

# ZTF-II and Fritz:

## a collection of conversation starters

Dmitry A. Duev

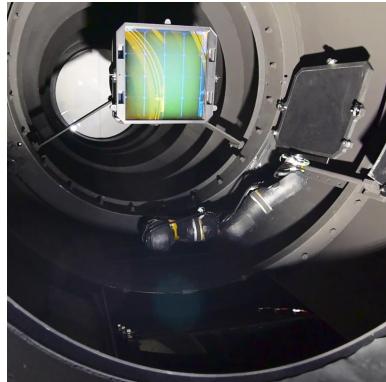
Research Scientist, Caltech

<https://duev.space>

On behalf of the Fritz/SkyPortal/Kowalski team

# Zwicky Transient Facility (ZTF)

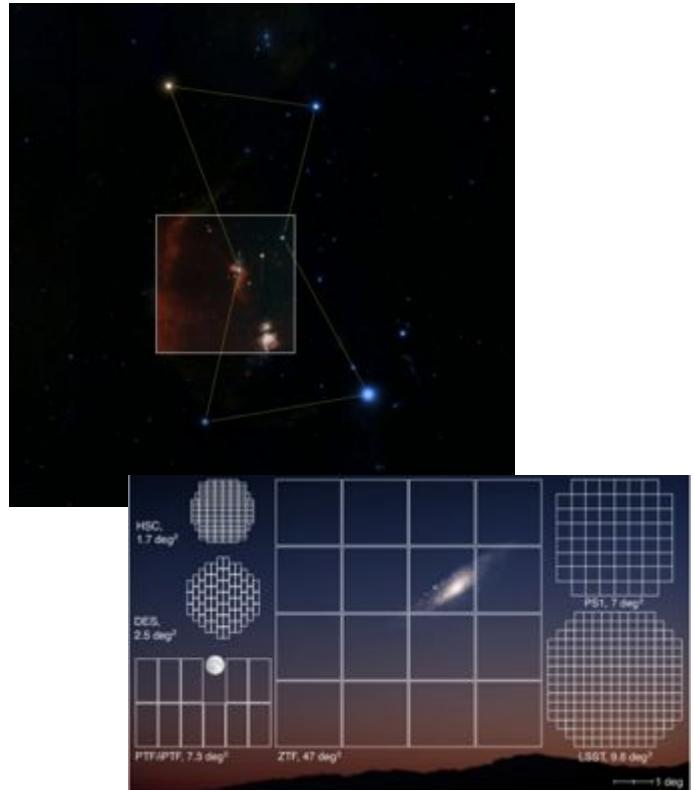
- Industrial-scale robotic optical sky survey
  - 48-inch Schmidt Telescope @ Palomar, CA
- Performs accurate measurements of billions of astronomical objects
- Registers ~million of transient events every night



# Zwicky Transient Facility at a glance

- Active detector area:  $\sim 47 \text{ deg}^2$
- Area survey rate: up to  $4300 \text{ deg}^2/\text{h}$
- Filters: g / r / i
- Nominal exposure time: 30 sec
- Single exp depth: 20.8 / 20.6 / 19.9
- Image quality (FWHM): 2.1" / 2.0" / 2.1"
- CCDs:  $16 \times 6\text{k} \times 6\text{k}$ , 1.0" / pix
- Survey entire Northern visible sky (to Decl. -30 deg) every couple nights

