Technical Execution Plan: High-Throughput Archival Discovery of Minor Solar System Bodies

1. Executive Summary and Strategic Architecture

The landscape of planetary science has undergone a fundamental phase transition. For centuries, the discovery of Minor Solar System Bodies—Main Belt Asteroids (MBAs), Near-Earth Objects (NEOs), and Trans-Neptunian Objects (TNOs)—was strictly the domain of photon-gathering hardware. Discovery was rate-limited by the aperture of the telescope and the darkness of the site. Today, we operate in the petabyte era of synoptic surveys. The limiting factor is no longer photon collection, but data mining efficiency. For a computer engineer possessing skills in Python, Computer Vision (CV), and algorithm design, the "Virtual Observatory" offers a discovery potential that rivals, and often exceeds, that of physical mid-sized observatories.

This Technical Execution Plan (TEP) outlines a rigorous engineering framework for constructing a software-only discovery pipeline. The objective is to mine the massive, publicly available image archives of the Zwicky Transient Facility (ZTF) and Pan-STARRS (PS1) to identify previously unknown moving objects. Unlike real-time brokers that filter alert streams for bright transients, this architecture focuses on "Deep Archival Mining"—the systematic reprocessing of science images to uncover faint movers ($V > 20.5$) that standard pipelines may have rejected due to aggressive thresholding or complex backgrounds.

The proposed architecture operates on a **Shift-and-Stack** and **Difference Imaging** paradigm. It treats the sky not as a static wallpaper, but as a dynamic database of time-series pixel values. The pipeline is designed to be autonomous, scalable, and compliant with the rigorous astrometric standards of the International Astronomical Union (IAU).

1.1 The Virtual Observatory Paradigm

The central thesis of this execution plan is that the "telescope" is the database. The ZTF survey alone generates millions of detections per night. While its primary pipeline is highly efficient, it is optimized for throughput rather than depth or completeness in complex regions (e.g., the Galactic Plane). A custom pipeline allows the engineer to:

1. **Relax Detection Thresholds:** Standard pipelines might cut at 5σ. A custom pipeline can probe to 3σ by leveraging multi-epoch linking constraints to reject noise, effectively increasing the "virtual aperture" of the survey.
2. **Deploy Advanced CV Algorithms:** Implement computationally expensive subtraction kernels (e.g., ZOGY or Optimal Image Subtraction) that real-time pipelines cannot afford

to run.
3. **Specific Target Optimization:** Tailor tracking algorithms for non-standard motion vectors, such as the ultra-fast streaks of close-approach NEOs or the slow, retrograde motion of Distant Retrograde Orbits, which general-purpose "great circle" linkers often miss.

1.2 System Architecture Overview

The pipeline is composed of five modular subsystems, each designed to handle a specific stage of the data lifecycle:

- **Ingestion Engine (Data Acquisition):** An asynchronous harvester that interfaces with the IRSA (ZTF) and MAST (Pan-STARRS) APIs. It manages the retrieval of FITS files, ensuring appropriate calibration levels and metadata synchronization.
- **The Vision Core (Blinking/Subtraction):** The image processing heart of the system. It utilizes astropy, reproject, and sep to align images, subtract static backgrounds, and perform difference imaging to isolate transients.
- **Linkage & Association Engine:** A geometric solver that connects isolated detections across temporal baselines (Tracklets) to distinguish physical motion from random noise.
- **Validation Layer:** A rigorous filter employing the IMCCE SkyBoT service to identify known objects and Machine Learning (ML) classifiers to reject sensor artifacts (ghosts, glints, cosmic rays).
- **Orbit Determination & Reporting:** A bridge to the Minor Planet Center, utilizing Find_Orb for astrometric solving and formatting data into the modern ADES XML standard for submission.

This report details the implementation of each module, identifying specific Python libraries, algorithmic choices, and the "gotchas" inherent to high-precision astrometry.

---

2. Data Acquisition: Mining the ZTF and Pan-STARRS Archives

The fuel for any discovery pipeline is high-cadence imaging. A single image is static; discovery requires the dimension of time. Therefore, data acquisition is not merely about downloading files, but about constructing "manifests" of observations that satisfy specific temporal constraints (cadence).

