Top level Phase B plan for LSST-Lasair

Andy Lawrence[1], Roy Williams[1], Stephen Smartt[2], George Beckett[1], Bob Mann[1], Ken Smith[2]

Abstract
This document provides a broad-brush top-level plan for delivering Lasair, the UK broker for LSST-alerts, planned as part of Phase B of the LSST:UK project. We set out background, goals, timeline, development methodology, and initial design, to help keep detailed planning on track. This is an internal planning document, and combines elements of two different Phase B workpackages.

[1] *Institute for Astronomy, University of Edinburgh, ROE, United Kingdom*
[2] *Astrophysics Research Centre, Queen's University Belfast, United Kingdom*

Contents

1. Introduction

1.1 Context

LSST:UK expects to build an event broker to receive and process the full LSST alert stream, and to provide an interface to help astronomers get science done with the alert stream in quasi-real time, as well as the accumulating database of alerts. The LSST project expects to provide the alert stream to a relatively small number of approved brokers, with these being chosen by a competitive process, which is underway at the time of writing this document (October 2019).

During Phase A of the LSST:UK project we made an excellent start by building a Lasair prototype that received and processed alerts from the Zwicky Transient Facility (ZTF). Phase B of LSST:UK is now funded and underway. A key part of the approved Phase B workplan is developing and implementing the full Lasair event broker.

1.2 Motivation

The aim of this document is to provide a top level plan for developing, testing, and implementing Lasair. The LSST:UK Phase B proposal sets out the broad goals, and in the Phase A project we developed a successful methodology of week by week incremental progress. Our intention with this document is to provide a framework that will keep the week-by-week progress on track, and set out clearly the goals, milestones and deliverables in a little more detail than given in the Phase B proposal. This document also serves to pull together in one place a short version of material relevant to Lasair that is spread across several other documents and wiki pages.

This document is not intended to be a detailed design or specfication document; these will emerge elsewhere. It is not intended for external consumption; it is an internal planning document.

1.3 Related documents

This planning document should be seen in the context of a number of key documents in other areas.

Phase A Technical Reports:

- LUSC-A-02 Overview of LSST approach to alerts and variable objects
- LUSC-A-03 Ingestion of transient object detections from PanSTARRS1 and ATLAS
- LUSC-A-08 The Lasair-ZTF Transient Broker
- LUSC-A-09 LSST:UK DAC: an initial analysis of requirements from Transient and Variable Science

Key related LSST project documents:

- LSST Overview, Ivezic et al 2019 Ap.J. 873, 111.
- Observing Strategy white paper, arXiv:1708.04058
- DMTN-102 LSST Alerts- Key numbers

- DMTN-118 Review of Timeseries Features
- LDM-151 Data Management Science Pipelines Design
- LDM-542 Science Platform Design
- LDM-554 Data Management LSST Science Platform Design Requirements
- LSE-612 Plans and Policy for LSST Alert Distribution
- LPM-17 LSST System Science Requirements Document
- LSE-163 Data Products Definition Document
- LSE-231 LSST Data Products categories

LSST:UK Phase B documents and related web pages:

- The Lasair-ZTF public prototype
- Lasair development wiki page
- The LSST:UK Science Requirements Document
- Phase B proposal (available at LUSC overview page)
- Phase B Project Management Plan
- Phase B deliverables and milestones (available at LUSC overview page)

2. The Lasair-ZTF prototype

Within Phase A of the LSST:UK project, we developed a broker to receive and process alerts from the ZTF project and provide them to external users. This work was carried out both for its intrinsic scientific value and utility for astronomers worldwide, and also as a Phase A prototype for our planned LSST broker. The system we developed was called "Lasair". When we developed the Phase B proposal for LSST:UK, we also called the proposed broker "Lasair". For future development, it is important to distinguish between the prototype and the final LSST product, especially because the prototype will continue to have its own intrinsic value. For this reason, we will refer to the prototype as "Lasair-ZTF".

