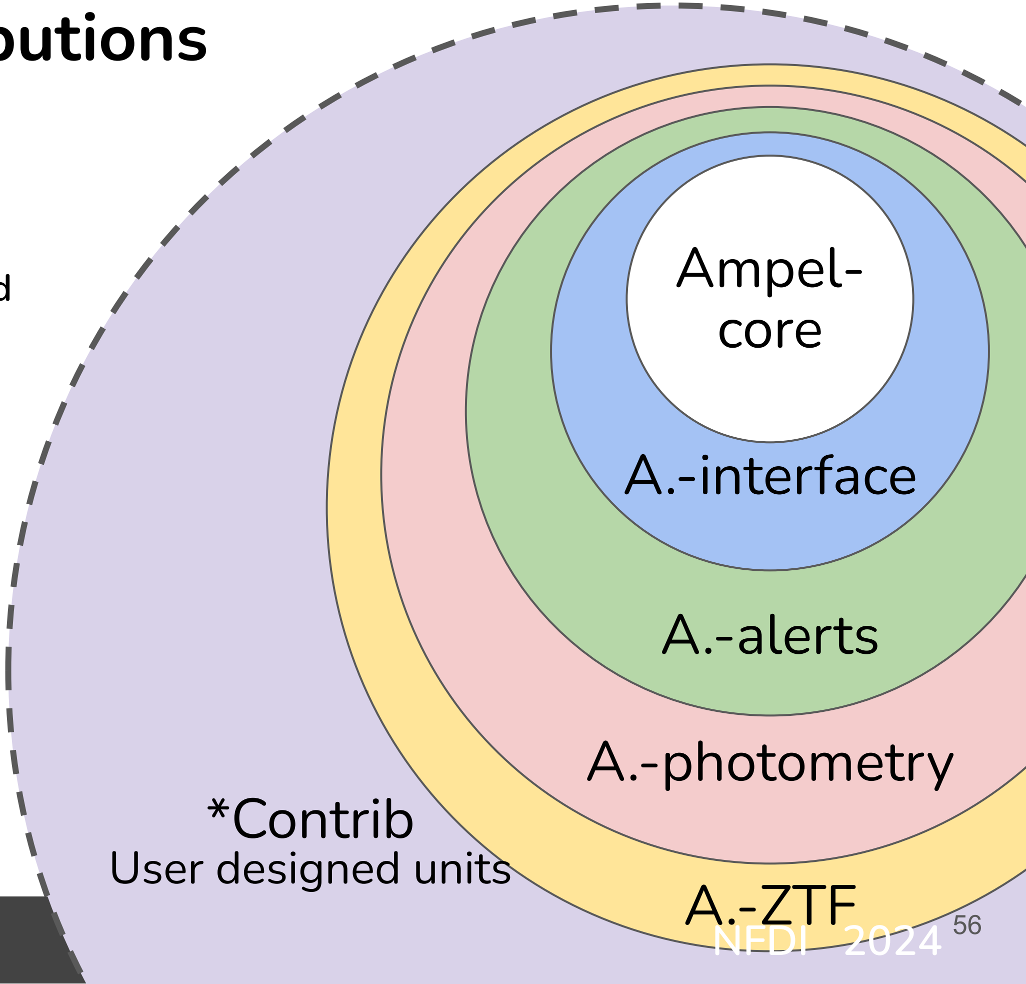
2 Merging user contributions

AMPEL Core:

- Workers at each tier executes units with requested input, allowing system control and parallelisation.
- Results stored in NoSQL (Mongo) DB.
- Built-in provenance tracking (event journal, logs and jobs)

Execute a job:

- locally to develop,
- at a cluster for large-volume archive runs
- in a live instance to analyze real-time data



Sum up:

- Exciting science goals!
- Need to react fast.
- Multiple, high throughput alert streams.
- A sociology with small, independent research groups each doing their own thing.