2.1 The Zwicky Transient Facility (ZTF)

ZTF is the workhorse for this pipeline. Located at the Palomar Observatory, its 47 square-degree field of view scans the northern sky with a cadence unmatched by other public surveys.

2.1.1 ZTF Data Model and Access

The ZTF camera consists of 16 CCDs, each divided into 4 readout quadrants, resulting in 64 individual data channels per exposure. The fundamental unit of archival retrieval is the **CCD-Quadrant**.[1]

- **Pixel Scale:** 1.01 arcseconds/pixel.
- **Limiting Magnitude:** $R \approx 20.5$ (single exposure).
- **Exposure Time:** Typically 30 seconds.

The data is hosted by the **Infrared Science Archive (IRSA)**. Access is mediated through the IRSA API, which supports both lightcurve queries and bulk image retrieval.

Data Products of Interest:

1. **sciimg.fits (Science Image):** The primary calibrated image. This is the "New" image in the subtraction equation. It has been bias-subtracted, flat-fielded, and astrometrically calibrated.[3]
2. **mskimg.fits (Mask Image):** A critical file often ignored by novices. This bitmask image flags pixels affected by saturation, aircraft trails, and, crucially, optical ghosts.[4]
3. **sexcat.fits (Source Catalog):** A catalog of sources extracted by SExtractor during the IPAC pipeline processing. While the custom pipeline will perform its own extraction, this catalog serves as a vital cross-check for astrometric solution quality.

2.1.2 The Query Strategy: Finding the "Tracklet"

Randomly downloading images is inefficient. To discover a mover, one requires a **Tracklet**: a sequence of at least 3 detections of the same object within a short window (typically 30-90 minutes) on a single night.

- **The Triplet Constraint:** The pipeline must query the ZTF Metadata Table (ztf_current_meta_sci) to identify fields and nights where the cadence supports tracklet formation.
- **Query Logic:**
  - Select a Field ID (e.g., fields near the Ecliptic plane).
  - Group metadata by nightdate, ccdid, and qid.
  - Filter for groups where the number of exposures $N \ge 3$ and the time span $\Delta t < 3$ hours.
  - *Reasoning:* A minimum of 3 points is required to define a linear motion vector and reject random noise coincidences (2 points can always be connected by a line; 3 points confirm the line).

Python Implementation:
The astroquery.ipac.ztf module is the standard interface. Alternatively, direct URL construction allows for higher throughput and parallelization.

Python

```python
# Conceptual Python Workflow for Metadata Filtering
from astroquery.ipac.ztf import Ztf
from astropy.time import Time

# Define search parameters
field_id = 600 # Example field
time_range = Time(['2024-01-01', '2024-02-01'])

# Query Metadata
meta = Ztf.query_metadata(
    'ztf_current_meta_sci',
    sql_query=f"field={field_id} AND fid=2"  # fid=2 is r-band
)

# Logic to group by night and count exposures
#... (Implementation of GroupBy logic)...
```

2.1.3 Handling ZTF Reference Images

To perform difference imaging, a Reference Image (I_{ref}) is required. ZTF provides pre-coadded reference images.

- **Requirement:** For every Science Image (I_{sci}), identifying the corresponding Reference Image is mandatory.
- **Matching Logic:** The reference must match the field, ccdid, qid, and filtercode of the science image.
- **Access:** Reference images are stored in a separate directory structure at IRSA but are retrievable via the same API protocols.[3]

2.2 Pan-STARRS (PS1) Usage

Pan-STARRS (PS1) data is deeper ($r \approx 21.8$) but typically has a sparse cadence that makes it less ideal for *initial* discovery of fast movers compared to ZTF. However, PS1 is the "Gold Standard" for **Precovery** (finding earlier observations of a candidate discovered in ZTF) and **Validation** (confirming a faint candidate).

Data Access via MAST:
The Mikulski Archive for Space Telescopes (MAST) hosts PS1 data.

