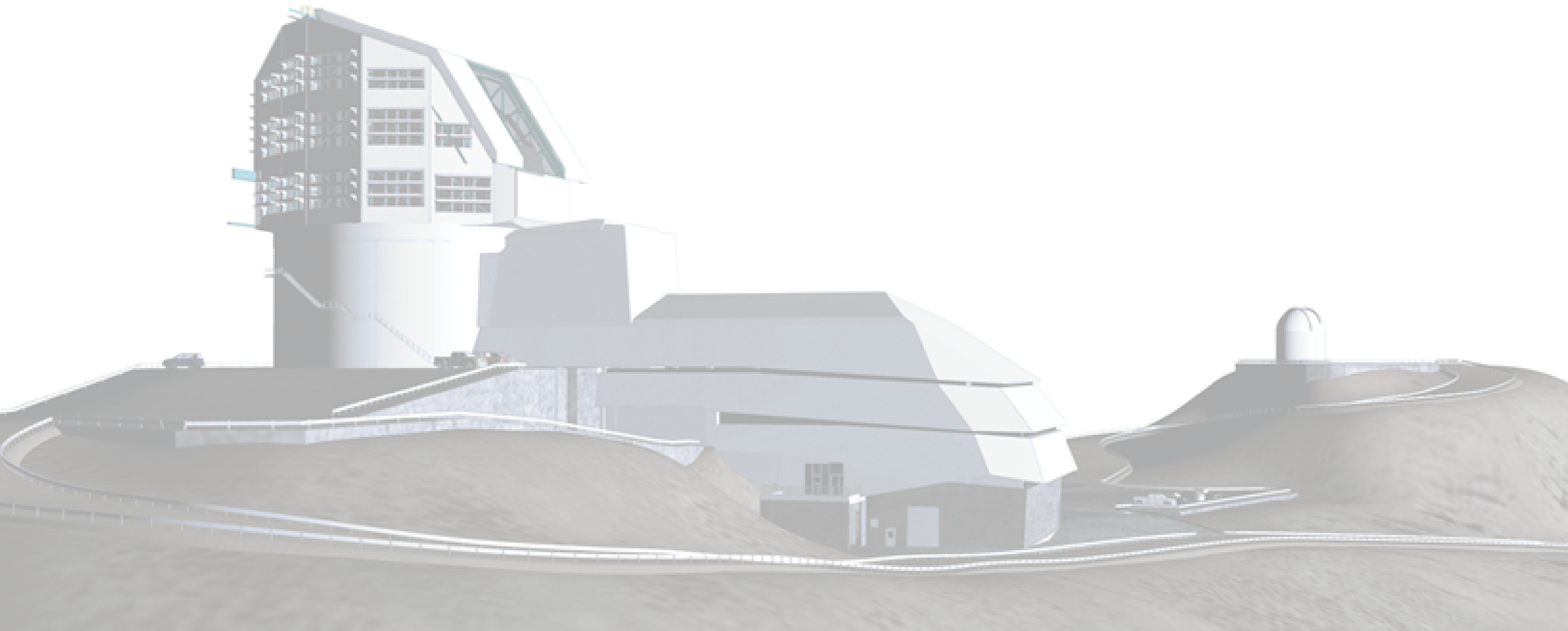
Vera C. Rubin Observatory
Data Management

A Hybrid Notification and Alert Retrieval Service

Eric C. Bellm, and Spencer Nelson

DMTN-165

https://doi.org/10.71929/rubin/2998002
Latest Revision: 2020-12-16



Abstract

We consider the implications of providing a subset of alert packet contents in the alert stream, with the full alert packets available for rate-limited retrieval using an identifier included in the “lightweight” alert. This hybrid approach would enable much wider dissemination of the complete lightweight alert stream, potentially to thousands more users than is possible in the current baseline. In turn this “alerts on your laptop” access would allow users to conduct more sophisticated filtering and analysis than the current Alert Filtering Service, increasing the overall scientific returns of the alert stream. Community brokers would still be able to retrieve all of the contents of the full alert stream within the 60 second latency window and would gain greater control over their ingestion of the bulk alert data. Technically, this hybrid system may represent a modest increase in complexity over the current baseline but is likely to be more operationally robust.