<https://ztf.caltech.edu>

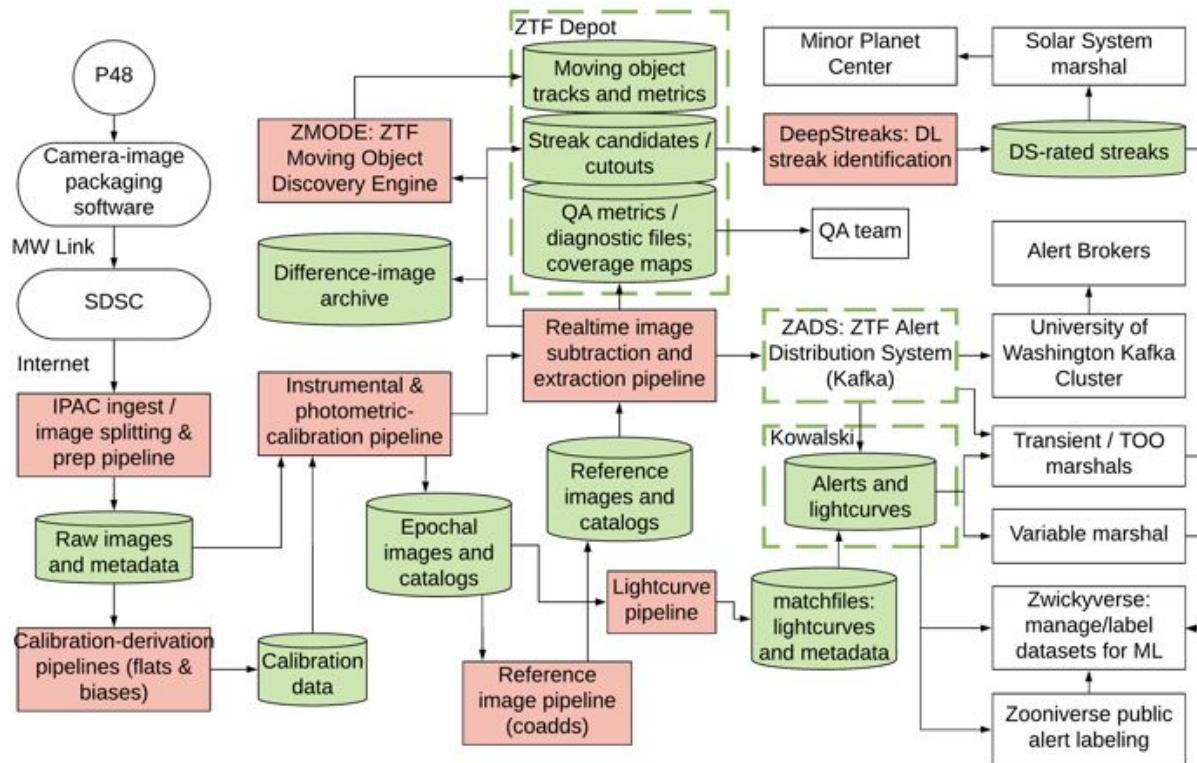


# ZTF Phase II

- Two night cadence public sky survey in  $g$  and  $r$
- Public data releases on a two-month cycle
- [Forced photometry service](#)
- 30-day forced photometry light curve is coming to alert packets!
  - <https://github.com/ZwickyTransientFacility/ztf-avro-alert/pull/7>
- ZTF Newsletter: Cosmic Newsflash
  - See first issue [here](#)
  - Please subscribe at <https://www.ztf.caltech.edu/page/ztf-newsletter>
- ZTF Summer School 2021
  - Applications due June 1
  - <https://www.ztf.caltech.edu/page/ztf-summer-school-2021>

We absolutely love the community and are thrilled to see all the different ways the ZTF data are used.

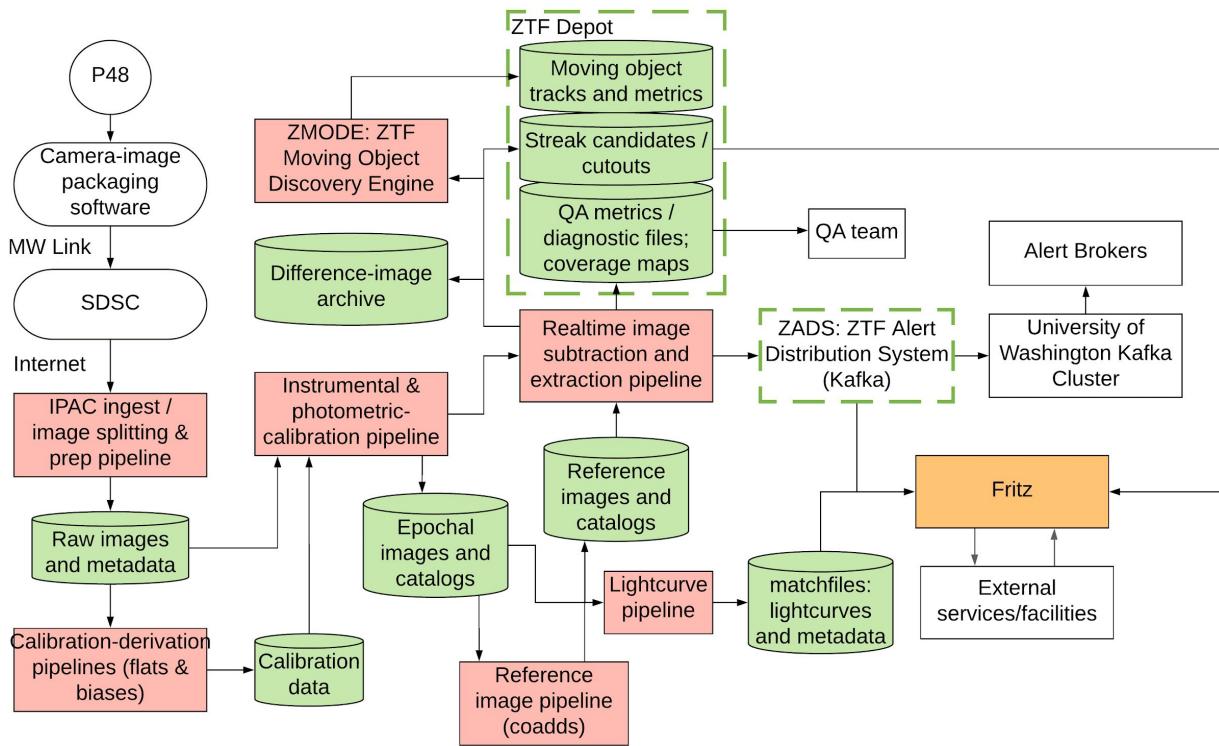
# ZTF-I: data/processing flow



ZTF acts as a discovery engine. Discoveries are followed-up using a wide range of instruments (including ZTF itself)