- **Image Cutout Service:** Unlike ZTF, where we download full CCDs, PS1 images are served as "Skycells" which are large and unwieldy. The efficient approach is to use the **MAST**

Image Cutout Service via astroquery.mast or panstamps.[7]
- **Warps vs. Stacks:**
  - **Warps:** These are single-epoch exposures. **Crucial:** You must use Warps for moving object detection.
  - **Stacks:** These are co-adds of multiple epochs. Moving objects are smeared out or rejected in Stacks. Stacks are useful only for identifying static background sources.[9]

2.3 NOIRLab (DECam)

For deep searches, the Dark Energy Camera (DECam) archive at NOIRLab provides a massive field of view.

- **Use Case:** Deep recovery of TNOs or faint MBAs.
- **API:** astroquery.noirlab allows searching the Astro Data Archive.[11]
- **Strategy:** DECam data is often taken in "blocks." Mining DECam requires identifying survey blocks (e.g., from the Dark Energy Survey) where the cadence was designed for TNOs or Supernovae, as these naturally support asteroid discovery.

2.4 Data Standardization and Storage

The Ingestion Engine must normalize the incoming data streams into a consistent internal format.

- **FITS Decompression:** ZTF data is often Rice-compressed (.fits.fz). The pipeline should use astropy.io.fits to handle this transparently.
- **Header Normalization:**
  - **Time:** Convert all timestamps to **Julian Date (UTC)**.
  - **Observatory Location:** Extract the observatory location (Lat/Lon/Alt) from headers (e.g., TELESCOP keyword) and map it to the MPC Observatory Code (I41 for ZTF, F51 for PS1).
- **Storage Hierarchy:**
  - **Hot Tier (NVMe):** Images currently in the "Blinking" queue.
  - **Cold Tier (HDD):** Raw FITS files for archival purposes.
  - **Database (SQL):** A metadata store (SQLite/PostgreSQL) tracking which Field/CCD/Quad/Night tuples have been processed. This prevents re-downloading and re-processing data, respecting API rate limits and bandwidth.

Table 1: Survey Parameter Comparison for Archival Mining

Feature	Zwicky Transient Facility (ZTF)	Pan-STARRS (PS1)	DECam (NOIRLab)
Aperture	1.2 meters	1.8 meters	4.0 meters

Field of View	47 sq. deg.	7 sq. deg.	3 sq. deg.
Pixel Scale	1.01 arcsec/pix	0.25 arcsec/pix	0.26 arcsec/pix
Limiting Mag	$R \approx 20.5$	$R \approx 21.8$	$r \approx 23.5$
Primary Use	**Discovery (High Cadence)**	**Validation / Precovery**	**Deep Search (TNOs)**
Data Unit	CCD-Quadrant (Full Frame)	Skycell (Cutouts preferred)	Full Mosaic / CCD
Archive	IRSA (Caltech)	MAST (STScI)	NOIRLab Astro Data Archive

---

3. The 'Blinking' Pipeline: Computer Vision & Difference Imaging

The core of the discovery engine is the "Blinking" pipeline. In the 20th century, this was done by a human toggling between two glass plates. In the Virtual Observatory, this is a sequence of algorithmic transformations: Registration, Subtraction, and Extraction. This process effectively removes the static universe, leaving only the dynamic universe (asteroids, comets, variables, and artifacts).

3.1 Pre-Processing and Background Estimation

Before images can be compared, they must be normalized. The raw science image (I_{sci}) contains a sky background that varies across the field (gradients from the moon, city light pollution, or galactic cirrus).

Background Estimation with SEP:
The pipeline utilizes SEP (Source Extractor Python), a C-optimized library that implements the algorithms of the classic Source Extractor software.13

1. **Mesh Calculation:** The image is divided into a grid (mesh), typically 64×64 pixels.
2. **Local Background:** The estimator calculates the median and RMS of pixels in each mesh box, clipping outliers (stars).
3. Subtraction: This background map is subtracted from the data array:

   $$I_{corr} = I_{raw} - Bkg_{map}$$

This step is critical to ensure that the detection threshold is uniform across the entire CCD.

3.2 Image Registration (Reprojection)

Difference imaging requires that the Science Image (I_{sci}) and Reference Image (I_{ref}) align at the sub-pixel level. Since they were taken at different times with slightly different telescope pointings and rotations, they are never naturally aligned.

The Reprojection Task:
The goal is to warp I_{ref} so that its pixels map 1:1 onto I_{sci}.

