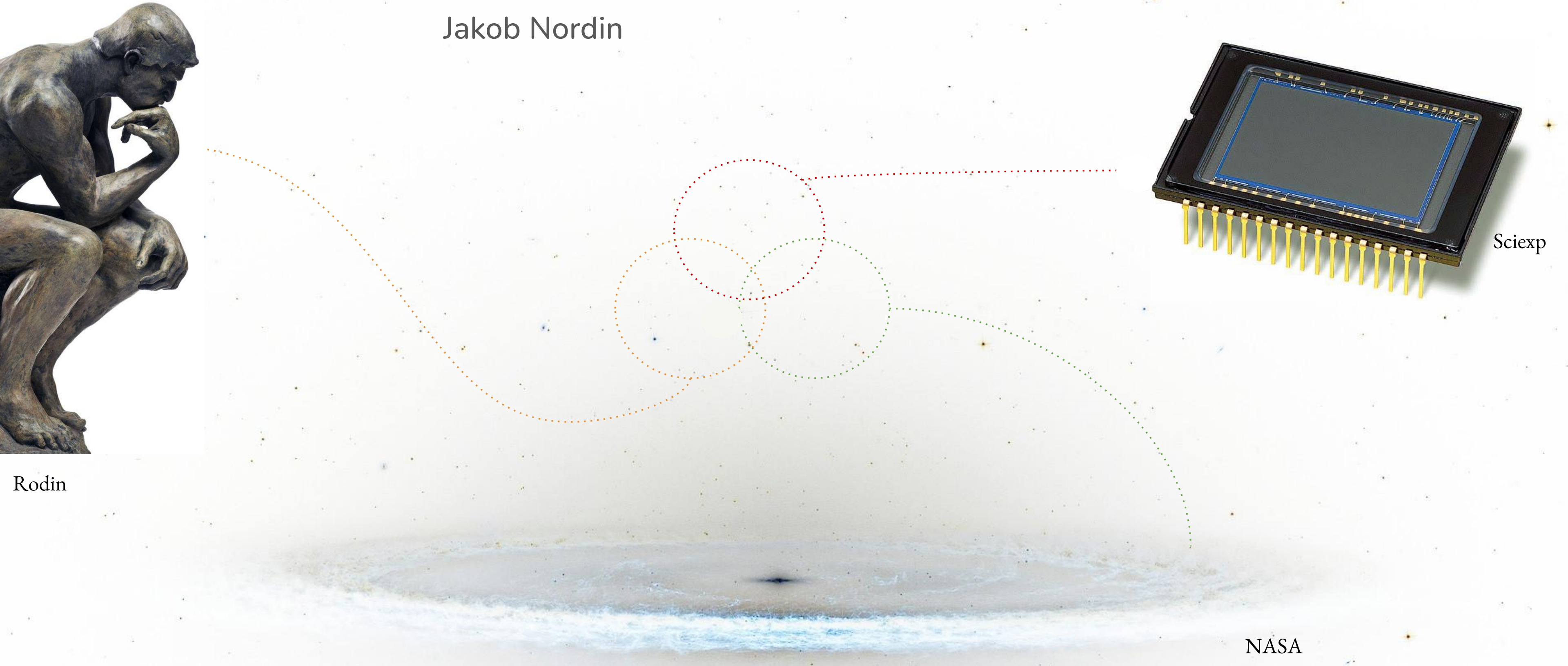
AMPEL

An analysis engine for the era of high-throughput multi-messenger observatories

Jakob Nordin

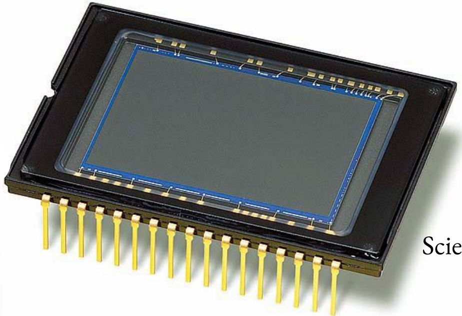


AMPEL

An analysis engine for the era of high-throughput multi-messenger observatories

Jakob Nordin

Rodin



Sciexp

NASA

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

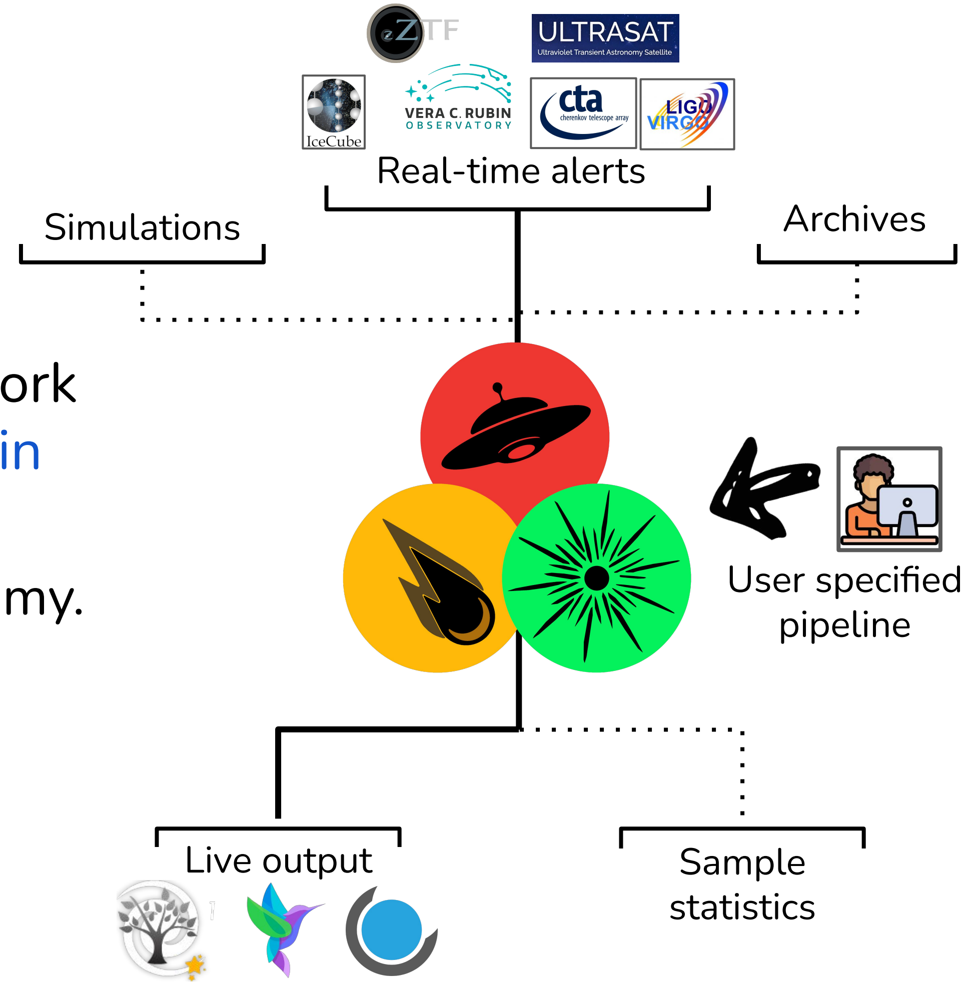
SKAO

AMPEL

Analysis and workflow framework for high throughput time-domain astronomy.