Single night	8h40m	Nominal survey	3 x 260 n
# on-sky exposures	~700	Volume of data products	~3.2 PB
Raw image data	~1 TB	Volume of ref images	~60 TB
Real-time data products	~4 TB	# CCD quad ref images	~ $2.8 \times 10^5$
# unvetted 5o alerts	~ $10^5 - 10^6$	Volume of matchfiles	~50 TB
# ML-vetted alerts	~ $10^3 - 10^5$	# matchfiles	~ $2.8 \times 10^5$
# unvetted streaks	~ $10^4 - 10^6$	# single-epoch PSF-fit source measurements	~800 B
# ML-vetted streaks	~ $10^2 - 10^3$	# single-epoch aperture source measurements	~230 B

# ZTF-II: data/processing flow



ZTF acts as a discovery engine. Discoveries are followed-up using a wide range of instruments (including ZTF itself)

Single night	8h40m	Nominal survey	3 x 260 n
# on-sky exposures	~700	Volume of data products	~3.2 PB
Raw image data	~1 TB	Volume of ref images	~60 TB
Real-time data products	~4 TB	# CCD quad ref images	~ $2.8 \times 10^5$
# unvetted 5o alerts	~ $10^5 - 10^6$	Volume of matchfiles	~50 TB
# ML-vetted alerts	~ $10^3 - 10^5$	# matchfiles	~ $2.8 \times 10^5$
# unvetted streaks	~ $10^4 - 10^6$	# single-epoch PSF-fit source measurements	~800 B
# ML-vetted streaks	~ $10^2 - 10^3$	# single-epoch aperture source measurements	~230 B

# Fritz: science data platform for ZTF-II

- Scalable, API-first system, with fine-grained access control
- Multi-survey data archive and alert broker
- Interactive, mobile-friendly collaborative platform for transient, variable, and Solar system science cases
- Workhorse for ML applications: classification and labeling at scale
- Follow-up observation management: robotic and classical facilities

Initiated in Feb 2020  
Beta up in Sep 2020  
MVP live in Nov 2020

The screenshot displays the Fritz web application's main dashboard. On the left is a sidebar with navigation links: Dashboard, Sources, Candidates, Favorites, Alerts, Groups, Observing Runs, and About. Below this is a search bar labeled 'Source'. The main area is divided into several sections:

- Recently Saved Sources:** A list of six sources saved within the last 7 hours, each with a thumbnail image, coordinates, and a brief description.
- News Feed:** A list of five recent news items, each with a thumbnail, source, and timestamp.
- Top Sources:** A list of three sources with the highest activity, each with a thumbnail, source, and timestamp.
- My Groups:** A sidebar listing groups: Fritz TNS, Redshift Completeness Factor, Sitewide Group, Treasures Public, and Palomar 1.2m Oschin.
- Forecast:** A weather forecast for Palomar 1.2m Oschin, showing current conditions and a 24-hour outlook.