- **Library:** reproject (part of the Astropy ecosystem).[14]
- **WCS Extraction:** Extract the World Coordinate System (WCS) headers from both images using astropy.wcs.
- **Algorithm Selection:**
  - **reproject_interp:** Uses bilinear or bicubic interpolation. Fast, but does not conserve flux perfectly and can alias high-frequency noise. Suitable for rapid scanning.
  - **reproject_adaptive:** Implements the DeForest (2004) algorithm. This is an anti-aliased, flux-conserving resampling method.[17] It is computationally more expensive but crucial when pixel scales differ or when high-precision photometry is required. For a discovery pipeline, reproject_interp is often a sufficient starting point, upgrading to adaptive for validation.

3.3 Optimal Image Subtraction (OIS)

Once aligned, we subtract the reference from the science image.

$$I_{diff} = I_{sci} - (I_{ref} \otimes K)$$

Simple subtraction ($I_{sci} - I_{ref}$) fails because the "Seeing" (Atmospheric Point Spread Function) changes between nights. If I_{sci} is blurry (bad seeing) and I_{ref} is sharp, simple subtraction leaves "donuts" (residuals) around every star.
The Alard-Lupton Algorithm:
The solution is Optimal Image Subtraction (OIS). This algorithm solves for a convolution kernel K that minimizes the residuals of static sources.

- **Library:** The ois Python package.[18]
- **Method:** ois.optimal_system(science, reference).
- **Kernel Basis:** The kernel K is modeled as a sum of Gaussians multiplied by polynomials. The algorithm performs a least-squares fit to determine the coefficients that best match the PSF of the reference to the science image.
- **ZOGY Integration:** Advanced implementations may use the **ZOGY** (Zackay, Ofek, and Gal-Yam) algorithm, which derives a "proper" difference image statistic (S_{corr})

derived from the PSFs and noise covariance of both images. While theoretically superior, standard OIS is robust and widely implemented.

Output: The result is a **Difference Image** (I_{diff}) where static stars are suppressed (values near zero) and new objects appear as positive PSFs.

3.4 Source Extraction and Artifact Rejection

The final step in the vision pipeline is to extract sources from I_{diff}.

- **Extraction:** Use sep.extract on I_{diff} with a threshold (e.g., 3σ or 5σ).[13]
- **Morphological Analysis:** Calculate parameters for every detection: flux, semi-major/minor axis, ellipticity ($1 - b/a$), and FWHM.

Artifact Rejection (The "Gotchas" of Vision):
This is the most critical engineering challenge. A naive subtraction yields thousands of false positives per image.

1. **Cosmic Rays (CRs):** High-energy particles impact the CCD, creating sharp, bright spots.
   - **Solution: L.A.Cosmic** (Laplacian Cosmic Ray Identification).[20] This algorithm identifies CRs by the sharpness of their edges (Laplacian) relative to the PSF. Run lacosmic on I_{sci} before subtraction or check if detections in I_{diff} have sharpness scores exceeding physical limits.
2. **Optical Ghosts (ZTF Specific):** Bright stars cause internal reflections. ZTF suffers from "counter-moving ghosts" caused by charge spillage during readout.[23] These artifacts appear to move in the opposite direction of the readout.
   - **Solution:** Use the **ZTF Mask Image** (mskimg.fits). Every pixel has a bitmask value. The pipeline must check if the coordinates of a candidate fall on a pixel with specific bits set (e.g., GHOST, HALO, SATURATED).[4]
3. **Registration Residuals (Dipoles):** Bad subtraction leaves "dipoles" (adjacent positive/negative lobes).
   - **Solution: RealBogus Classifiers**. Train a Convolutional Neural Network (CNN) or Random Forest on cutouts of detections. Real sources look like Gaussian PSFs; artifacts look like dipoles or streaks. For a simpler implementation, filter by **Ellipticity** (asteroids are round or slightly trailed; artifacts are often highly elliptical) and **Negative/Positive Ratio** (real sources have no immediate negative neighbor).

---

4. Tracklet Formation and Orbit Determination

Having reduced the images to a list of candidate coordinates, the pipeline enters the domain of astrodynamics. The goal is to solve the combinatorial problem of linking points into physical trajectories.