Realizes code-to-data in astronomy.

https://github.com/AmpelAstro



Real-time alerts

Simulations

Archives

User specified pipeline

Live output

Sample statistics

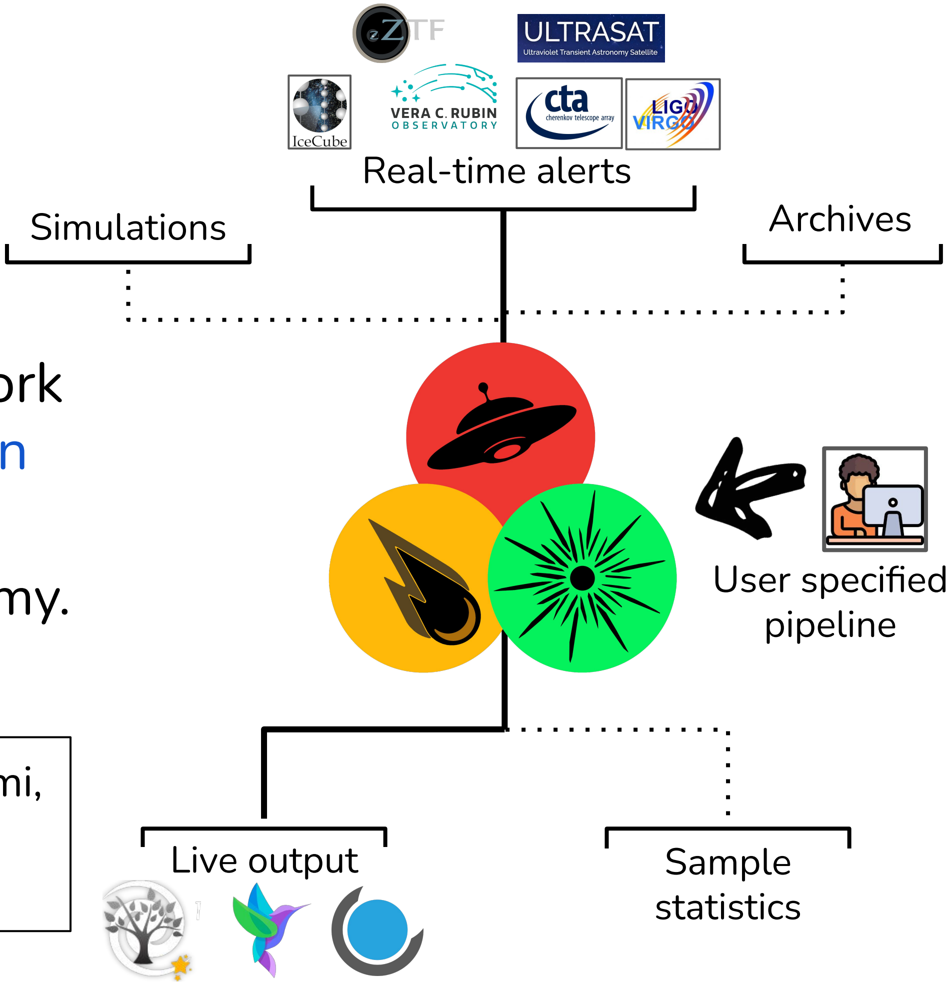
AMPEL

Analysis and workflow framework for high throughput time-domain astronomy.

Realizes code-to-data in astronomy.

https://github.com/AmpelAstro

Team: JN, **V. Brinnel, J. van Santen**, M. Giomi, S. Reusch, M. Kowalski, R. Stein, A. Townsend, A. Ernst, S. Velzen, ...



ZTF
ULTRASAT
Ultraviolet Transient Astronomy Satellite
IceCube
VERA C. RUBIN OBSERVATORY
cta
cherenkov telescope array
LIGO
VIRGO

Real-time alerts

Simulations

Archives

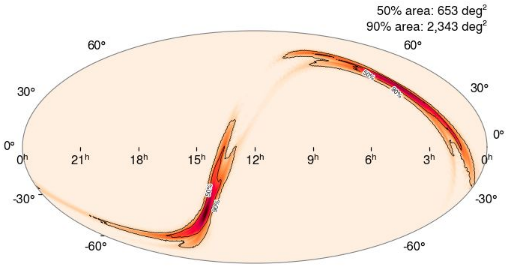
User specified pipeline

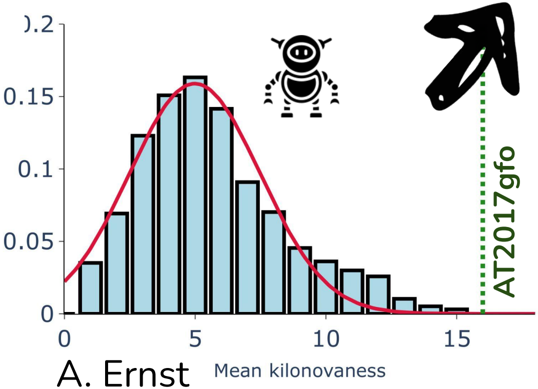
Live output

Sample statistics

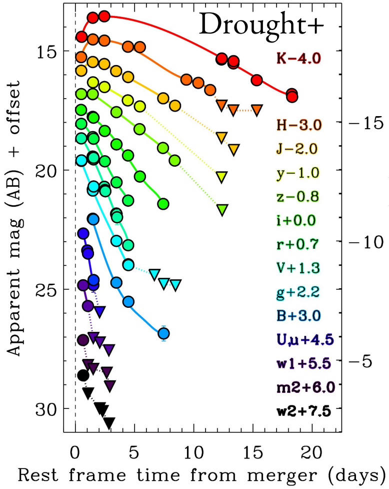
S1: MM 101 - finding Kilonovae

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



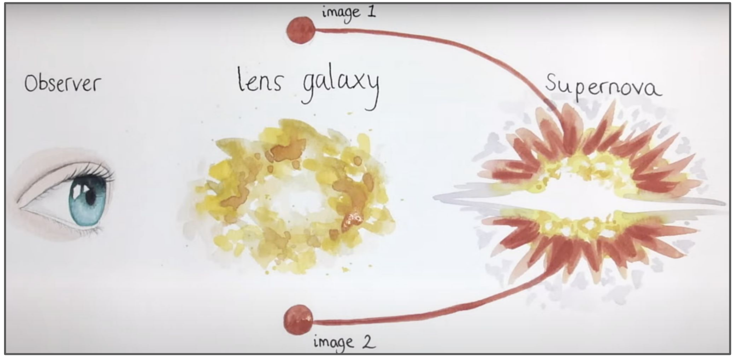


A. Ernst



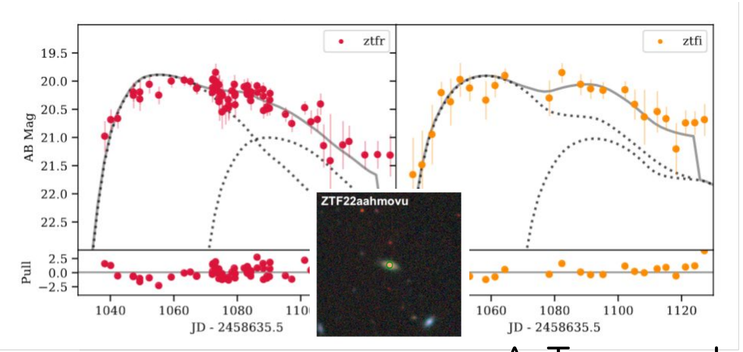
Any day now...

S2: **Searching for Gravitationally Lensed Supernovae**



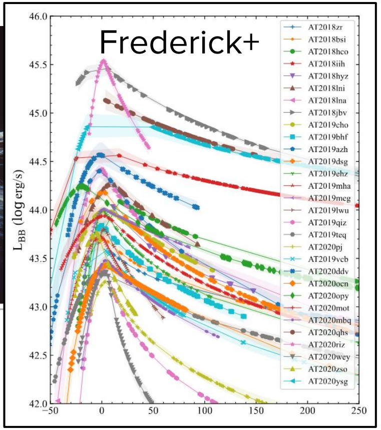
N. Ahrendse

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

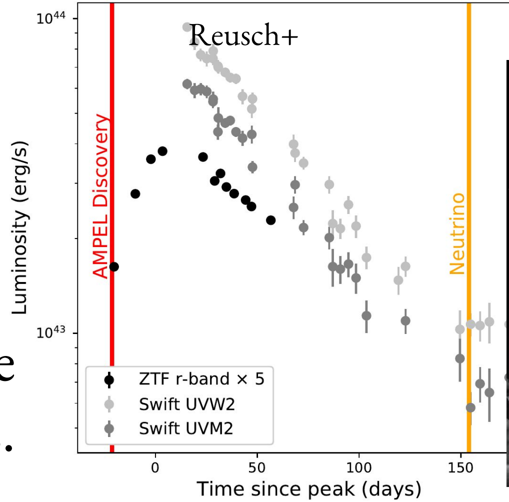


A. Townsend

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



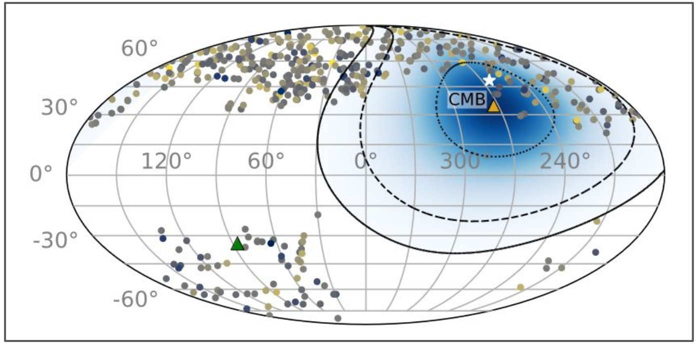
Dramatic increase in TDE detection rate.