A huge information management challenge! Who did what why using which algorithm?

AMPEL design goals:

Let people do what they are best at:

- **Scientist tune their detection algorithms.**
- **Instrument specialists design input feeds.**
- **Software designers create a scalable systems.**

Throughout provenance and reproducibility is built in from the ground.

AMPEL in astrophysics today

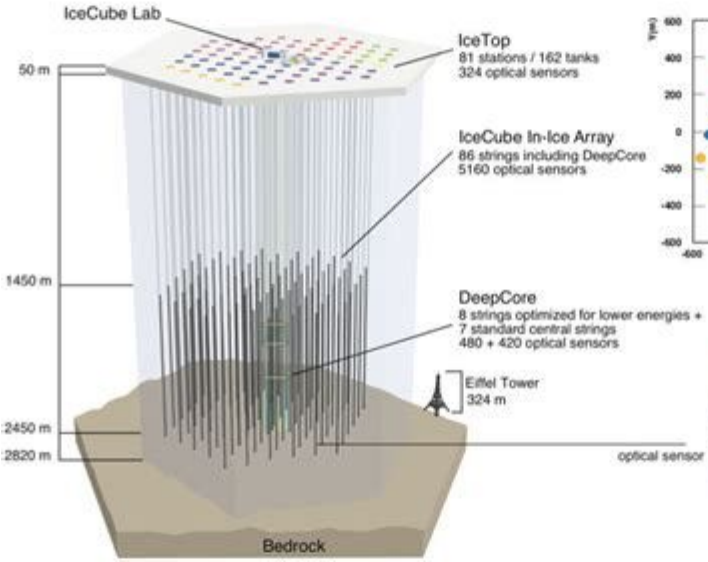
- Analysis engine for real-time reactions and multi-messenger astronomical analysis.
- Central hub for current and future large astronomical observatories: ZTF, VRO, Ligo/Virgo, Icecube, Ultrasat, CTA …
- Hosting parallel sequence of top of the line ML classifiers, critical for high throughput usage.
- Live instance maintained at DESY (Zeuthen)
- Maintained; ~ 10yr mission planning

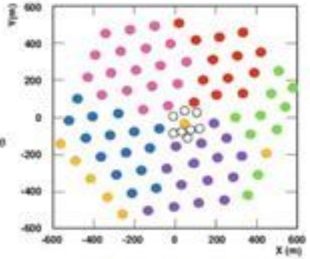
Astrophysics III

- Iphone: 7.5mm^2
- ZTF: 40cm^2
- Neutrinos (“ghost particles”) create in cosmic acceleration.

All explosive transient phenomena which suddenly appears.

Astrophysics III

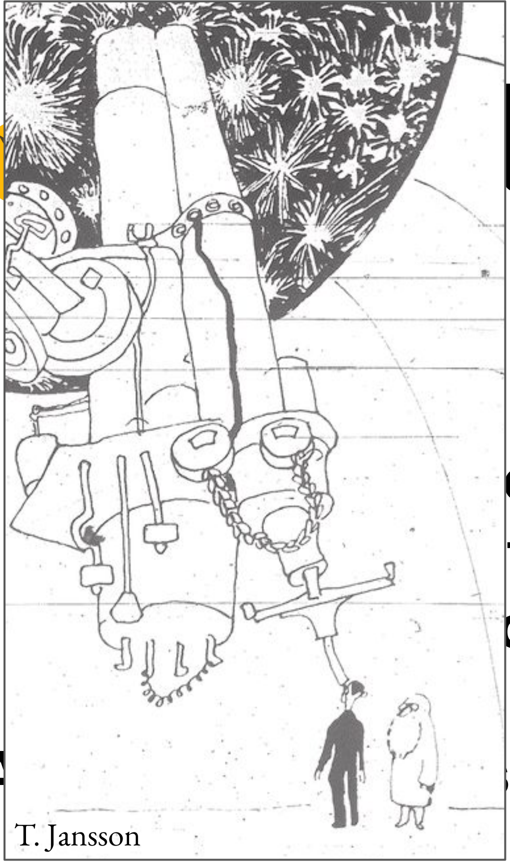




IceCube Digital Optical Module (DOM)