4.1 The Linkage Logic (Tracklet Generation)

A single detection is meaningless. Discovery requires a **Tracklet**: a set of ≥ 3 detections consistent with linear motion on the sky.

Combinatorial Search:
Given three catalogs of detections D_1, D_2, D_3 at times t_1, t_2, t_3:

1. **Select Seed:** Pick a detection $d_i \in D_1$.
2. **Velocity Window:** Define a maximum possible velocity (e.g., $v_{max} = 1.0$ deg/day for Main Belt, 5.0 deg/day for NEOs). This defines a search radius r in D_2 and D_3.
3. **KD-Tree Search:** Use scipy.spatial.cKDTree to efficiently query nearest neighbors in D_2 within the cone defined by $v_{max} \times (t_2 - t_1)$.
4. Vector Projection: For a pair (d_i, d_j), calculate the velocity vector $\vec{v}$. Project this vector to time t_3.

   $$Pos_{pred} = Pos_{d_j} + \vec{v} \times (t_3 - t_2)$$
5. **Confirmation:** Search D_3 for a detection near Pos_{pred}. If found, form a candidate tracklet (d_i, d_j, d_k).
6. **Linear Fit Test:** Perform a least-squares linear fit (Great Circle fit) to the RA/Dec vs Time. If the residuals are within the astrometric error (e.g., $< 1.0''$), the tracklet is valid.

4.2 Validation: Filtering Known Objects

Before claiming a discovery, one must ensure the object is not already known. The number of known asteroids exceeds 1.3 million.

- **Tool: IMCCE SkyBoT** (Sky Body Tracker).[24]
- **Method:** Use astroquery.imcce.skybot to query the specific field of view and epoch.
- **Matching:** If a known asteroid is returned within a small radius (e.g., $5''-10''$) of the tracklet, associate the tracklet with that object.
- **Discovery:** If the tracklet passes the SkyBoT check (no known object) and passes the artifact filters, it is a **Discovery Candidate**.

4.3 Orbit Determination (OD)

A tracklet represents a "Short Arc" (typically < 2 hours). While insufficient for a secure orbit, an Initial Orbit Determination (IOD) provides a preliminary solution to assess viability (e.g., is it Earth-impacting? Is it bound to the Sun?).

- **Software: Find_Orb** by Bill Gray.[26] This is the industry standard for arc fitting.
- **Automation:** Use the adam-fo Python wrapper or shell scripting to invoke the console version of Find_Orb (find_o).[28]
- **Algorithm:**
  - **Väisälä Method:** Best for very short arcs. It assumes the object is at perihelion or

circular to constrain the solution space.
  - **Herget Method:** Used as the arc extends.
- **Output:** Keplerian Elements ($a, e, i, \Omega, \omega, M$) and RMS residuals.
- **Filter:** Reject orbits with unphysically high RMS ($> 2.0''$) or impossible elements (e.g., hyperbolic $e > 1$ without cometary appearance).

4.4 The "Digesting" Process (NEO Scoring)

To prioritize candidates, use **digest2**.

- **Function:** This software ingests the astrometry and calculates the statistical probability that the object is an NEO, MBA, TNO, etc., based on its motion vector and population models.[29]
- **Score:** The output is a "digest score" (0-100). A score > 65 suggests a high probability of being an NEO. These objects should be prioritized for immediate submission to the MPC's **NEO Confirmation Page (NEOCP)**.

4.5 The "Two Night" Rule (Precovery)

The MPC generally requires observations on **two distinct nights** to designate a new object.[30] A single tracklet is a "One Night Stand" (ONS).

- **Engineering Solution:** The pipeline must maintain a "Tracklet Database."
- **Linking:** When processing Night N, the Linkage Engine queries the database for orphans from Nights $N-1, N-2,... N-7$.
- **Precovery:** If a tracklet on Night N yields a preliminary orbit, the pipeline can propagate this orbit backward to Night $N-1$, calculate the position, and perform a targeted "force photometry" search at that location in the archival images. If a faint blip is found, the arc is extended, and the discovery is secured.

---

5. Submission: The ADES XML Paradigm

The final stage is the formatting of data for the scientific community. The MPC has deprecated the legacy 80-column ASCII format in favor of the **Astrometry Data Exchange Standard (ADES)**, an XML-based schema. This shift is critical for the modern engineer as it allows (and requires) the reporting of astrometric uncertainties.[31]

5.1 The ADES Schema Structure

An ADES submission is an XML document containing a hierarchical structure.