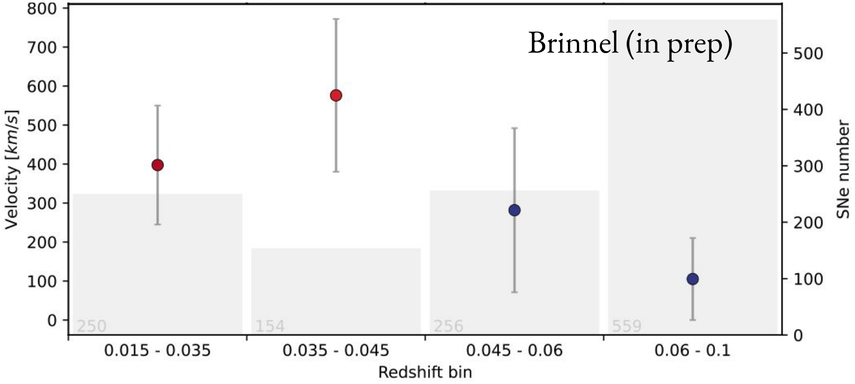


Coincident with IceCube neutrino alerts.

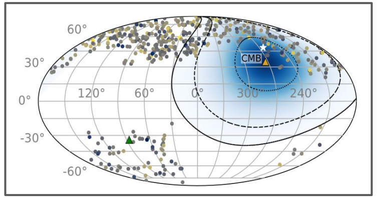
S4: **Bulk flows - where are we going?**

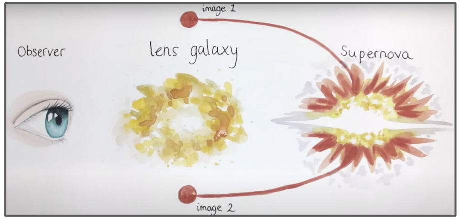
Large samples of Type Ia supernovae map the dynamic Universe.





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





Different science, same structure:

Astronomical Transients

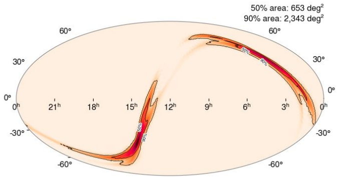
Wide-field detectors

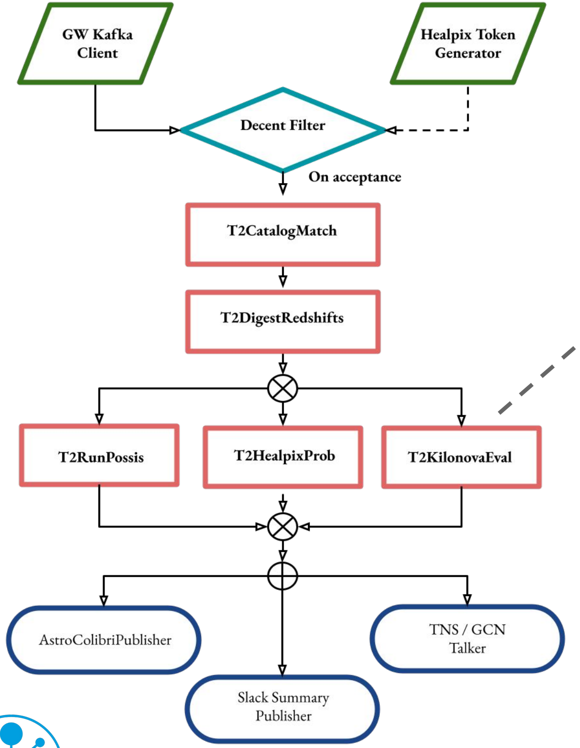
Theoretical Model

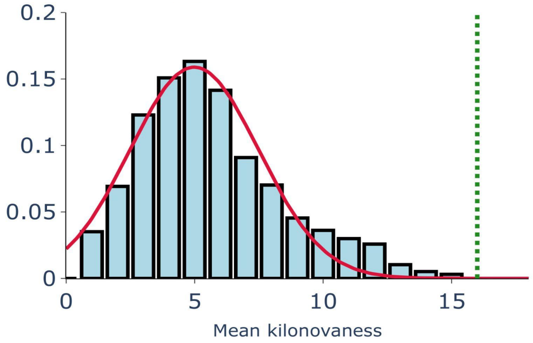
AMPEL allows a "prediction" to be systematically compared with large data volumes, based on an ecosystem of existing and contributed units.

Workflow design process

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





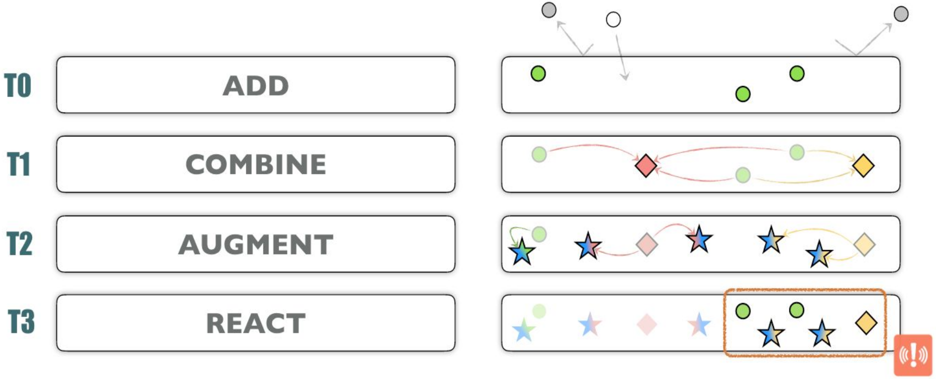


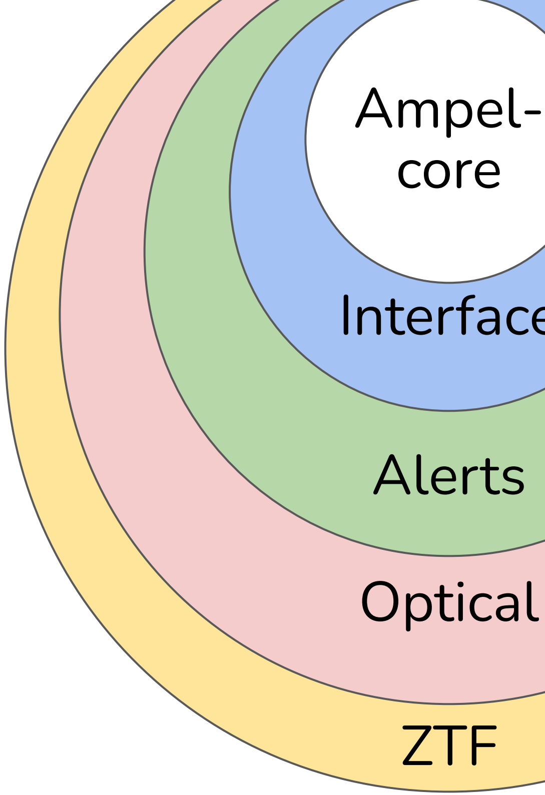
2. **Develop**
   - Create interface to science specific algorithm(s).
   - Test ...