Change Record

Version	Date	Description	Owner name
1	2020-12-16	First release.	Eric Bellm
1.1	2025-10-09	Add DOI	Tim Jenness

Document source location: `https://github.com/lsst-dm/dmtn-165`

Contents

A Hybrid Notification and Alert Retrieval Service

1 Background

As it conducts the Legacy Survey of Space and Time, the Rubin Observatory will produce a near-real-time alert stream to notify astronomers around the world of all the transients, variables, and moving objects that it detects in difference imaging. Rubin Observatory alerts are immediately world-public and are intended to facilitate timely follow-up of time-critical events. Alerts are sent to third-party community brokers for further enhancement and redistribution and are also available to Data Rights holders through the Alert Filtering Service.

In order to provide all of the information needed to make rapid classification and follow-up decisions, the alert packets are "rich"–they contain not only the information about the latest detection, but also past detection history, forced photometry and/or upper limits, the associated `DIAObject` or `SSObject` record, linkages to counterparts in the most recent LSST data release, and image cutouts. The alert contents are public and freely shareable [RDO-013].

Because of the large amount of data in each alert, the alert packets are relatively large. DMTN-102 aggregates relevant sizing information about alerts. Alert sizes of 82 KB have been estimated from simulations, although since the image cutouts are sized with the detection footprint, packet sizes of a few megabytes have been seen in processing of precursor data. To
DMS-REQ-0274
send 10,000 alerts of 82 KB within a nominal 5-second window of the allowed 60-second alert latency, more than 1 Gbps of outbound bandwidth is required. This constraint in turn limits the number of community alert brokers that can be supported; a minimum of five brokers is required. Accordingly Rubin Observatory is conducting a proposal process to select which
DMS-REQ-0391
community brokers will be allowed to directly receive the full alert stream.

DMTN-093 describes the baseline technical design for the Rubin Observatory Alert Distribution System. Alerts are serialized to a strongly-schemaed binary format, Apache Avro, for compactness. The open-source distributed streaming platform Apache Kafka provides the alert distribution interface—selected community brokers will connect to the Rubin Kafka cluster and consume the alert stream via Kafka clients. This approach has been demonstrated at smaller scale by the Zwicky Transient Facility (Patterson et al., 2019).

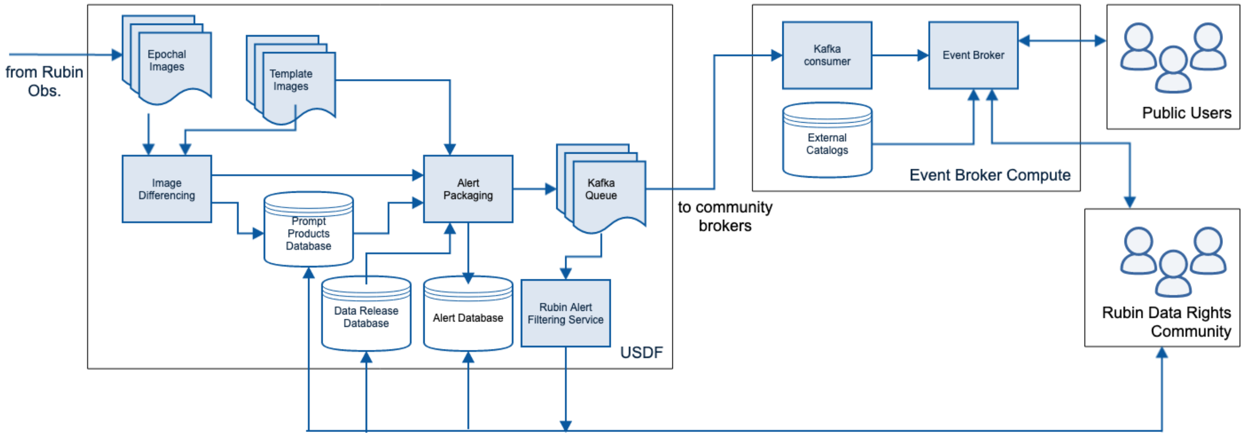


FIGURE 1: Data flow diagram for the baselined Alert Distribution System.