In this section we summarise the methods, utility, and achievements of Lasair-ZTF.

The Lasair-ZTF prototype ingests the ZTF sky survey in preparation for ingesting the LSST sky survey. ZTF and its infrastructure has been designed for similarity with LSST, so infrastructure built, and lessons learned are an excellent preparation for the full LSST transient stream, that will be 10 to 100 times as voluminous as ZTF. In this prototype, we have a nightly ingestion process that has run for over a year without major failure, and we have released a public website that is attracting attention from the UK and international astronomers. The philosophy of Lasair-ZTF development has been to evolve "from working to working", so that we are starting with a broker for ZTF, developing its functionality on the basis of user experience, with scalability testing in parallel.

Lasair-ZTF is built with a relational database at the centre, with ingestion modules loading data, annotation modules adding that data, web and Jupyter aspects to provide user access, together with a nascent real-time stream aspect.

Spatial filters ask if the event is within a specified geometry (a subset of the celestial sphere), and include 'skymap' queries that come from gravitational-wave alerts, where the geometry is actually a probability density on the sky. They also include the 'watchlist' type of geometry that consists of a set of sources (a catalogue), each with a radius.

Lasair-ZTF also supports annotation of events, for example, by contextual search of massive catalogues, where nearby 2MASS or PanSTARRS sources are found and associated with the event. Lasair-ZTF also computes statistics of the light curve or each object, such as minimum and maximum magnitude, etc. Other annotations include user comments on objects, and crossmatches with the Transient Name Server (TNS).

3. Lasair in the LSST:UK Phase B plan

Phase B of the LSST:UK project began in July 2019 (Project month 1, or PM1), with a planned conclusion in March 2023 (PM45). As with Phase A, it is divided into three main workpackages - Management (WP1); DAC (WP2); and DEV (WP3). At the conclusion of the proposal review process, Lasair-related work was funded in two sub-packages:

WP2.3 Alert Handling Infrastructure
Total staff effort: 17 staff months
Lead: Andy Lawrence
Other key staff: Roy Williams, Dave Morris

WP3.2 Lasair: UK Transient Broker
Total staff effort: 36 staff months
Lead: Stephen Smartt
Other key staff: Ken Smith, Dave Young, Roy Williams

Two other related DEV sub-packages were originally proposed but cut in the review process: WP3.3 *Transient Classification and Spectroscopic Followup*, and WP3.4 *UK Variability broker*. These planned sub-packages would have been our key internal "customers". The work concerned may go ahead to some extent via unfunded effort, and through the international Science Collaborations, and we will need to pay attention to the requirements and expectations of the related communities and collaborations; but we won't able to commit dedicated staff effort to support them.

The Phase B programme has four broad **Objectives**, all of which are relevant to Lasair and should drive us throughout. In brief, these are:

- Maintain leadership in the LSST Science collaborations

- Deploy a DAC infrastructure to support exploitation of LSST by UK astronomers
- Participate in the LSST Commissioning Programme
- Develop services to integrate LSST data with other data sources

The Phase B programme has a number of formal **Deliverables**. The ones related to Lasair, with associated delivery dates, are:

- PM9: D3.2.1 SHERLOCK Version 2.0 Event Classifier (Mar 2020)
- PM16: D2.3.1 Lessons learned on setup of ancillary catalogues for cross-matching ZTF alert stream (Oct 2020)
- PM33: D3.2.2 Lasair User Documentation (Mar 2022)
- PM38: D2.3.2 Design of infrastructure for Lasair (Aug 2022)
- PM45: D3.2.3 Design and implementation of web front-end for LASAIR - public and the data-rights version. (Mar 2023)

In addition to the formal deliverables, the Phase B plan sets out **Milestones**. The ones relating to Lasair, together with related WP, are:

- PM9: MS3 (WP 3.2) Lasair 2.0 inc. SHERLOCK
- PM21: MS11 (WP3.2) Lasair 3.0 inc. ”tech preview” of web interface and tools
- PM33: MS17 (WP3.2) Lasair 4.0: final web interface and user docs
- PM38: MS19 (WP2.3) Infastructure for Lasair (receive and process LSST stream)
- PM45: MS23 (WP3.2) Lasair 4.0 for non-proprietary data

4. Lasair goals

The Phase B proposal gives the top-level goal for WP3.2 as *For every alert, provide an answer to the question “what has been detected (resolved in time) at this position in the sky, at every wavelength from x-ray to radio?”*. To achieve this, the proposal listed a number of specific goals. These have now formally evolved into the requirements listed in the LSST:UK Science Requirements Document. We reproduce those here, slightly re-phrased and shortened.

- **R2.01 A searchable database containing all the LSST alerts**
- **R2.02 Light-curves:** assimilate all diaSource alerts in diaObjects: providing interactive webpages (linked to database), plots, ability to select ranges, and to submit user added points.
- **R2.03 Postage stamps:** all LSST detections and most recent non-detections. Plus multi-colour images from LSST and multiple other surveys.
- **R2.04 Massive catalogue cross-match:** with star, galaxy, AGN, x-ray, radio catalogues, using Sherlock, and machine learning.
- **R2.05 Crossmatch to all previously known transients**: e.g. supernovae, gamma ray-bursts, TDEs, x-ray and radio burst sources
- **R2.06 A database query platform and user-owned storage:** Both a SQL query form and a Jupyter platform.
- **R2.07 Real time cross match** to multiple other time domain surveys at various wavelengths.
- **R2.08 Spectroscopic and/or photometric redshifts:** and hence absolute mags, from other catalogues
- **R2.09 Classification from light curves:** Combine all the above and 24hr-48hr lightcurve information to probabilistically classify all transients as : supernova – kilonova – GRB – Tidal Disruption Event – AGN – XRB – CV – eruption star – microlens – orphan
- **R2.10 Multi-messenger cross-matching:** Probabilistic association with GW and neutrino sources, based on public LIGO and Ice Cube information, and potentially other sources.
- **R2.11 Provide a stream of transients to external collaborations:** such as 4MOST and SOXS programmes.
- **R2.12 Provide users with a means to upload a ”Watch-list”:** of up to 10^6 objects, including adjustable search radius, and the means to trigger on magnitude variations.
- **R2.13 Collect a detailed list of additional user requests:** and implement these where possible.

The additional user requests mentioned in requirement R2.13 are being collected from the LSST:UK community, and the international science collaborations, via our LSST:UK membership of those. They are being collected on the wiki here. They are perhaps better described as detailed requests, rather than additional requirements. They will be considered cycle-by-cycle as we go along, and considered within the broader context of making sure that we achieve the main requirements.

The Lasair team itself has also been discussing additional capabilities, which potentially could be added to the main requirements list. These will be considered by the LSST:UK Project Scientist. Items currently being considered are:

- **User filtering of near-real-time alerts:** including complex filtering and selection, and the ability to receive “push” alerts
- **User-defined stored queries:** so that these can be quickly activated when required.
- **VO integration:** for example providing a TAP service interface so that other tools can interrogate the alerts database

The Science Requirements Document does not list specific requirements for Work Package 2.3, and the Phase B proposal did not set out listed goals. However, from the Phase B proposal text, we suggest that the following should be seen as WP2.3 requirements:

- Provide the hardware and software infrastructure necessary to receive and process the LSST alert stream
- Operate the process of receiving and processing the LSST alert stream
- Ingest and curate the LSST databases necessary to enable WP3.2 to achieve its goals
- Build, ingest and curate the additional external databases necessary to enable WP3.2 to achieve its goals
- Provide the platform within which WP3.2 and possibly other DEV teams can filter streams as required
- Provide the ability to execute queries on the related databases in real time
- Provide the hardware and software infrastructure necessary for users to run more advanced processing jobs on the related databases (through Jupyter notebook or other systems)