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

Implementation

- *Workflow deconstruction*: tiers & states
- *Provenance* built in.
- Software “onion” with efficient core.
- Domain experts provide *analysis modules* for specific science goals
- [python, mongoDB, typehints, …]





Outlook

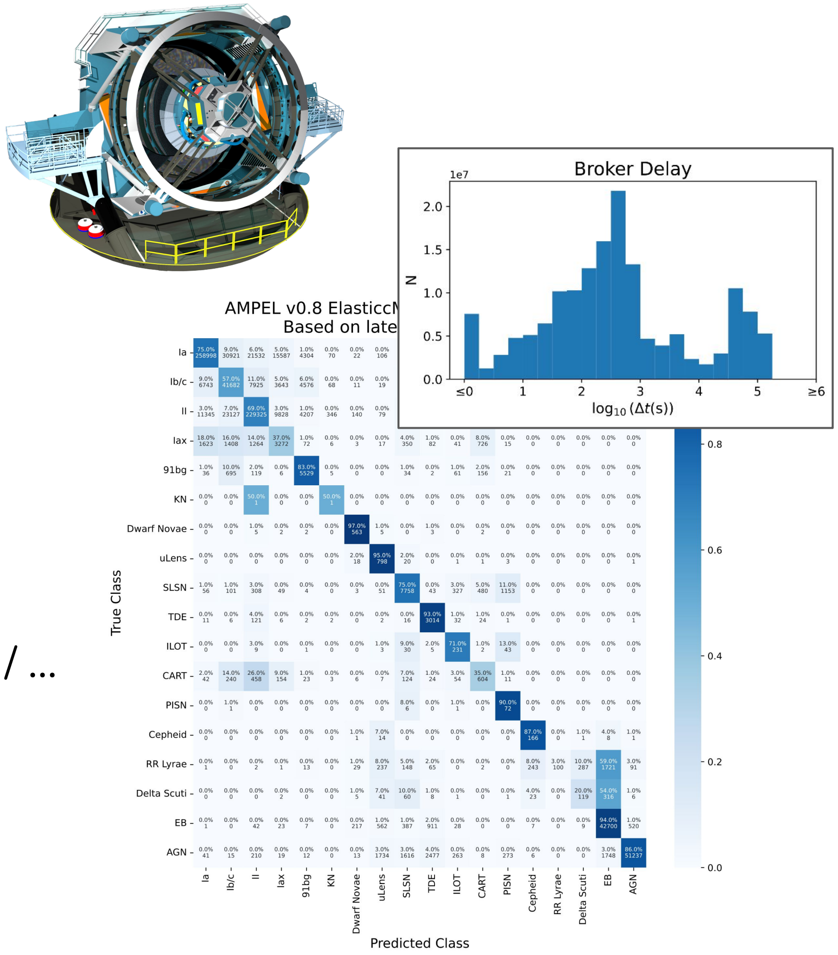
Vera Rubin Observatory:

- 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.



True Class \ Predicted Class	Ia	Ib/c	II	Iax	91bg	KN	Dwarf Novae	uLens	SLSN	TDE	ILOT	CART	PISN	Cepheid	RR Lyrae	Delta Scuti	EB	AGN
Ia	75.0% 258998	9.0% 30921	6.0% 21532	5.0% 15587	1.0% 4304	0.0% 70	0.0% 22	0.0% 106										
Ib/c	9.0% 6743	57.0% 41682	11.0% 7925	5.0% 3643	6.0% 4576	0.0% 68	0.0% 11	0.0% 19										
II	3.0% 11345	7.0% 23127	69.0% 229325	3.0% 9828	1.0% 4207	0.0% 346	0.0% 140	0.0% 79										
Iax	18.0% 1623	16.0% 1408	14.0% 1264	37.0% 3272	1.0% 72	0.0% 6	0.0% 3	0.0% 17	4.0% 350	1.0% 82	0.0% 41	8.0% 726	0.0% 15	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
91bg	1.0% 36	10.0% 695	2.0% 119	0.0% 6	83.0% 5529	0.0% 5	0.0% 0	0.0% 0	1.0% 34	0.0% 2	1.0% 61	2.0% 156	0.0% 21	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
KN	0.0% 0	0.0% 0	50.0% 1	0.0% 0	0.0% 0	50.0% 1	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
Dwarf Novae	0.0% 0	0.0% 0	1.0% 5	0.0% 2	0.0% 2	0.0% 0	97.0% 563	1.0% 5	0.0% 0	1.0% 3	0.0% 0	0.0% 2	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
uLens	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	2.0% 18	95.0% 798	2.0% 20	0.0% 0	0.0% 1	0.0% 1	0.0% 3	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 1
SLSN	1.0% 56	1.0% 101	3.0% 308	0.0% 49	0.0% 4	0.0% 0	0.0% 3	0.0% 51	75.0% 7758	0.0% 43	3.0% 327	5.0% 480	11.0% 1153	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
TDE	0.0% 11	0.0% 6	4.0% 121	0.0% 6	0.0% 2	0.0% 2	0.0% 0	0.0% 2	0.0% 16	93.0% 3014	1.0% 32	1.0% 24	0.0% 1	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 1
ILOT	0.0% 0	0.0% 0	3.0% 9	0.0% 0	0.0% 1	0.0% 0	0.0% 0	1.0% 3	9.0% 30	2.0% 5	71.0% 231	1.0% 2	13.0% 43	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
CART	2.0% 42	14.0% 240	26.0% 458	9.0% 154	1.0% 23	0.0% 3	0.0% 6	0.0% 7	7.0% 124	1.0% 24	3.0% 54	35.0% 604	1.0% 11	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
PISN	0.0% 0	1.0% 1	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	8.0% 6	0.0% 0	1.0% 1	0.0% 0	90.0% 72	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
Cepheid	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	1.0% 1	7.0% 14	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	87.0% 166	0.0% 0	1.0% 1	4.0% 8	1.0% 1
RR Lyrae	0.0% 1	0.0% 0	0.0% 2	0.0% 1	0.0% 13	0.0% 0	1.0% 29	8.0% 237	5.0% 148	2.0% 65	0.0% 0	0.0% 2	0.0% 0	8.0% 243	3.0% 100	10.0% 287	59.0% 1721	3.0% 91
Delta Scuti	0.0% 0	0.0% 0	0.0% 0	0.0% 2	0.0% 0	0.0% 0	1.0% 5	7.0% 41	10.0% 60	1.0% 8	0.0% 1	0.0% 0	0.0% 1	4.0% 23	0.0% 0	20.0% 119	54.0% 316	1.0% 6
EB	0.0% 1	0.0% 0	0.0% 42	0.0% 23	0.0% 7	0.0% 0	0.0% 217	1.0% 562	1.0% 387	2.0% 911	0.0% 28	0.0% 0	0.0% 0	0.0% 7	0.0% 0	0.0% 9	94.0% 42700	1.0% 520
AGN	0.0% 41	0.0% 15	0.0% 210	0.0% 19	0.0% 12	0.0% 0	0.0% 13	3.0% 1734	3.0% 1616	4.0% 2477	0.0% 263	0.0% 8	0.0% 273	0.0% 6	0.0% 0	0.0% 0	3.0% 1748	86.0% 51237

Infrastructure “landscape”

APPEC

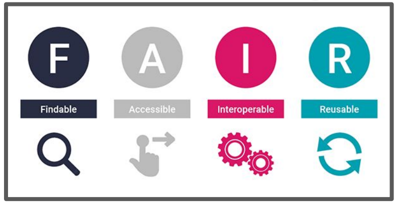
DZA

ESCAPE
European Science Cluster of Astronomy & Particle physics ESFRI research Infrastructures

SCiMMA

IVOA

PUNCH 4NFDI



green computing

ACME
Astrophysics Centre for Multi-messenger studies in Europe

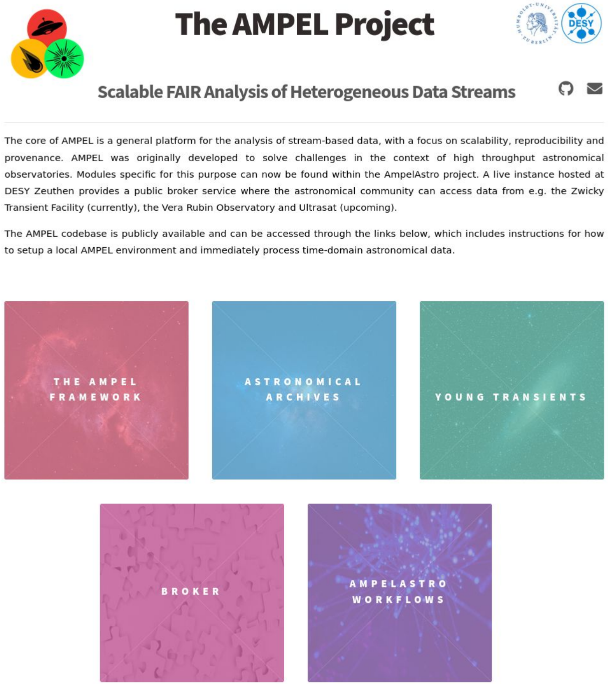
- Rapid technology / AI development.
- Provenance / FAIR / Green Computing *actually* matters.
- Multiple semi-parallel efforts: How to work efficiently?