5.1.1 Observation Context (<obsContext>)

This section defines *who* observed and *how*.

- **<observatory>:** This is the **Observatory Code**.
  - *Critical Rule:* You must use the code of the telescope that took the images (e.g., **I41** for ZTF, **F51** for Pan-STARRS, **W84** for DECam). You **cannot** use a personal code for archival data.
  - *Attribution:* You assert your role via the <submitter> and <measurers> tags.
- **<telescope>:** Description of the optics (e.g., "Palomar 48-inch Oschin Schmidt").
- **<fundingSource>:** (Optional) Grant numbers if applicable.

5.1.2 Observation Data (<obsData>)

This section lists the individual detections.

- **<optical>:** The tag for optical astrometry.
- **<trkSub>:** Your internal tracking ID (e.g., USER0001). This links observations of the same object within the batch.
- **<obsID>:** A globally unique ID for the specific data point (e.g., I41_20231201_0001).
- **<date>:** The observation mid-point in UTC, formatted to high precision (e.g., 2024-01-01T05:30:15.123Z).
- **<band>:** The filter used (e.g., r, g, i).
- **<stn>:** The station code (repeats context).
- **<mode>:** CCD.
- **<astCat>:** The star catalog used for astrometric reduction. This is mandatory for modern astrometry. Use Gaia3 (Gaia DR3) or PS1.[33] This allows the MPC to apply debiasing corrections to your positions.
- **<ra> / <dec>:** J2000 Right Ascension and Declination in degrees.
- **<rmsRA> / <rmsDec>: The Game Changer.** ADES requires the uncertainty of the position.
  - *Calculation:* $\sigma \approx 0.6 \times \frac{FWHM}{SNR}$.
  - *Value:* For a typical ZTF detection ($SNR \sim 10$, $FWHM \sim 2''$), $\sigma \approx 0.12''$. Reporting this accurately allows the orbit determination software to weight your observations appropriately.

Table 2: Mandatory ADES XML Tags for Optical Discovery [30]

Tag	**Description**	**Example / Note**
<obsID>	Unique Observation ID	ZTF_ARCHIVE_User01_0001
<trkSub>	Tracklet ID	TempObj_A (Links points of same object)

<date>	Mid-exposure Time	2024-05-20T12:00:00.00Z (UTC)
<ra>, <dec>	Coordinates	Decimal Degrees (J2000)
<rmsRA>, <rmsDec>	Astrometric Uncertainty	Arcseconds (e.g., 0.15)
<mag>	Magnitude	Apparent Mag (e.g., 21.2)
<band>	Filter Band	r, g, i
<stn>	Observatory Code	I41 (ZTF), F51 (Pan-STARRS)
<astCat>	Reference Catalog	Gaia3, PS1
<mode>	Instrumentation Mode	CCD

5.2 Submission Mechanism

1. **Generation:** Use Python's lxml library to construct the XML tree programmatically.
2. **Validation:** Before sending, the pipeline must validate the XML against the official **MPC ADES XSD Schema** (submit.xsd). This prevents rejection due to syntax errors.[31]
3. **Transport:** Submissions are made via HTTPS POST using cURL.
   - **URL:** https://minorplanetcenter.net/submit_xml
   - **Parameters:** source (the file), ack (your email), object_type (e.g., "NEO Candidate" or "Incidental Astrometry").
   - **Snippet:**
     Bash
     ```bash
     curl -F "source=@submission.xml" -F "ack=user@email.com" -F "object_type=Discovery" https://minorplanetcenter.net/submit_xml
     ```

---

6. The 'Gotchas': Engineering Challenges and Solutions

This section details specific failure modes identified in archival mining operations, differentiating a theoretical design from a working production system.