5. Responsibilities in Lasair-related work

In recent strategy meetings, we decided to agree the individual responsibilities for Lasair using a ARIC model, i.e. specifying who is *Accountable*, i.e. to the Consortium, and to STFC, for seeing that the deliverables are achieved; who is *Responsible*, i.e who will in practice lead the work; who is *Involved*, i.e. other individuals carrying out segments of the work; and who needs to be *Consulted*, i.e. kept informed of progress, and/or consulted on preferences, requirements, interfaces etc. We decided these roles in a few key areas:

Requirements Definition:
A: Stephen Smartt
R: Stephen Smartt
C: Int. Sci.collabs (via Andy Lawrence and Stephen Smartt); LSST:UK consortium (via Project Scientist)
I: LSST Project

Design(LSST Platform):
A: Andy Lawrence
R: Roy Williams, Gareth F
C: Dave Morris
I:

Design(Classifier):
A: Stephen Smartt
R: Ken Smith
C: Andy Lawrence
I:

Infrastructure (LSST Platform):
A: Bob Mann
R: George Beckett
C: IRIS
I:

Lasair-ZTF:
A: Andy Lawrence
R: Roy Williams
C: ZTF
I: ZTF

The Accountable individual will work with the LSST:UK project manager to see that the work is tracked, and necessary reports are provided, but of course working closely with the other team members.

6. Development methodology

We are trying to keep a balance between long term planning and consistency and short term flexibility and agility. We can think of our development process as divided into timescales.

Four Year Plan. Over the whole lifetime of the project, we will be guided by this document, which in turn is in part an extract and summary of the overall LSST:UK Phase B plan. We will keep in mind the overall goals, deliverables and milestones, which will be considered fixed unless there is a good reason to evolve them. This document also sets out initial design ideas and broad technology issues; but these may be expected to evolve.

Six Month Planning Cycles. Every six months we will take stock of progress, and take note of any deliverables and milestones due in the coming six months. We will then debate and agree some goals and tasks to guide our work over the coming six months. These tasks will likely include key *component implementations*. We will also debate and decide any key *decision points* required in this six month window - for example on basic technology choices which would be hard to row back from later. We will often need to plan *experiments or tests* to help us make those key decisions. Finally, we will make sure that any tasks and goals planned are attached to specific people, and achievable given their available effort during the cycle.

The dates of these intended planning cycles are included in the schedule section below, and are phased to match key Phase B dates.

Fortnightly Progress Meetings. Finally, we will continue the method we used for developing the Phase A prototype, of having regular but short progress meetings. These meetings keep momentum, allow us to react flexibly to developments, help to share and solve problems that come up, and make sure that we do not drift during each planning cycle. A new feature that we intend is to have standing agenda items. This will help us to keep focus and track progress.

7. Relation between Lasair-LSST and Lasair-ZTF

A core question is the relation between the existing Lasair-ZTF prototype, and the final Lasair intended for LSST. Two extremes (neither of which are ideal in their pure form) would be as follows:

(A) **Evolve prototype**. One possibility would be to continue to develop Lasair-ZTF iteratively, evolving it into the final Lasair.

(B) **Start again**. Another possibility would be to shut down the prototype, report on the lessons learned, and start again with a consistently engineered new version of Lasair.

Option (B) is not a good idea simply because of the scientific usefulness of Lasair-ZTF. We should commit to running an operational Lasair-ZTF without interruption. Furthermore, the feedback and suggestions we get from real live users has been very important, and will continue to be so. Last but not least, we need visible proof of our operational capability during the LSST Community Broker review process.