2 Motivation

The large size of the alert packet in the baseline design creates several problems. Most importantly, it makes outbound bandwidth from the Data Facility the primary constraint on the system. As a result, only a small number of consumers can receive the full alert stream. This limits the reach and impact of the alert stream and hinders the flexibility and creativity of scientists. Moreover, both brokers and users must receive and seek through the entirety of each large packet on their system to find the next packet, even if they intend to filter it out and discard it—this makes handling the alert stream more technically challenging than is necessary. Because of the requirement to send alerts to brokers within 60 seconds of readout, outbound bandwidth usage is extremely "peaky"–most of the traffic is sent within ~5 seconds of every 39 seconds. Finally, Kafka, the baselined technology for alert distribution, is optimized for small messages (~1 kB[1]), and performance can degrade as message size increases. The large range of alert packet size due to variable-size cutouts poses a particular challenge in this regard as we must configure a maximum message size for Kafka. While these implementation details are specific to Kafka, they reflect a current architectural design consensus within the software community that notification should be separated from data delivery. It is difficult to build a system that simultaneously provides rapid, reliable notifications as well as efficient delivery of bulk data!

Our overarching goal for considering a hybrid alert architecture is to enhance the scientific return of the LSST alert stream by ensuring a large and robust broker and user ecosystem

[1] https://docs.cloudera.com/documentation/kafka/latest/topics/kafka_performance.html

and by improving performance and access for individual users relative to the baseline. At the 2019 community broker workshop, there was clear consensus among the attendees that the Project should explore avenues that would broaden access to the alert stream. At the same time we want to maintain the rich content of the alert packet to enable rapid filtering, sophisticated scientific analysis, and straightforward sharing of the public alerts.

3 The Hybrid Alert Packet Concept

Before describing the hybrid alert concept, we first stress that we are not advocating for a change to the relevant requirements. We will still produce all of the required alert contents and be capable of transmitting them to the required number of brokers within the required 60 second window after readout completes.

LSR-REQ-0101
OTT1
DMS-REQ-0391
numStreams
DMS-REQ-0274

In the hybrid alert packet concept, the realtime alert stream still contains information about all detected `DIASources`. However, the transmitted packets contain much less information than the full alert packets described in the DPDD [LSE-163]. This lightweight "notification stream" would contain a bare minimum of science data to allow users to determine a subset of alerts that are of potential interest and retrieve the full packets only for those alerts. We expect that many science use cases will only require retrieval of a few percent or less of the corresponding full alerts.

Each lightweight alert packet would include a link to allow the user to retrieve the full alert packet containing all of the DPDD-specified quantities (image cutouts, `DIASource` history, forced photometry where available, timeseries features, etc.). Rate limits assigned to each user would manage bandwidth use. The typical workflow will be for users to inspect the contents of the lightweight alert, filter them down to a relevant subset (e.g., alerts coincident with the LMC; alerts in deep drilling fields; fast-evolving transients; NEO-like alerts), and then request and retrieve the complete alert packets for that subset to further analyze.

Appendix A provides a draft set of potential lightweight alert contents. They are not meant to completely characterize any event, but merely to provide maximum scientific distinguishing power, to allow users to determine if this alert is likely a transient, variable, or moving object. It is expected that users will still need to retrieve the full alert packet to be able to achieve their science goals.