Summary:

AMPEL realizes code-to-data in astronomy

- FAIR distribution point of e.g. VRO data
- Install and run locally:

https://ampelproject.github.io/ampelastro

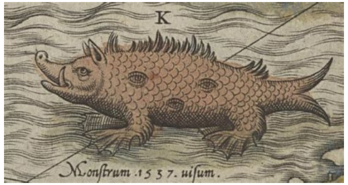


The AMPEL Project

Scalable FAIR Analysis of Heterogeneous Data Streams

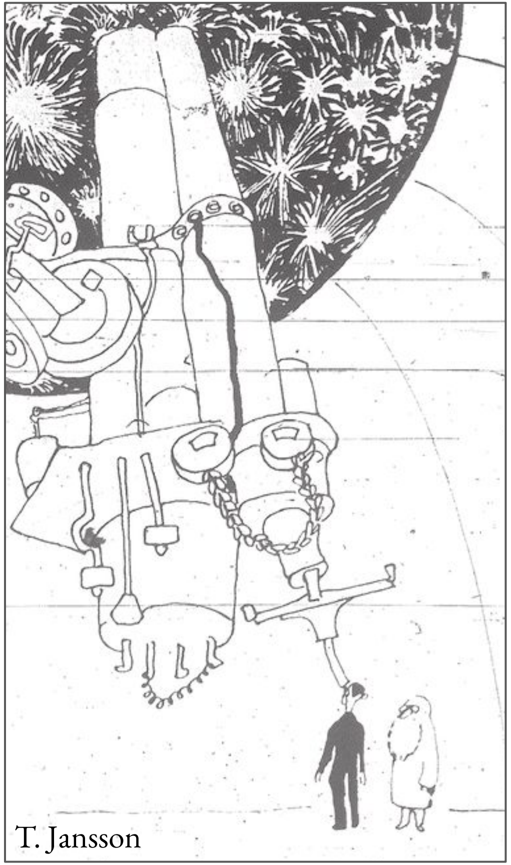
The core of AMPEL is a general platform for the analysis of stream-based data, with a focus on scalability, reproducibility and provenance. AMPEL was originally developed to solve challenges in the context of high throughput astronomical observatories. Modules specific for this purpose can now be found within the AmpelAstro project. A live instance hosted at DESY Zeuthen provides a public broker service where the astronomical community can access data from e.g. the Zwicky Transient Facility (currently), the Vera Rubin Observatory and Ultrasat (upcoming).

The AMPEL codebase is publicly available and can be accessed through the links below, which includes instructions for how to setup a local AMPEL environment and immediately process time-domain astronomical data.

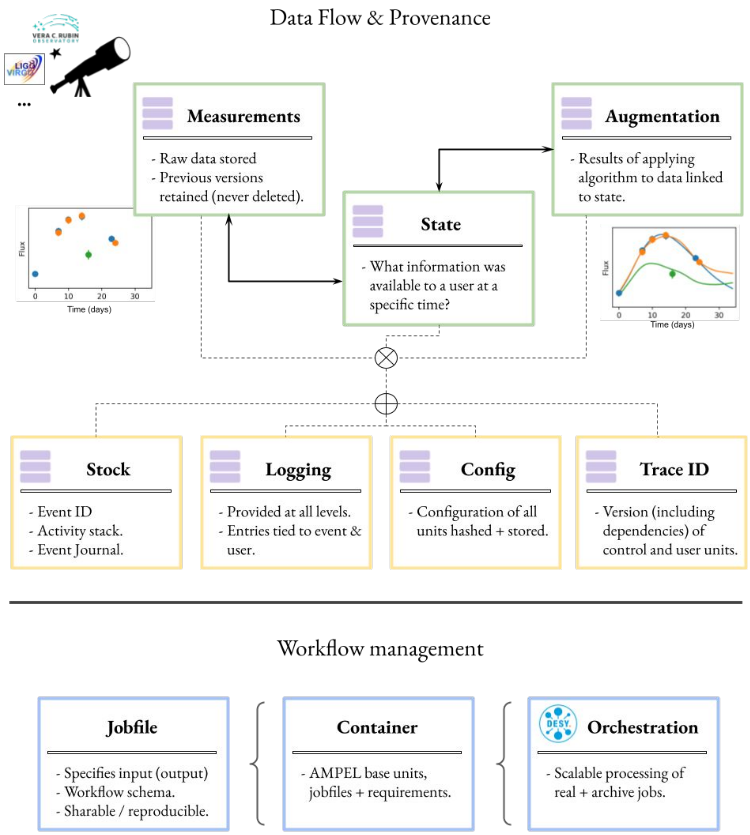


New tools are needed to allow creative exploration of large, heterogeneous data samples.

Backup



T. Jansson



Data Flow & Provenance

Measurements
- Raw data stored
- Previous versions retained (never deleted).

State
- What information was available to a user at a specific time?

Augmentation
- Results of applying algorithm to data linked to state.

Stock
- Event ID
- Activity stack.
- Event Journal.

Logging
- Provided at all levels.
- Entries tied to event & user.

Config
- Configuration of all units hashed + stored.

Trace ID
- Version (including dependencies) of control and user units.

Workflow management

Jobfile
- Specifies input (output)
- Workflow schema.
- Sharable / reproducible.

Container
- AMPEL base units, jobfiles + requirements.

Orchestration
- Scalable processing of real + archive jobs.

Bottom-Up Provenance

Modularity / Flexibility

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

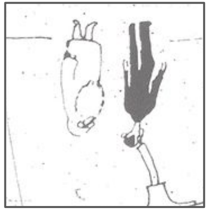
Provenance

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

Operational

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

Re vs Pro-active



“Inverted” work order:

① Get some data

② Develop analysis (software)

③ Conclude based on results

→

③ Define project hypothesis

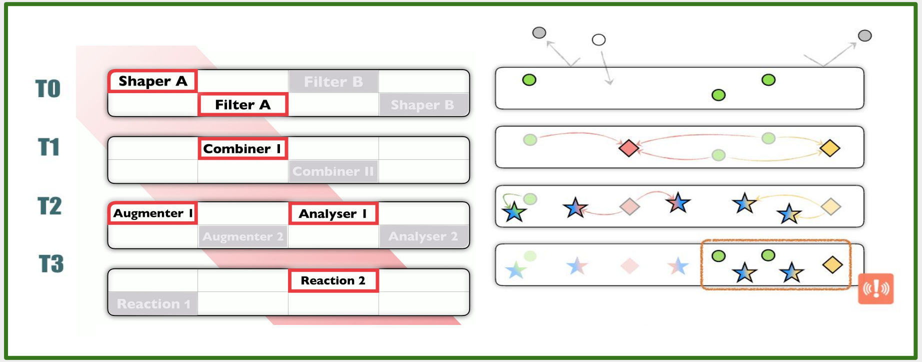
② Develop and test model

① Connect to data streams

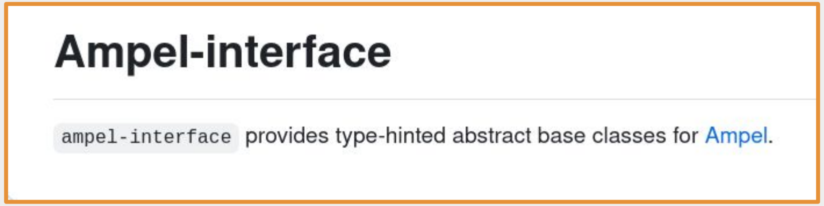
AMPEL x4:

②

Method for deconstructing time-domain analysis.