Option (A) in its simplest form is not ideal either. It wastes the opportunity to re-think the optimum design solution, based on the lessons learned, and to take advantage of new technologies. There may be scalability issues or other problems that become apparent with time. We may find that to meet LSST demands, an evolutionary approach runs into a dead-end, and a complete overhaul is necessary anyway. We should not risk this happening while trying to run simultaneous ZTF and LSST services.

Our aim should be get the best of options (A) and (B), by running parallel streams, with staged decision points about implementation, switch-over as necessary, etc. A key goal in Cycle-1 should be to produce a plan of how to do this, but as we write this document in Oct 2019, we might envisage components of this plan as follows:

Operate Lasair-ZTF. We should support and maintain a public working version of Lasair for ZTF data.

Lasair-ZTF developments. For the time being, we should continue to add user utility improvements to Lasair-ZTF, but avoid making major changes to the technical implementation. As now, this would involving developing and testing on a Lasair-dev system before release to the operational version, but this dev system is *not* the same as the LSST experimental system.

Lasair-LSST design. We should be producing a technology-free top-level design for the final version of Lasair. Our preliminary attempt at this is discussed in section 9, and is probably good enough to proceed.

Technology research. We should be researching the technology options for Lasair-LSST, and reporting findings to the group.

Lasair-LSST experiments. To inform our final decisions, we should undertake experiments that test technology options and architectural variations, using simulated LSST data. In the short term then we might be running three systems - operational Lasair-ZTF, Lasair-ZTF-dev, and Lasair-LSST-test - but this is necessary to save work in the future.

Design Review and Implementation Plan. With research and experiments in hand, we should undertake a review that produces a plan for moving ahead. We should be open to either a complete design/rebuild or the possibility that Lasair-LSST will be more or less a clone of Lasair-ZTF. We need to take decisions in time to have Lasair-LSST ready for commissioning. As well as the design, UI and underlying technology, another issue will be whether Lasair-ZTF and Lasair-LSST look like independent entities, or more like switches or options on a single tool.

8. Key dates and schedule

Several components go into our overall schedule. (i) LSST project key dates, such as the start of commissioning. (ii) Review and decision dates on broker selection. (iii) LSST:UK Phase B deliverable and milestone dates. (iv) Lasair project planning cycle dates. (v) Other internal and external key dates that we decide to pay attention to during our planning process.

Some of these dates are essentially fixed, and others will change as we progress. The project schedule will therefore be a living thing. Below are the key dates as we understand them at the time of writing this document (October 2019). Some of them are rather loose guesses.

```
PM1  Jul 2019 preparatory planning
PM4  Oct 2019 Cycle-1 start
PM9  Mar 2020 submit CB proposal
PM9  Mar 2020 D3.2.1 Sherlock V2.0
PM9  Mar 2020 MS3  Lasair 2.0 release
PM10 Apr 2020 Cycle-2 start
PM10 Apr 2020 CB result
PM16 Oct 2020 D2.3.1 ZTF lessons learned
PM16 Oct 2020 Cycle-3 start
PM16 Oct 2020 commissioning start
PM21 Mar 2021 MS11 Lasair 3.0 release
PM22 Apr 2021 Cycle-4 start
PM24 Jun 2021 access to commissioning alerts
```

```
PM28 Oct 2021 Cycle-5 start
PM30 Dec 2021 Contingency period starts
PM33 Mar 2022 D3.2.2 Lasair user docs
PM33 Mar 2022 MS17 Lasair 4.0 release
PM34 Apr 2022 Cycle-6 start
PM38 Aug 2022 D2.3.2 Lasair inf. design
PM38 Aug 2022 MS19 Lasair infrastructure
PM40 Oct 2022 Cycle-7 start
PM40 Oct 2022 Full LSST operations start
PM45 Mar 2023 D3.2.3 Web front end
PM45 Mar 2023 MS23 Lasair 4.0 non-prop
PM45 Mar 2023 Phase B end
PM46 Apr 2023 Phase C start
PM49 Jul 2023 LSST DR1 release
```