The size of the lightweight alert is ∼200 bytes/packet. Averaged over 39 seconds per visit this implies an average bandwidth required of 400 kbps; to transmit all of the lightweight alerts in a visit within 5 seconds requires 3.2 Mbps. This implies that we can easily serve thousands of users the lightweight alert stream using a 10 Gbps network interface from the Data Facility. A night of observing would produce a total volume of lightweight alerts of 2 GB. In contrast to the full alert stream, which is ∼ 820 GB per night, it is easy to imagine a user accessing the full alert stream from their laptop, or perhaps connecting a small or ad-hoc automated telescope to it. We believe that democratizing access to the alert stream will enable some of the most creative uses of the stream and ensure innovative use by scientists.

Users can fetch the full alert packet, but these requests would be rate limited to keep them from unfairly consuming all available bandwidth. Choosing an appropriate value for the rate limit is a policy decision which must balance the science benefits of letting users access full packets against providing access to more users.

For example, the rate limit could be set to permit users to get all data as long as they spread their requests out. To keep up with an event generation rate of 10,000 events per 39 second exposure, users would need to retrieve 270 events per second; at 80 KB per event, this would require 164 Mbps of bandwidth per user. At this volume, we could support 60 users within the 10 Gbps bandwidth budget. Tighter rate limits would permit more users, but those users would have to choose to receive only a subset of the available full payloads.

The discussion above focused on science users. Most (though perhaps not all) community alert brokers will still wish to retrieve all available alerts. In this design, brokers can simply be viewed as users with very high rate-limits, sufficient to allow transmitting all alerts out of the Data Facility within the 60-second latency window.

4 Technical Implementation

We consider a technical implementation derived from the current baseline as described in DMTN-093. As the Alert Production pipeline runs, it would produce both the full-sized alert packets and their lightweight counterparts, both in Avro format. The lightweight alerts would be distributed to community brokers and science users using Kafka. The full-sized alerts would be stored on disk in an object store or other system accessible by HTTPS. Brokers and users would obtain an identifier from lightweight alerts and request the corresponding full-

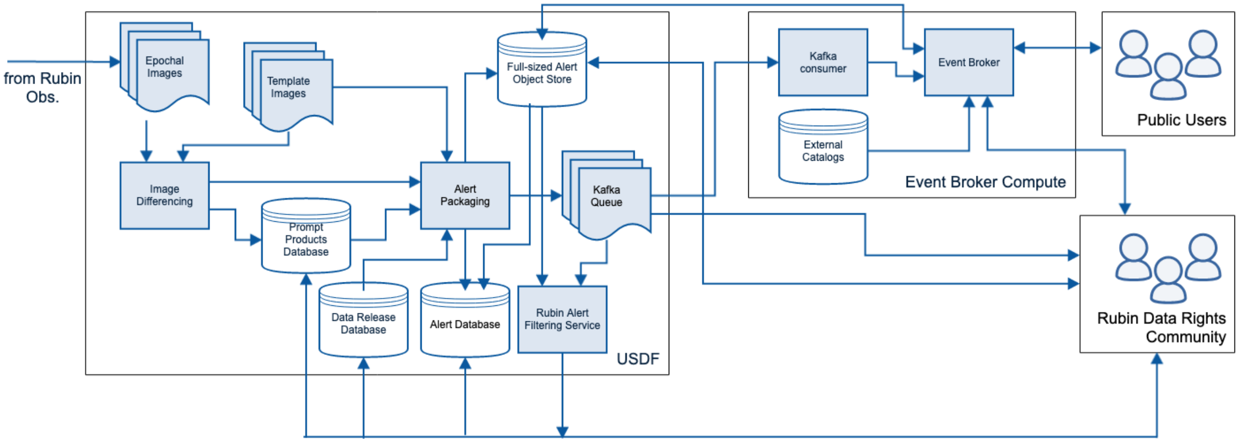


FIGURE 2: Data flow diagram for the proposed lightweight alert system.

sized alerts via HTTPS.

Rate limits would be enforced by examining an HTTP header in each request which identifies the user. These identifiers would be provided through a registration system, integrated with the project's identity and access management systems. When a request for a full-sized alert is received, the HTTP server which provides access to the alerts inspects the header and checks that it has not exceeded its limit. This implementation is commonly provided out of the box by open source HTTP servers[2]. In general, we do not expect rate limits to impose any significant latency increase for retrieving data, since enforcement happens in memory on the HTTP server. Enforcement will take less than 1 msec in most cases, and certainly less than 100 msec in all cases.