<https://github.com/fritz-marshall/fritz>

<https://docs.fritz.science>

# Fritz: meet the core team



Joshua Bloom



Michael Coughlin



Arien Crellin-Quick



Dmitry Duev

Initially, a  
Caltech+Berkeley initiative



Daniel Goldstein



Matthew Graham



Mansi Kasliwal



Don Neill



Guy Nir



Kyung Min Shin



Leo Singer

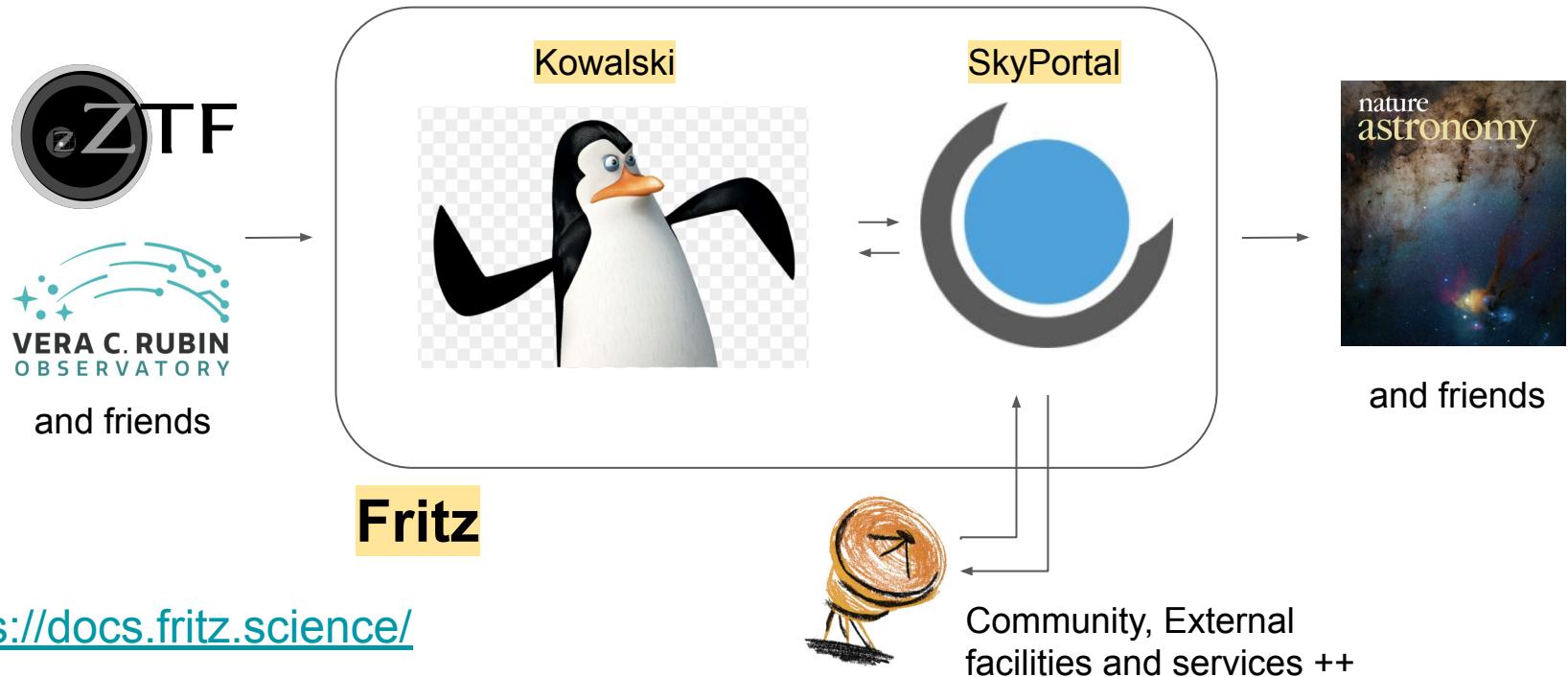


Stéfan van der Walt

...But many others have  
contributed as well -  
open-source at its best

# Fritz: Schematic overview

Observe → Mine/Discover → Study and Characterize → Profit!



# Fritz: features

- Open source (free to use, modify, and distribute)
- API-first system: rich APIs for machine usage
- Powerful alert stream enhancement/filtering capabilities
- Extendible & scalable design: async Python backends, React/Redux frontend
- Fine-grained access control
- Authentication via OAuth
- Real-time Slack-like messaging, notifications
- Rich visualization capabilities
- Follow-up management
- Distributed computation via Dask
- Docker compose or Kubernetes deployment
- Well-tested, extensive docs, CI/CD

See the [tutorial](#) presented  
at this workshop

# Fritz: features

- Open source (free to use, modify, and distribute)
  - GitHub, Issues, PRs, ZenHub, code review, CI/CD

skyportal / skyportal

[Code](#) [Issues 195](#) [Pull requests 19](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

[master](#) [14 branches](#) [0 tags](#) [Go to file](#) [Add file](#) [Code](#)

**About**  
Collaborative platform for time-domain astronomy  
skyportal.io  
astrometry transient-astronomy  
variable-stars fast  
machine-learning collaborative-research  
Reading  
View license

**Releases**  
Create a new release

**Packages**  
No packages published  
Publish your first package

**Contributors** 18

**Languages**  
Python 63.7% JavaScript 35.3% CSS 1.0% Shell 0.2% Dockerfile 0.1%

[README.md](#)

fritz-marshall

[Search or jump to...](#) [Pull requests](#) [Issues](#) [Codespaces](#) [Marketplace](#) [Explore](#)

[Repositories 6](#) [Packages](#) [People 11](#) [Teams 1](#) [Projects](#) [Settings](#)

[Find a repository...](#) [Type: All](#) [Language: All](#) [Customize pins](#) [New](#)

**fritz-deploy** Private Deployment scripts  
Python 7 0 1 (1 issue needs help) Updated 15 hours ago

**fritz** The Zwicky Transient Facility Phase II Marshal.  
astrometry broker marshall time-domain variable-stars transient-astronomy ztf

**fritz-beta-feedback** A place for beta testers to discuss issues or request new features related to the Fritz Marshal  
Python 12 9 26 (1 issue needs help) Updated 15 hours ago

**skyportal** Forked from [skyportal/skyportal](#) Collaborative platform for time-domain astronomy  
Python 27 0 0 11 Updated 16 days ago

**doc** Generated documentation  
HTML 20 2 0 0 0 Updated 20 days ago

**fritz-marshall.github.io** Deployed website  
HTML 0 0 0 0 0 Updated on May 18

dmitryduev / kowalski

[Code](#) [Issues 6](#) [Pull requests 4](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

[master](#) [4 branches](#) [0 tags](#) [Do file](#) [Add file](#) [Code](#)