New observatories detect gravitational waves and n

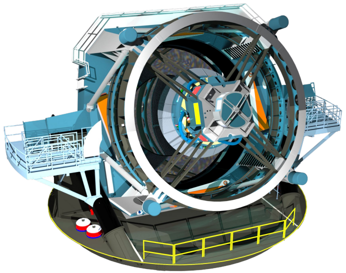
trophysics IV

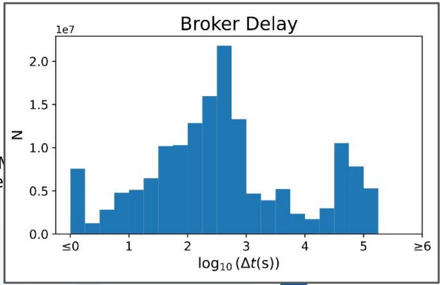


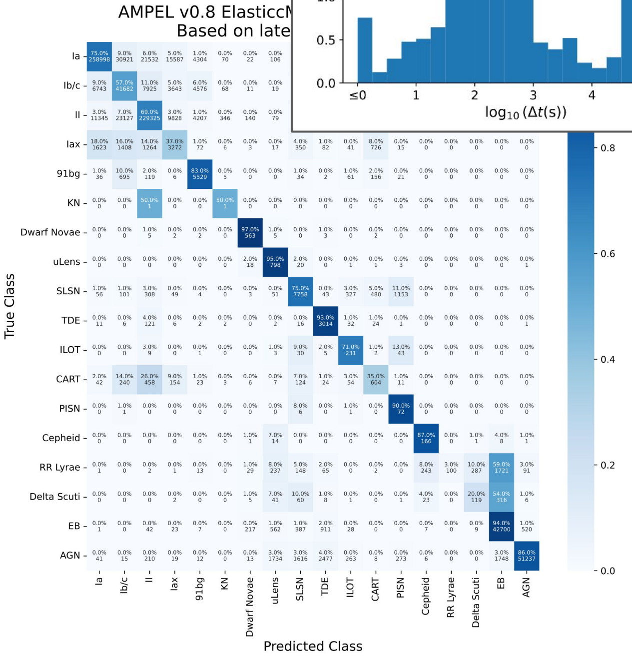
ovae: thermonuclear detonation of a star.
tron stars merge to form black hole.
ost particles") create in cosmic acceleration.

A ient phenomena which suddenly appears.

Outlook







Vera Rubin Observatory:

- AMPEL one of six LSST endpoints
- Fast response w. ~80% ML classification (ELAsTiCC data challenge)
- Public German interface!

More streams:

- ULTRASAT / LS4 / GOTO / LVK-O4 / SKA / ...

Improved usability.

High Throughput Time-Domain Astronomy

The Zwicky Transient Facility already saturates what human observers can parse, understand & publish (8000+ SNIa, 2000+ other SN).

ZTF

~ 1 million alerts per night → ~ 10 million alerts per night

~ 1TB / night → ~ 20TB / night

VERA C. RUBIN OBSERVATORY

~ Data rate of 2 TB/s → ~ Data rate of 157 TB/s

SKAO

Cosmic Neutrino Group:

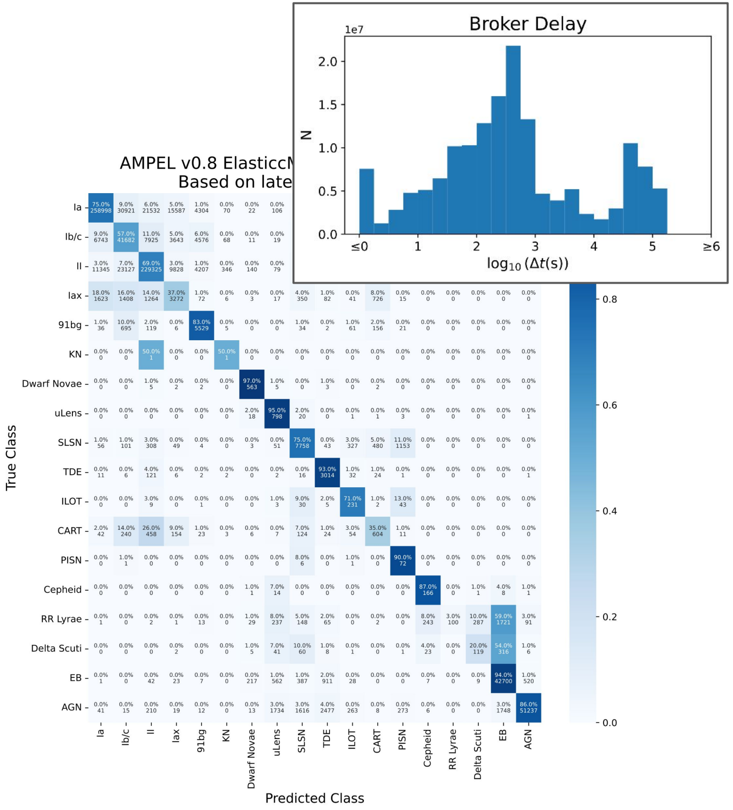
IceCube (see Summer's talk)

- ❑ Real-time Multi-Messenger studies.
- ❑ Cosmology with Type Ia supernovae.
- ❑ Era of small, robotic telescopes.
- ❑ ML classification of transients.
- ❑ AMPEL platform.

ML classification of transients

ML methods needed to respond to detection "torrent". Current generation based on combined encoder+decoder models, boosted decision trees, domain specific feature extraction and noise augmentation.

Models provided the best scores during the ELaSTiCC data challenge simulation of the LSST survey.



… one more thing

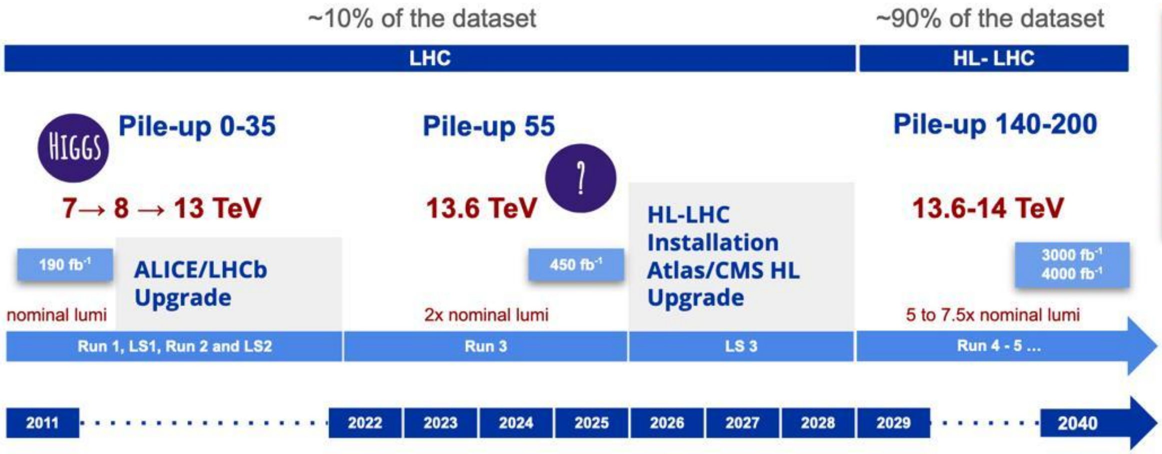
Everything is better in the UV

In principle,

- **Sensitivity to hot, fast transients**
- **Large field of view**
- **Good resolution**