The link to retrieve the full-sized alert must be permanent, even if the full-sized alert is moved. This suggests use of an identifier such as a DOI, URI, or IVORN[3]. Indirection, particularly through a third party, may add substantial latency for both publication/availability as well as each retrieval, so the need for permanence must be balanced against performance.

[2] For example, it is provided by HAProxy (https://www.haproxy.com/blog/four-examples-of-haproxy-rate-limiting/) and nginx (https://www.nginx.com/blog/rate-limiting-nginx/)

[3] https://www.ivoa.net/documents/IVOAIdentifiers/20160523/index.html

5 Implications for the Alert Filtering Service

In the current baseline, Rubin Observatory runs an Alert Filtering Service (AFS) within the Data Facility that allows a limited number of users (≥100 simultaneous) with Data Rights to upload filters and have a limited number of alerts (≤ 20 per visit) forwarded to them. In addition to the limited capacity, the AFS restricts the user filters to only operate on the contents of the alert packets themselves. There is currently no latency requirement for alert delivery through the AFS.

DMS-REQ-0343
numBrokerUsers
numBrokerAlerts

We suggest that the lightweight alert packet design would obviate the scientific need for the AFS, reducing Project scope. Instead of providing filter code to run on Data Facility services, users with data rights could run filters on their own hardware with a great deal more flexibility. In particular, users would be free to build filters that crossmatched to external catalogs or realtime streams or performed computationally-intensive fitting or image-processing tasks. They would be free to choose their own programming language, libraries, and environment. The need to implement a local service would impose modest additional difficulty relative to a Project-provided AFS; this has equity implications. The Project would at minimum need to provide tutorial material illustrating a basic implementation.

One particular advantage of this scheme to users is that within their overall rate limit budget, they could receive more alerts from single visits at the cost of higher latency. The AFS is likely to have had hard limits at 20 full-sized alerts per visit, after which no more would be forwarded. In contrast, a user wanting many more alerts from a high-priority single visit (say, a Deep Drilling field) could simply request more full-sized alerts over a longer period of time.

Descoping the AFS would reduce both effort and risk for the construction project and operations effort.

6 Implications for the Alert Database

The store of full-sized alerts is conceptually quite similar to the “Alert Database,” a component of the alert system that is required as a record of all transmitted alerts but that has relatively few other enumerated requirements. Particularly if flexible identifiers pointing to the full-sized alerts are used, it seems useful to investigate whether the Alert Database and the store of full-sized alerts could be the same system, technically. This would provide a seamless tran-

DMS-REQ-0094

sition for the user between active and archival alerts. Users could retrieve full-sized alerts by their URIs, and it would be straightforward to implement additional queries (cone search, time window, etc.) by querying the Prompt Products Database and retrieving the appropriate alert URIs. It would also be necessary to archive the lightweight alerts, but due to their small size this should not be a large perturbation[4].

7 Other Potential Approaches

Are there other ways to increase the reach of the alert stream in the baseline, full-size alert scenario? Given the scale of the full-sized alerts, any such approach would still rely on community brokers to provide alert access. However, the total number of community brokers could be increased by allocating more bandwidth at the Data Facility, by providing a ”fan-out” service replicating the alert stream, and/or by relaxing the 60-second latency requirement.

Another option to consider is to continue providing the baselined full stream to the general purpose brokers. The hybrid model could then be reserved for data rights holders, including specialized brokers, as a direct replacement for the Alert Filtering Service (§5).

8 Advantages

The most significant advantage of this system is that it makes alerts immediately available to a much broader audience. This was identified by attendees as a major priority from the 2019 Broker Workshop, in preference to a hard selection of a subset of the attending broker teams. “Alerts on your laptop” is also an exciting pitch for average scientists and is likely to provide large improvements in science use and uptake of the alert stream, even if users eventually choose to migrate to more full-service brokers.