**About**  
Kowalski: a tool for time-domain astronomy  
Readme MIT License

**Releases**  
No releases published Create a new release

**Packages**  
No packages published Publish your first package

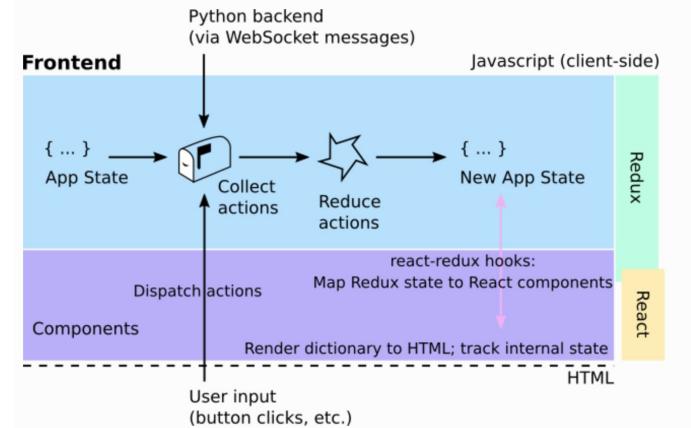
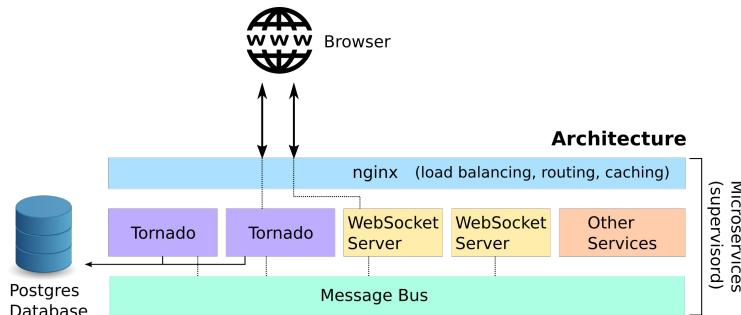
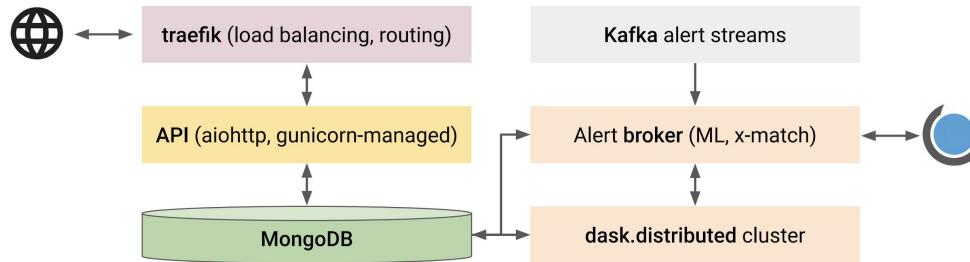
**Contributors** 4  
dmitryduev Dmitry Duev  
dannygoldstein Danny Goldstein  
stefanv Stefan van der Walt

**Languages**  
Python 99.0% Dockerfile 1.0%

**Kowalski** [View on GitHub](#)

# Fritz: features

- Extendible & scalable design: async Python backends, React/Redux frontend



# Fritz: features

- API-first system: rich APIs for machine usage

Python

```
import requests

token = 'ea70a5f0-b321-43c6-96a1-b2de225e0339'

def api(method, endpoint, data=None):
    headers = {'Authorization': f'token {token}'}
    response = requests.request(method, endpoint, json=data, headers=headers)
    return response

response = api('GET', 'http://localhost:5000/api/sysinfo')

print(f'HTTP code: {response.status_code}, {response.reason}')
if response.status_code in (200, 400):
    print(f'JSON response: {response.json()}'
```

Command line (curl)

```
curl -s -H 'Authorization: token ea70a5f0-b321-43c6-96a1-b2de225e0339' http://
```

Response

In the above examples, the SkyPortal server is located at <http://localhost:5000>. In case of success, the HTTP response is 200:

```
HTTP code: 200, OK
JSON response: {'status': 'success', 'data': {}, 'version': '0.9.dev0+git20200
```

The screenshot shows two side-by-side API documentation pages from Fritz.

**Left Panel: Add a new source**

REQUEST BODY SCHEMA: application/json

```
root
  > q_search... (highlighted)
  > del_source...
  > update_source...
  > retrieve_all_sources...
  > add_new_source...
  > send_out_a_new_source_notification...
  > retrieve_a_spectrum...
  > update_spectrum...
  > delete_a_spectrum...
  > upload_spectrum...
  > pane_spectrum_from_ASCII_file...
```

Request samples (Payload):

```
{ "ra": 0,
  "dec": 0,
  "name": "test",
  "ra_err": 0.001,
  "dec_err": 0.001,
  "ra_dia": 0.0001,
  "dec_dia": 0.0001,
  "z": 10000 }
```