③



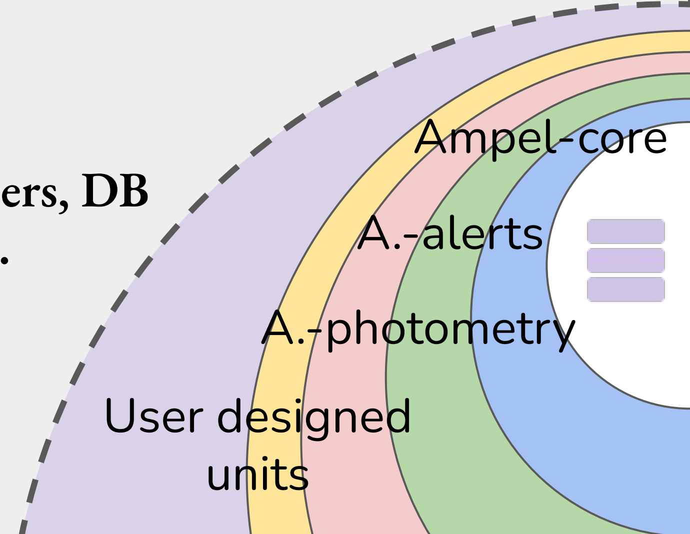
Ampel-interface

`ampel-interface` provides type-hinted abstract base classes for Ampel.

Interfaces between analysis units.

④

Execution tiers, DB and *workers*.



Ampel-core

A.-alerts

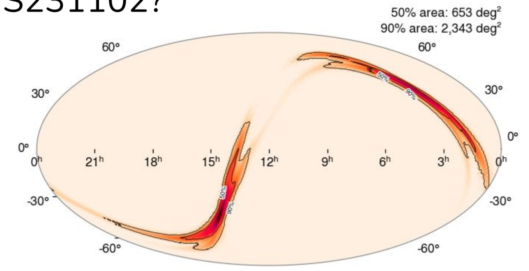
A.-photometry

User designed units

AMPEL

Workflow: KN search with LVK/ZTF

Any ZTF kilonova candidates from S231102?



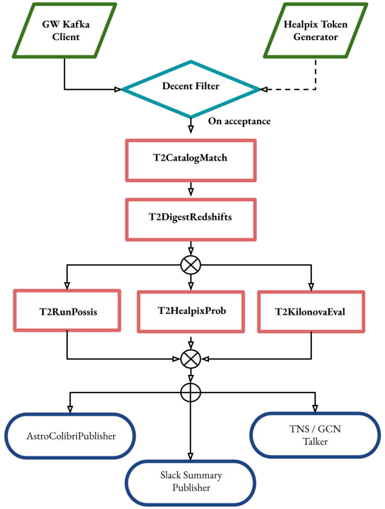
... no

Create a python 3.10 environment w. poetry and:

```
git clone https://github.com/AmpelAstro/Ampel-HU-astro.git
cd Ampel-HU-astro/
git checkout ampelgwO4
poetry install -E "ztf sncosmo extcats notebook"
ampel config build -out ampel_conf.yaml >& ampel_conf.log
ampel job --config ampel_conf.yaml --schema remote/lvk_S231102w.yml --secrets vault.yaml*
```

*Get access token to put in vault from JN.

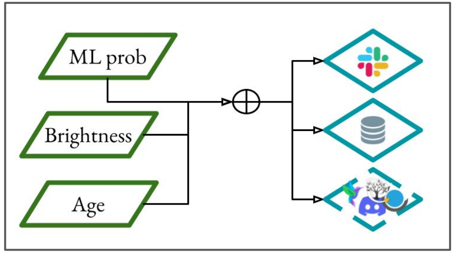
Workflow: ML classification of LSST simulation

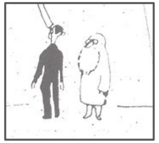


Try it out (with python3.10 + poetry):

- `git clone --branch elasticc2 https://github.com/AmpelAstro/Ampel-HU-astro.git`
- `cd Ampel-HU-astro/`
- `poetry install -E "ztf sncosmo extcats notebook elasticc"`
- `ampel config build -out ampel_conf.yaml >& ampel_conf.log`
- `ampel job --config ampel_conf.yaml --schema examples/elasticc_alerttar.yml`

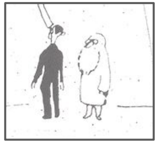
Will install a local AMPEL environment, download sample ELAsTiCC alerts and run three ML classifiers.





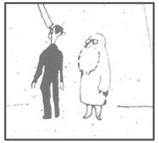
Exercise: From Idea to Implementation

Requirements: **Flexibility** **CS algorithms** **Scalable** **Provenance**



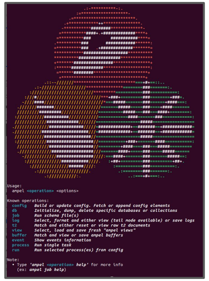
Exercise: From Idea to Implementation

Requirements:	**Flexibility**	**CS algorithms**	**Scalable**	**Provenance**
↓				
Implications:	**Modularity**	**Interfaces, [python]**	**DB / unit independence**	**Immutability, workflow, …**

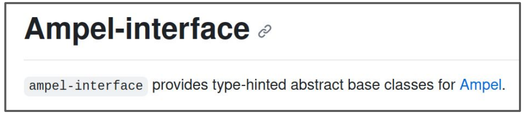


Exercise: From Idea to Implementation

Requirements:	Flexibility	CS algorithms	Scalable	Provenance
Implications:	Modularity	Interfaces, [python]	DB / unit independence	Immutability, workflow, ...
AMPEL:	Local job, common tools	Base classes, git integration	Four distinct action tiers	The State

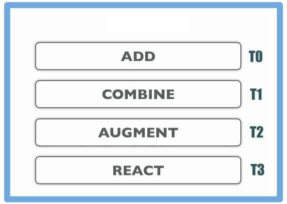


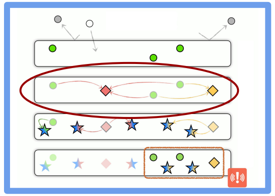
```
Usage:
  ampel <operation> <options>

Known operations:
  config   Build or update config. Fetch or append config elements
  db       Initialize, dump, delete specific databases or collections
  job      Run schema file(s)
  log      Select, format and either view (tail mode available) or save logs
  t2       Match and either reset or view raw t2 documents
  view     Select, load and save fresh "ampel views"
  buffer   Match and view or save ampel buffers
  event    Show events information
  process  Run single task
  run      Run selected process(es) from config

Note:
  • Type 'ampel <operation> help' for more info
    (ex: ampel job help)
```

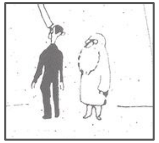


Ampel-interface

`ampel-interface` provides type-hinted abstract base classes for Ampel.







Exercise: From Idea to Implementation

Requirements: Flexibility | CS algorithms | Scalable | Provenance

Implications: Modularity | Interfaces, [python] | DB / unit independence | Immutability, workflow, ...

AMPEL: Local job, common tools | Base classes, git integration | Four distinct action tiers | The State

Lessons:

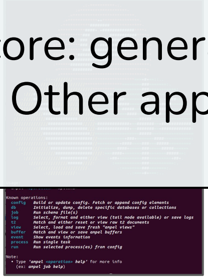
Combining speed, flexibility and provenance is hard...

... mix of CS, domain knowledge & information theory...

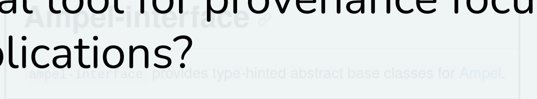
... but possible. 30k lines of code (V. Brinnel!)

Not suitable for everything. Scope important.

Ampel-core: general tool for provenance focus analysis of data streams. Other applications?