6.1 The Timing Precision Trap

The Problem: FITS headers often provide OBSJD or DATE-OBS. In ZTF data, OBSJD refers to the start of the exposure.[1]
The Consequence: A standard asteroid moves $30''$/hour ($0.5''$/minute). In a 30-second exposure, it moves $0.25''$. If you report the start time, your position is offset by half the motion ($\approx 0.12''$). For a fast NEO ($1000''$/hour), the error is massive ($\approx 4''$), causing the orbit fit to fail entirely.
The Fix: The pipeline must parse EXPTIME (Exposure Time) from the header and calculate the mid-point:

$$t_{mid} = t_{start} + \frac{EXPTIME}{2}$$

Always report t_{mid} in the ADES <date> tag.

6.2 ZTF Counter-Moving Ghosts

The Problem: Bright stars in the ZTF field of view create internal reflections. Due to the readout electronics, "charge spillage" creates ghost images that appear to move in the direction opposite to the readout scan.[23]
The Mimicry: These ghosts are often faint, fuzzy, and move in a straight line, perfectly mimicking a Main Belt Asteroid.
The Fix:

1. **Geometric Filter:** The pipeline must calculate the vector from the detection to the nearest bright star. If the detection lies on the readout axis and moves symmetrically opposite to the star, it is a ghost.
2. **Mask Check:** The ZTF mskimg.fits contains specific bit flags for ghosts. The pipeline must inspect the mask at the detection coordinates.

6.3 Reference Frame Confusion

The Problem: Astrometric solvers (like scamp or astrometry.net) might output coordinates in the Topocentric frame (apparent place) or J2000.
The Fix: ADES strictly requires J2000 (ICRF) coordinates. Ensure the solver configuration is explicitly set to J2000 output. Mixing frames introduces systematic errors of several arcseconds.

6.4 The "One Night Stand" (ONS)

The Problem: Finding a tracklet on a single night is technically a discovery, but the MPC will not designate it without a second night of observations to confirm the orbit.[30] Single-night submissions are often "Vaulted" (stored but not published).
The Fix: The pipeline is not a "fire and forget" system. It must be a Database of Tracklets.

- **Action:** When a tracklet is found on Night N, the system automatically triggers a search in the database for unlinked tracklets on Night $N-1$ through $N-7$ consistent

with the orbit.
- **Submission Strategy:** Do not submit single-night MBAs. Wait until the pipeline links a second night. For NEO candidates (high digest2 score), submit immediately to the NEOCP, as other observatories may follow up.

7. Conclusion

The transition from "looking at the sky" to "querying the sky" represents the future of asteroid discovery. This Technical Execution Plan provides the roadmap for a computer engineer to build a scientifically valid discovery engine using the ZTF and Pan-STARRS archives. By mastering the data structures (FITS/CCD-Quads), implementing robust vision algorithms (OIS/SEP), and adhering to strict astrometric standards (ADES/J2000), the engineer can effectively operate a world-class survey from a laptop. The "telescope" is no longer a mirror on a mountain; it is the API endpoint of the IRSA archive.

Works cited

1. The ZTF Science Data System (ZSDS) Advisories & Cautionary Notes, accessed on December 25, 2025, https://irsa.ipac.caltech.edu/data/ZTF/docs/ztf_extended_cautionary_notes.pdf
2. ztf_release_notes_dr22.pdf - NASA/IPAC Infrared Science Archive - Caltech, accessed on December 25, 2025, https://irsa.ipac.caltech.edu/data/ZTF/docs/releases/dr22/ztf_release_notes_dr22.pdf
3. IRSA - ZTF API - NASA/IPAC Infrared Science Archive, accessed on December 25, 2025, https://irsa.ipac.caltech.edu/docs/program_interface/ztf_api.html
4. ZTF Metadata - NASA/IPAC Infrared Science Archive - Caltech, accessed on December 25, 2025, https://irsa.ipac.caltech.edu/docs/program_interface/ztf_metadata.html
5. ztf_release_notes_dr23.pdf - Zwicky Transient Facility - Caltech, accessed on December 25, 2025, https://irsa.ipac.caltech.edu/data/ZTF/docs/releases/dr23/ztf_release_notes_dr23.pdf
6. IRSA - ZTF Lightcurve Queries, accessed on December 25, 2025, https://irsa.ipac.caltech.edu/docs/program_interface/ztf_lightcurve_api.html
7. panstamps.downloader module - Read the Docs, accessed on December 25, 2025, https://panstamps.readthedocs.io/en/master/_api/panstamps.downloader.html
8. Get Images from the PS1 Image Server — MAST Notebook Repository - GitHub Pages, accessed on December 25, 2025, https://spacetelescope.github.io/mast_notebooks/notebooks/PanSTARRS/PS1_image/PS1_image.html
9. Pan-STARRS1 data archive home page - PS1 Public Archive - MAST - STScI Outerspace, accessed on December 25, 2025, https://outerspace.stsci.edu/spaces/PANSTARRS/pages/298812201/Pan-STARRS1+