**Right Panel: Query Kowalski**

REQUEST BODY SCHEMA: application/json

```
root
  > auth...
  > users...
  > queries...
  > Query_Kowalski... (highlighted)
  > filters...
  > lab...
  > Documentation_Pinned_to_Doc...
```

Request samples (Payload):

```
{ "query_type": "STANDARD",
  "aggregate": true,
  "q": "name: 'Cassiopeia A' AND type: 'galaxy' AND redshift < 10000",
  "catalog": "skycat_v1",
  "sort_by": "redshift",
  "order": "ASC" }
```

Responses (Example):

```
HTTP/1.1 200 OK
Content-type: application/json
Example: {"count": 1, "results": [{"id": 1, "name": "Cassiopeia A", "ra": 215.0, "dec": 69.0, "redshift": 10000.0, "type": "galaxy"}]}
```

## Complete OpenAPI specs:

<https://docs.fritz.science/api.html>

<https://skyportal.io/docs/api.html>

<https://kowalski.caltech.edu/docs/api>

# Fritz: features

- Fine-grained access control
  - Custom row-level security (RLS) framework in the API layer
  - ACLs, Roles; Groups
- Authentication via OAuth
- Real-time Slack-like messaging, (email/text)
- Rich visualization capabilities
- Follow-up observations management (aka TOM)
  - Both robotic and classical facilities (SEDM end-to-end)
- Distributed computation via Dask
- Docker compose or Kubernetes deployment
  - Everything containerized / versioned
  - Production deployment: GCP + Local
- Well-tested, extensive docs, CI/CD, staging

This screenshot shows the Fritz interface's main dashboard. It includes sections for 'Recently Saved Sources' (listing several recent observations), 'Top Sources' (showing counts for various sources), and a 'News Feed' (displaying recent updates from various sources). A detailed view of a source, 'Palmer - 2m Ouchin', is shown at the bottom, providing information about its classification (Type I), coordinates, and a forecast chart.

This screenshot shows another view of the Fritz dashboard. It displays 'Recently Saved Sources', 'News Feed' (with entries like 'AGN' and 'New entry'), and 'Top Sources'. The 'News Feed' section includes a 'Forecast' section for 'Palmer - 2m Ouchin'.

This screenshot provides a detailed view of a specific source, 'ZTF21aanwmxn'. It includes a 'Finding Chart' (with sub-sections for NEW, RFT, SLO, RPT, and PFT), an 'Auto-annotations' table (listing various parameters like Origin, Key, Value, Author, and Cred), and a 'Comments' section where users can leave notes.

# Fritz: features

- Fast, robust multi-survey data archive
  - ZTF alert stream
  - ZTF light curves + features + SCoPe classifications
  - >30 external catalogs
- Powerful alert stream enhancement & filtering capabilities
  - MongoDB aggregation pipelines
  - Performant for filters of different complexity
  - Simple dask.distributed-based service(s)
  - Public alert DBs for filter design/debugging
  - Multiple ML models
  - Cross-matches
  - Filtering enhanced data
  - Automated checks, no filter code audit
  - [Can post results from external brokers]

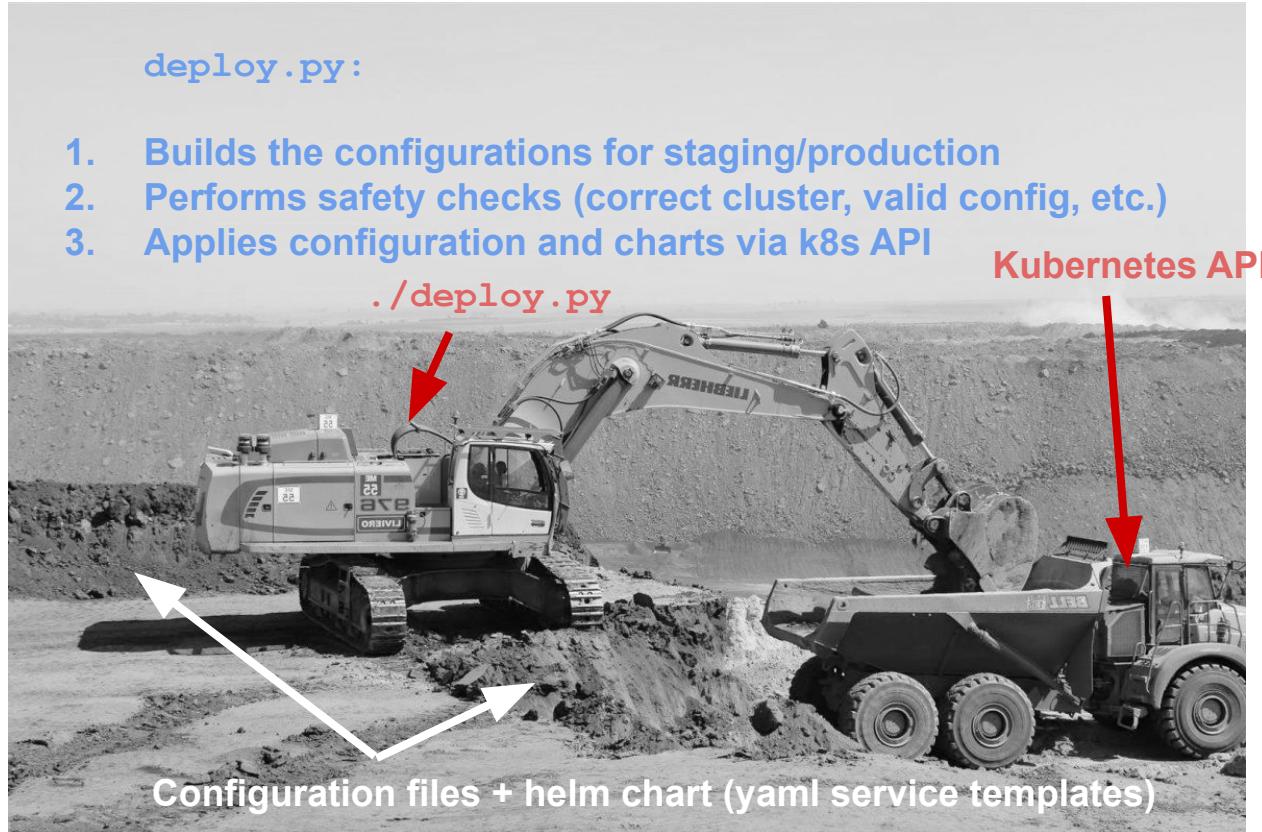
The screenshot shows the Fritz application's main dashboard. On the left is a sidebar with links to Dashboard, Sources, Candidates, Favorites, Alerts, Groups, Observing, Runs, and About. Below this is a search bar labeled "Source". The main area is divided into several sections:

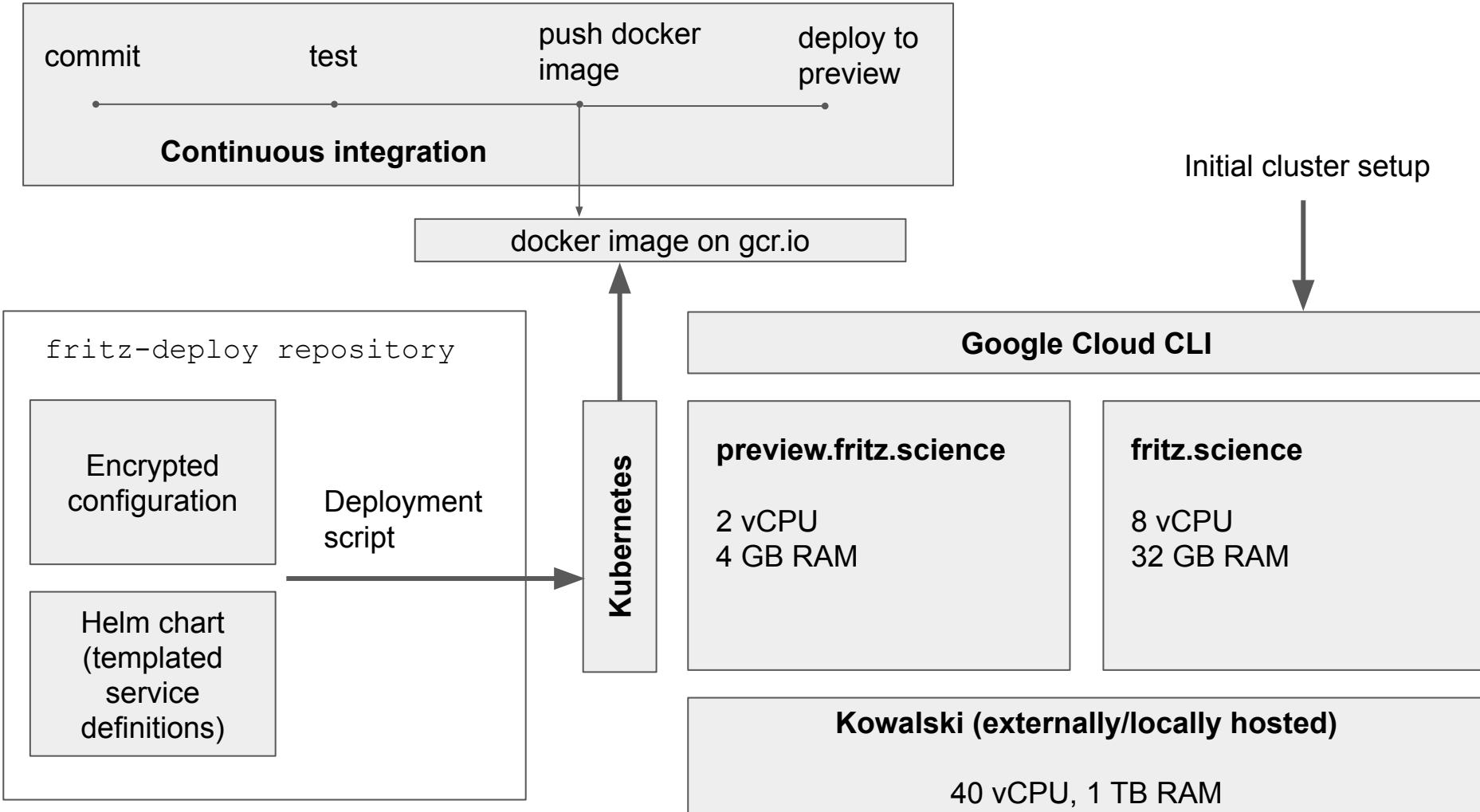
- Recently Saved Sources:** A grid of thumbnail images of astronomical objects with their names and IDs.
- News Feed:** A list of recent activity items, each with a thumbnail, title, source, and timestamp.
- Top Sources:** A list of the most viewed sources, including their names, IDs, view counts, and thumbnails.
- My Groups:** A section showing "Fritz TNS", "Sitemwide Group", "Treasures", and "Public". It also displays a count of 492 new sources over the last 7 days.
- Keck II 10m:** A weather and sky conditions summary for Keck II 10m.