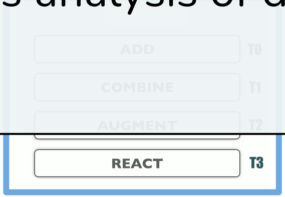


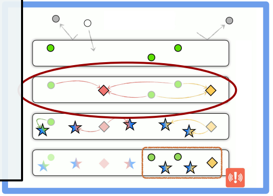
```
Usage:

Known operations:
  config   Build or update config. Fetch or append config elements
  db       Initialize, dump, delete specific databases or collections
  job      Run schema file(s)
  log      Select, format and either view (tail mode available) or save logs
  t2       Match and either reset or view raw t2 documents
  view     Select, load and save fresh "ampel views"
  buffer   Match and view or save ampel buffers
  event    Show events information
  process  Run single task
  run      Run selected process(es) from config

Note:
  • Type 'ampel <operation> help' for more info
    (ex: ampel job help)
```



Ampel-interface

ampel-interface provides type-hinted abstract base classes for Ampel.



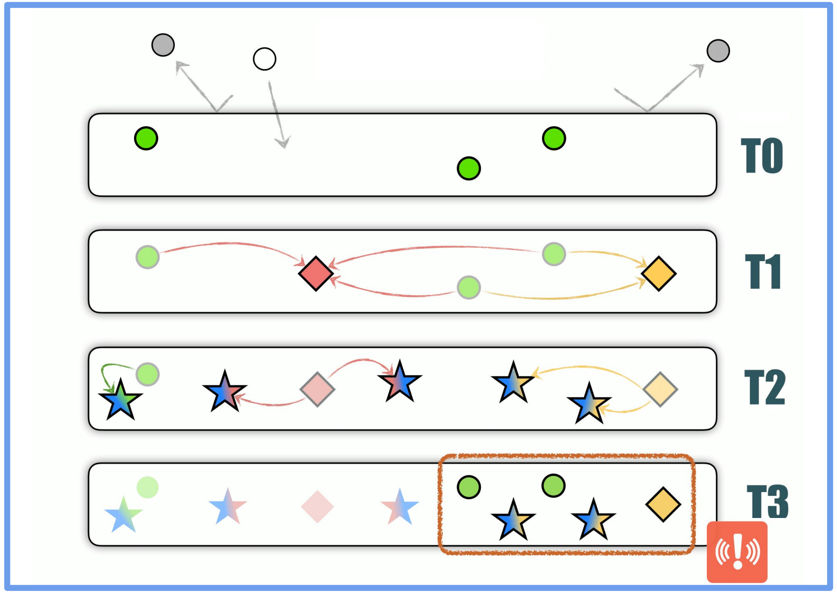


1 Deconstructing scientific workflows.

Modularity requires rules for how modules interact.

Theory: workflows can be constructed from operations of four different kinds.

Four separate execution layers, each distinguished by input/output types.

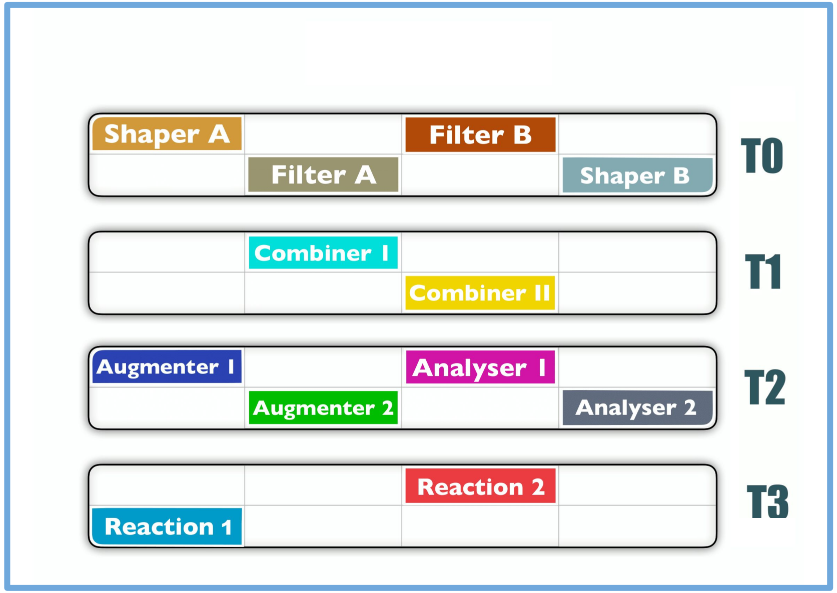


2 Merging user contributions

User:

Implements operations as units bound to a Tier.

Collects a series of units into a channel.

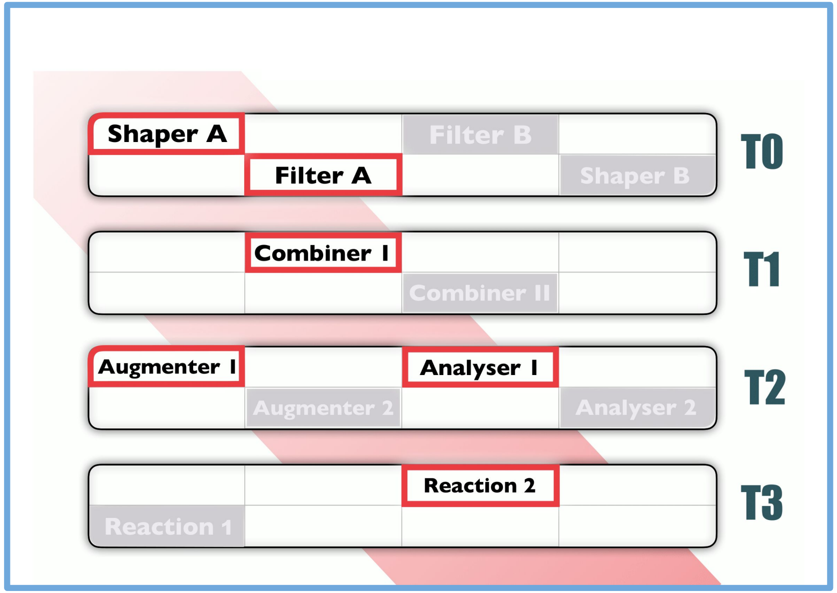


2 Merging user contributions

User:

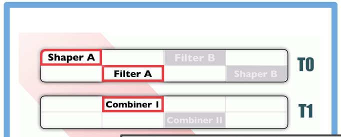
Implements operations as units bound to a Tier.

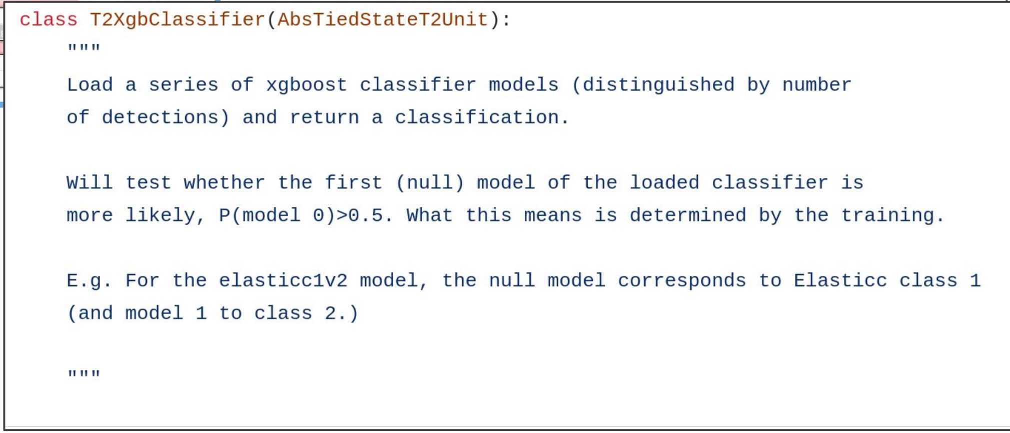
Collects a series of units into a channel.