data+archive+home+page
10. PAN-STARRS - MAST Archive - Space Telescope Science Institute, accessed on December 25, 2025, https://archive.stsci.edu/missions-and-data/pan-starrs
11. NSC (NOIRLab Source Catalog) - Astro Data Lab, accessed on December 25, 2025, https://datalab.noirlab.edu/data/nsc
12. Astroquery — astroquery v0.4.12.dev285, accessed on December 25, 2025, https://astroquery.readthedocs.io/
13. Tutorial — sep 1.4.1 documentation - Read the Docs, accessed on December 25, 2025, https://sep.readthedocs.io/en/stable/tutorial.html
14. astronomical image reprojection in Python - Zenodo, accessed on December 25, 2025, https://zenodo.org/records/1162674
15. Python-based Astronomical image reprojection - maintainers @astrofrog @Cadair @svank - GitHub, accessed on December 25, 2025, https://github.com/astropy/reproject
16. reproject_interp — reproject v0.19.0 - Read the Docs, accessed on December 25, 2025, https://reproject.readthedocs.io/en/stable/api/reproject.reproject_interp.html
17. Regular celestial images and cubes — reproject v0.9.1 - · DOKK, accessed on December 25, 2025, https://dokk.org/documentation/astropy-reproject/v0.9.1/celestial/
18. Optimal Image Subtraction (OIS) — ois 0.1a0 documentation, accessed on December 25, 2025, https://optimal-image-subtraction.readthedocs.io/
19. quatrope/ois: Optimal Image Subtraction - GitHub, accessed on December 25, 2025, https://github.com/quatrope/ois
20. cosmicray_lacosmic — ccdproc v2.5.2.dev63+g18b6edf73, accessed on December 25, 2025, https://ccdproc.readthedocs.io/en/latest/api/ccdproc.cosmicray_lacosmic.html
21. lacosmic - PyPI, accessed on December 25, 2025, https://pypi.org/project/lacosmic/
22. Cosmic-Ray Rejection by Laplacian Edge Detection - Scribd, accessed on December 25, 2025, https://www.scribd.com/document/867737696/Cosmic-Ray-Rejection-by-Laplacian-Edge-Detection
23. ZTF Charge Spillage Ghosts - Caltech, accessed on December 25, 2025, http://nesssi.cacr.caltech.edu/ZTF/Web/Ghosts.html
24. srvwrapper - PyPI, accessed on December 25, 2025, https://pypi.org/project/srvwrapper/
25. SkyBoT | LTE, accessed on December 25, 2025, https://ssp.imcce.fr/webservices/skybot/
26. Find_Orb integration, accessed on December 25, 2025, https://birtwhistle.org.uk/Docs/EphInterpolator/Find_Orbintegration.html
27. Find_Orb Orbit determination software - Project Pluto, accessed on December 25, 2025, https://www.projectpluto.com/find_orb.htm
28. adam-fo · PyPI, accessed on December 25, 2025, https://pypi.org/project/adam-fo/

29. arXiv:2309.16407v1 [astro-ph.EP] 28 Sep 2023, accessed on December 25, 2025, https://arxiv.org/pdf/2309.16407
30. Guide to Minor Body Astrometry - Minor Planet Center, accessed on December 25, 2025, https://www.minorplanetcenter.net/iau/info/Astrometry.html
31. ADES - Minor Planet Center, accessed on December 25, 2025, https://minorplanetcenter.net/iau/info/ADES.html
32. ADES Data Submission - Minor Planet Center, accessed on December 25, 2025, https://www.minorplanetcenter.net/mpcops/documentation/ades/
33. ADES Field Values - IAU Minor Planet Center, accessed on December 25, 2025, https://www.minorplanetcenter.net/mpcops/documentation/valid-ades-values/