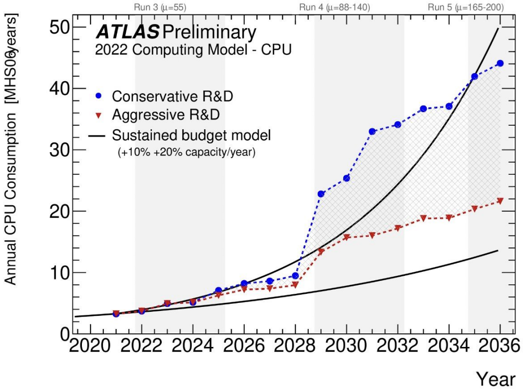
… just need to escape the atmosphere

High Luminosity LHC – advanced resource needs



Developments for reconstruction of

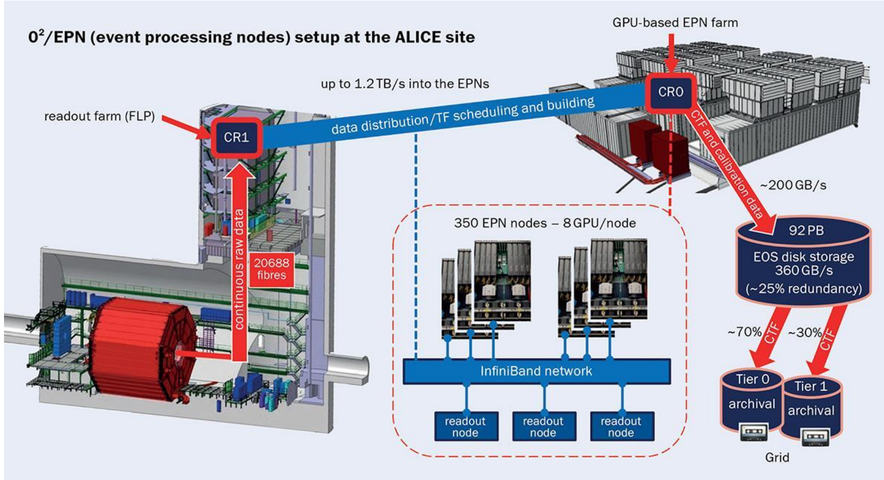
- Increased data rates
- Larger data volumes
- More complex signals



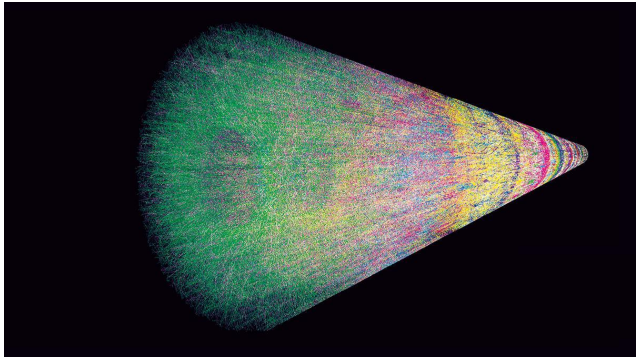
- Resource needs compared to ongoing Run-3 increase be a factor 10 to 60
- CPU capacity can increase ~ 10%/year

High-throughput data processing on GPUs – ALICE

Overview of the ALICE detector dataflow.
All detector front-end cards are read out and readout nodes are connected to the **EPN farm**.
Here **data processing** takes place on **GPUs**.
Output is transferred to the CERN distributed storage system, EOS. Credit: ALICE collab.



Processing of data rates of TB/s on a GPU cluster



Visualisation of a 2 ms time frame of PbPb collisions at 50 kHz interaction rate in ALICE Time Projection Chamber, showing **tracks from different primary collisions in different colours**. Credit: ALICE collab.

[Cern courier: ALICE sustainable computing]