Shaper A		Filter B		T0
	Filter A		Shaper B	
	Combiner I			T1
		Combiner II		
Augmenter I		Analyser I		T2
	Augmenter 2		Analyser 2	
		Reaction 2		T3
Reaction 1				

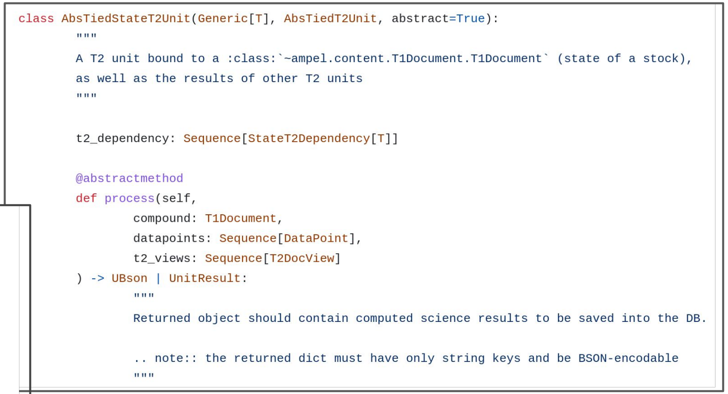
2 Merging user contributions





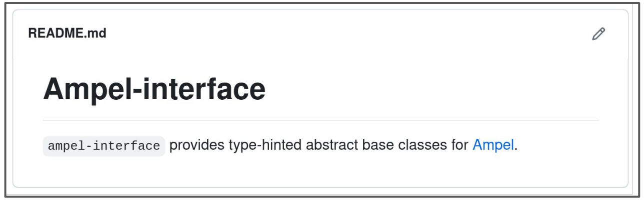
```python
class T2XgbClassifier(AbsTiedStateT2Unit):
    """
    Load a series of xgboost classifier models (distinguished by number
    of detections) and return a classification.

    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.)

    """
```

Unit: python module inheriting from Ampel-interface class.



```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
                """
```

Class methods determine I/O.



README.md

Ampel-interface

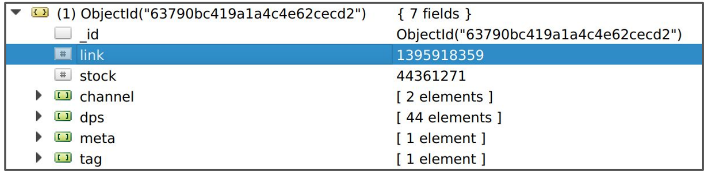
`ampel-interface` provides type-hinted abstract base classes for Ampel.

3 States: key to provenance and deduplication

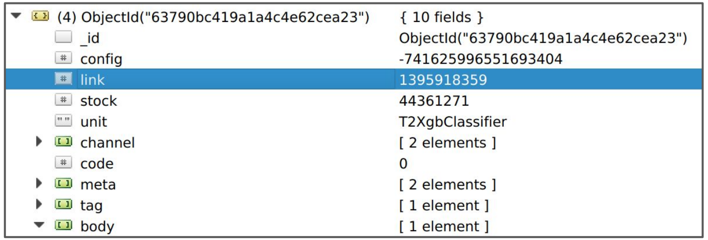
A *state* is the collection of data points associated with an (assumed) object at a specific time and visible to a specific user.

AMPEL systematically records every transient state as an immutable object.

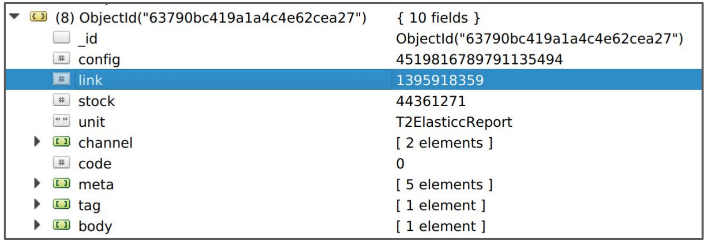
Every computation is directly tied to a state, a key component of the enforced data provenance model. Also enables deduplication - identical computations on the same state are elided.



```
▼ (1) ObjectId("63790bc419a1a4c4e62cecd2")   { 7 fields }
      _id                                     ObjectId("63790bc419a1a4c4e62cecd2")
      link                                    1395918359
      stock                                   44361271
    ▶ channel                                 [ 2 elements ]
    ▶ dps                                     [ 44 elements ]
    ▶ meta                                    [ 1 element ]
    ▶ tag                                     [ 1 element ]
```



```
▼ (4) ObjectId("63790bc419a1a4c4e62cea23")   { 10 fields }
      _id                                     ObjectId("63790bc419a1a4c4e62cea23")
      config                                  -741625996551693404
      link                                    1395918359
      stock                                   44361271
      unit                                    T2XgbClassifier
    ▶ channel                                 [ 2 elements ]
      code                                    0
    ▶ meta                                    [ 2 elements ]
    ▶ tag                                     [ 1 element ]
    ▼ body                                    [ 1 element ]
```



```
▼ (8) ObjectId("63790bc419a1a4c4e62cea27")   { 10 fields }
      _id                                     ObjectId("63790bc419a1a4c4e62cea27")
      config                                  4519816789791135494
      link                                    1395918359
      stock                                   44361271
      unit                                    T2ElasticcReport
    ▶ channel                                 [ 2 elements ]
      code                                    0
    ▶ meta                                    [ 5 elements ]
    ▶ tag                                     [ 1 element ]
    ▶ body                                    [ 1 element ]
```

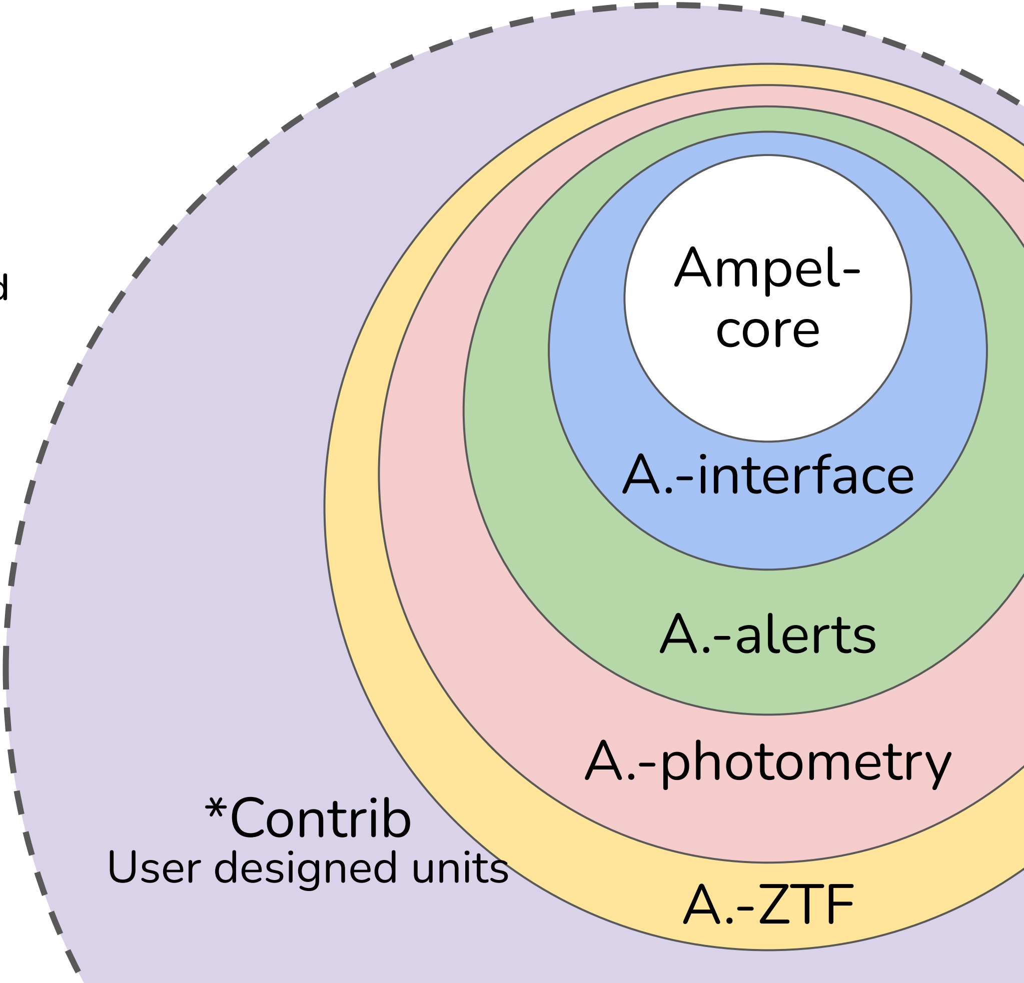
3 Core / Interface / DB

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


Ampel-core
A.-interface
A.-alerts
A.-photometry
*Contrib
User designed units
A.-ZTF