9. Initial conceptual design

The basic philosophy and design concepts for Lasair-LSST will most likely remain as we have developed them for Lasair-ZTF. The working philosophy of Lasair-ZTF has been iterative, developing from "working-to-working". We will keep that successful working practice, but as we re-engineer for LSST, we wish at the same time to take a step back and agree a clearly understood high-level architecture in diagrammatic form. This is not meant to be a detailed design, but rather a broad-brush picture that gives us a clearly understood and relatively stable framework within which detailed design can be developed iteratively. Furthermore, the high-level design should be, as far as possible, technology-free. It is a conceptual design within which technology choices can be made to produce implementations.

As we started to develop our high-level design, we found it desirable to produce both a very high level "components diagram" and a slightly less high level "architecture diagram". Current versions of both of these are appended here.

We expect that the conceptual architecture diagram will evolve and improve. However, we also hope that quite soon it will become relatively stable - the intention is that it provides a firm framework to guide our detailed work.

10. Technology issues

10.1 Lasair-ZTF technology

Lasair-ZTF sends out its alerts using Kafka, as planned also by LSST. We therefore learned how to receive, process, and ingest a Kafka stream, which was a new venture for Edinburgh and Belfast. Thereafter, Lasair-ZTF was built with deliberately conservative technology choices, in order to make a working service as simply and rapidly as possible.

(i) Events are ingested into a relational database, with as little delay as possible. The design assumption is that science is done by users interrogating that database.

(ii) The database system is built using MySQL and pieces of Python glue.

(iii) Contextual information is added to the events using Sherlock, Belfast-built software that has been successfully operating for some time.

(iv) The user interface is built on a website built constructed with Django, with some additional pieces of technology such as plotting with Plotly.

(v) A Jupyter Hub service was set up, for users to run analysis on the database using Python notebooks. However, this has been seen as an experimental no-guarantees facility, made available to selected users on request, rather than being a fully-public service.

10.2 Technology options for Lasair-LSST

Technology questions and implementations are going to evolve significantly in the coming months to years, but these are the key issues as we see them in Oct 2019.

(i) **Volume testing**. We need to find out what breaks as we scale up. This is moderately urgent to understand, as we need soon to fix on a deployment design.

(ii) **Hardware deployment**. The working assumption is migration to the IRIS infrastructure. Work on this has already started, but we need to understand its implications. An advantage of IRIS is its scale: we can utilise large numbers of machines to achieve arbitrarily high speeds.

(iii) **Streams versus databases**. We need to be able to feed filtered streams in quasi-real time to users and machines used by other projects, and to provide access to the historical database of transients. As noted above, we have built the historical query system, and we have now started building quasi real-time streams. Can we maintain both history-query and real-time-stream methodologies?

(iv) **Technology choices**. Should we stay with conservative technology (i.e. MySQL) or should we plan for deployment of one or more new technologies that may have advantages? Examples that are under discussion include Kafka Streams / KSQL, Cassandra, NoSQL databases, and AXS (= SparkSQL + Parquet).

(v) **Using Lasair-LSST**. Astronomers may use Lasair-LSST in several ways ('aspects'). Web pages and forms; Jupyter notebooks; programming an API; consuming a Kafka stream; or installing a version of our software and services. The first three are our priority now. However we note that a Jupyter notebook service will likely be provided by the UK-DAC for more general purposes, and furthermore it is likely that this will be a clone of the service provided by the LSST Project "Science Platform"

Acknowledgments

The authors of this document would like to thank all other members of the group working on building Lasair, as well as the relevant scientists across the LSST:UK consortium.