This screenshot shows the "New Connection" dialog box for MongoDB. It includes fields for "Host", "Port", "Database", "Auth Source", and "Auth Mechanism". There are tabs for "Favorites" and "Recent". A "CONNECT" button is at the bottom right. To the right of the dialog is a "New Connection" sidebar with instructions for connecting to Atlas and a "CREATE FREE CLUSTER" button. Further right is a "New to Compass" sidebar with information about clusters and a "CONNECT" button. At the bottom right is a JSON preview pane showing Fritz's connection details:

```
_id: "845141035015010004_ZTF19aarzgrw"
schemasvsn: "3.2"
publisher: "ZTF (www.ztf.caltech.edu)"
objectId: "ZTF19aarzgrw"
candid: 845141035015010004
candidate: Object
  jd: 2458599.6410301
  fid: 2
  pid: 845141035015010004
  diffmaglim: 19.739463806152344
...
> prv_Candidate: Array
> cutoutScience: Object
> cutoutTemplate: Object
> cutoutDifference: Object
> coordinates: Object
```

# Fritz: deployment





# Fritz: some stats

- Fritz: initiated in Feb 2020, beta up Sep 2020, MVP live Nov 2020
- SkyPortal DB (~300GB)
  - 530k Objects (Candidates), 46k Sources
  - 100M photometry points
  - 2k spectra
  - 170 users worldwide
  - 86 groups, 30 use Kowalski filters
- Kowalski DB (~50TB)
  - ~380M alerts
  - ~4B light curves
  - Typically serves ~100M queries/day

# LLs: open source is ❤️

- Fritz/SkyPortal/Kowalski is open source
  - A huge part of devs is volunteer labor of love
  - While the core dev team is relatively small (in terms of effective person-hours), dozens of people have contributed meaningfully
- Leveraging what GitHub/OSS has to offer
  - Issues to track bugs and feature requests
  - PRs + thorough code review
  - GitHub Actions as the CI/CD platform
  - Don't argue about style, enforce pre-commit hook (black, flake8, eslint...)
- Project management tool: ZenHub
  - Should be as close as possible to GH

# LLs: team and community is ❤️

- Extraordinary individuals with a broad range of expertise
  - Respectful + supportive environment (that naturally extends to mutual admiration) allows the team to go far (check out [this article](#) for the spirit)
  - Critical but fair and open-minded review of ideas and code allows to iterate fast, converging on better solutions
  - Staying in sync: Slack + Daily 15-min stand-ups + weekly 1h meetings
- Constructive community feedback is essential for success
  - Beta-testers!
  - Enormously useful for finding/fixing bugs and implementing new features
  - Need to be clear about the communication channels: a dedicated Slack channel for smaller issues + GH issue templates for larger stuff
  - Critically important: prioritization + clear big picture for the project



François Chollet ✅ @fchollet · Apr 12

Doing everything your users request is the shortest path to ruining your product -- especially when these users come from a legacy product and just want the new thing to be like the old thing

# LLs: testing

- You are not testing your code enough
  - No, you're not!
  - From unit testing to integration testing through API and frontend, every bit is essential
  - Is that docker image still building from scratch? Note the word “continuous” in CI
  - Never underestimate the scale of a disaster that six innocently-looking lines of code can bring
  - Database migrations should be tested both ways - roll-backs are more common than we'd wish
  - Understand (and embrace!) flakiness
- Staging environment
  - Helps catch a lot of bugs before they have a chance to reach production, e.g. innocently-looking migrations that can take forever

# LLs: production is really hard

- Even harder is to deliver updates/new features to prod
  - Testing *is* your friend, but it won't catch everything that can happen
    - Running a subset of the test suite on a read-only replica of the prod db
  - Weekly (at least) deployment to prod
  - Resilient infrastructure for deployment
    - Monitoring the performance of the different components
    - Query Insights on the GCP
    - API endpoint response times, temporal evolution