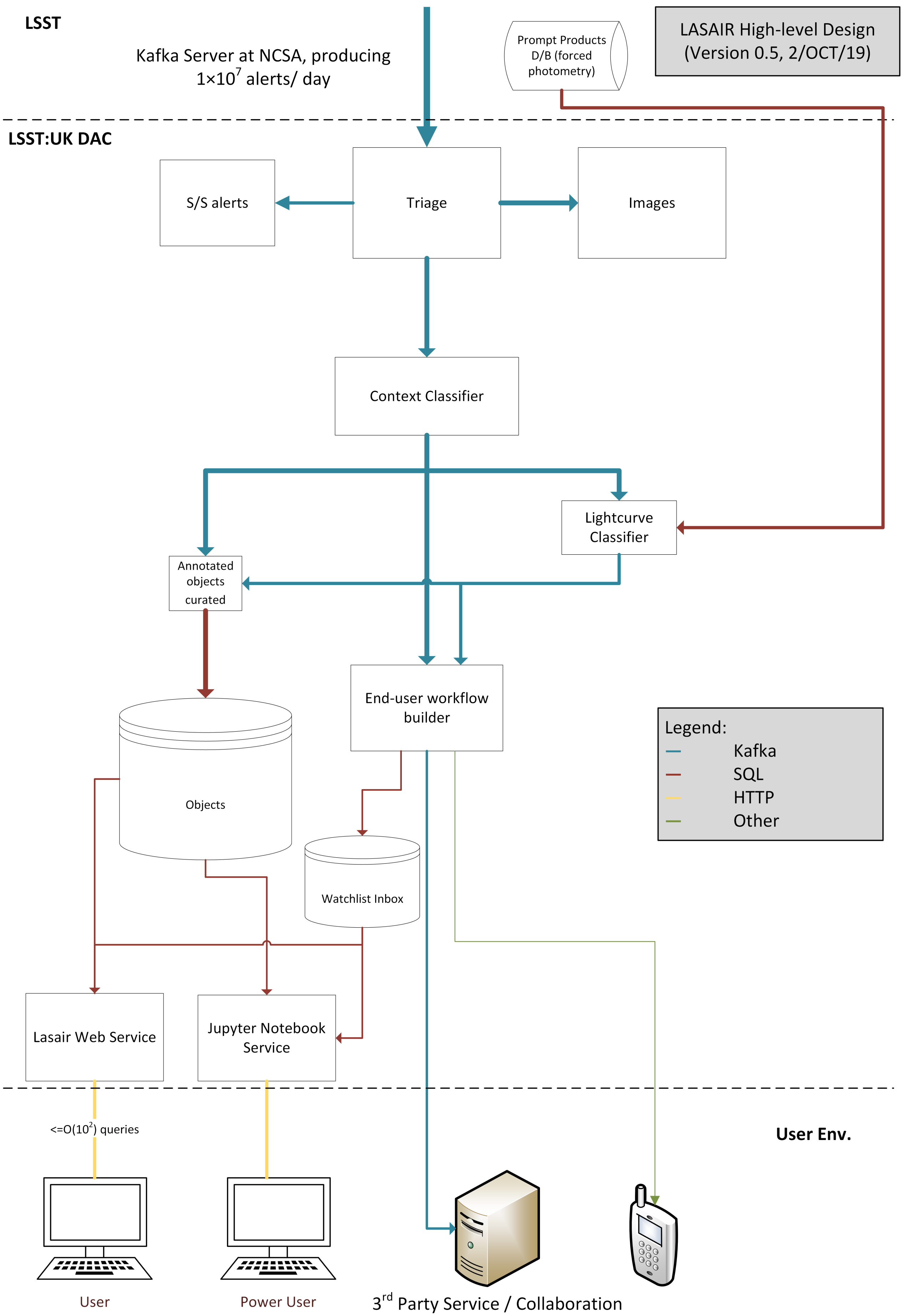

Figure 1. Very high-level components diagram for Lasair-LSST: as of Oct 2019

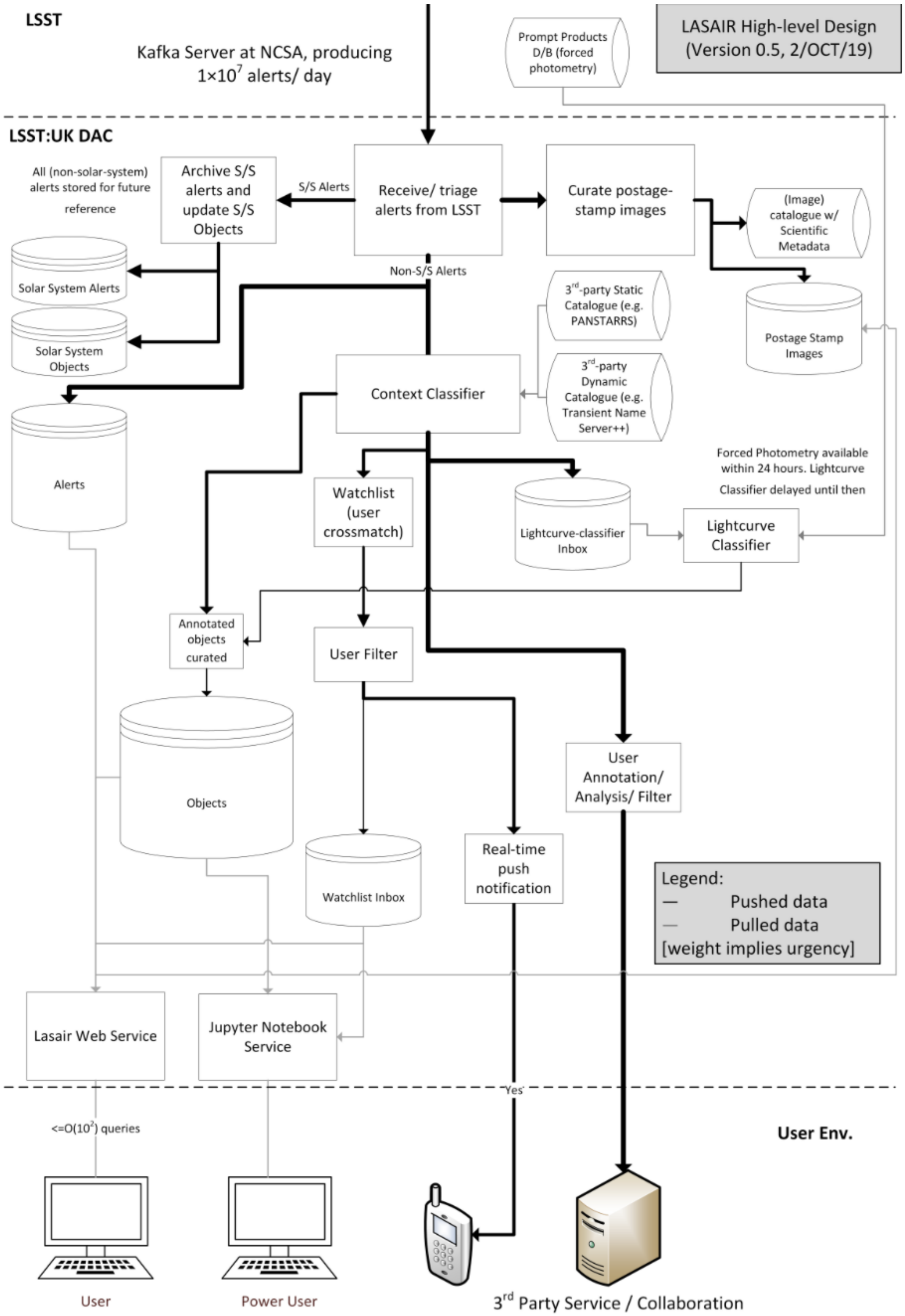

Figure 2. High-level conceptual architecture diagram for Lasair-LSST: V0.5 as of Oct 2nd 2019