We expect improved technical performance, as the the lightweight stream and the full packet archive systems can be designed separately. Bulk transport and rapid notification workloads tend to have quite different architectures.

While we will still meet the 60-second latency requirement, in practice we expect that this

[4] If the lightweight alerts are a strict subset of the full-sized alerts, they could be regenerated dynamically instead.

design will result in more effective use of the datacenter bandwidth. In the baseline design all of the alerts must be transmitted to the selected brokers within a relatively narrow window to meet latency requirements. The system will still be capable of this performance. However, since brokers will now control retrieval of the full packets, in practice they are likely to spread their queries out in time, smoothing the "peaky" nature of the traffic demands and enabling more users to make use of the connection.

This ability to control the flow of the data likely also provides a technical advantage for broker systems in handling the large data volumes. Within the lightweight stream, brokers will no longer need to stream through all of the bytes of a full alert in order to get to the next one, and they will then be able to fan out retrieval and processing of the full alerts in parallel.

Some science users have requested more information be included in the alerts—in particular, a third cutout image, and larger cutout images than the DPDD requires. These further increases to the size of already heavy alerts can be more easily accommodated if the delivery of the full alert payload is decoupled from the realtime notification.

While not a feature of the design presented here, we speculate that users may wish to retrieve only a portion of the full-sized alert—the cutouts, say, or the `DIASource` history. A small service extracting only the relevant alert portion would provide further improvements in bandwidth use.

Finally, the proposed system could replace the alert filtering service while providing potential interface improvements with the alert archive, as discussed in §5 and §6.

9 Challenges and Concerns

Among the technical challenges presented by the hybrid alert design is that it requires the Rubin Observatory project to design and administer two systems (the lightweight stream and the full-size alert retrieval system) rather than one (a full-sized alert stream). Given the late stage of the construction project it is reasonable to question whether it is prudent to consider such a change; the alert production system must be functional at the start of operations to support early science. However, the baseline also requires construction of the alert database and the alert filtering service, and as discussed (§5–6) the lightweight alerts design may provide some simplifications to those systems.

In our baseline, only a small number of alert brokers receive the Kafka stream. While moving to smaller alerts will improve Kafka's performance, distributing a lightweight Kafka stream to a much wider audience may require further attention. We do not anticipate challenges scaling the Kafka system itself to many consumers, but Kafka is primarily used as a streaming solution *within* organizations, and as of this writing its capabilities for handling many untrusted third-party consumers is less mature.

It will require effort and scientific consensus-building with the Rubin community to identify a minimal subset of alert contents for lightweight alerts that maintains its small size but provides the necessary information for efficient filtering. There will be community demand to repeatedly increase the number of included fields to satisfy individual science cases. A potential limiting case would be to include the entire triggering DIASource and corresponding DIAObject or SSObject records, resulting in a "middleweight" alert of about 2 kB. For a fixed bandwidth budget, the increased packet size would necessarily result in a smaller audience able to receive the notification stream; some degree of optimization may be possible.

While providing many more individual users direct access to the lightweight stream is a major aim of this design, it will increase the burden of support on the operations team. Some users who may have used third-party community brokers to access alert data will now connect directly to a Project resource and require support. And unlike in the baselined Alert Filtering Service, they will be processing this data on their own computer systems, increasing the complexity of the support task.

For the small number of brokers who are allowed to receive the full stream, the additional round-trip to retrieve the full alert packet imposes a small amount of additional latency (in the average case, roughly 100 msec). Additionally it requires brokers to re-engineer existing alert retrieval code. The handful of full-stream brokers may view these as pure costs with no benefit relative to the baseline. We believe these costs are justified by the gains of being able to distribute the alerts more widely; the modest additional latency is not scientifically significant. As discussed above, brokers may ultimately find it technically advantageous to be able to have their own systems direct and parallelize the retrieval of the largest volumes of data. The hybrid system may also make a variety of catch-up, re-sync, and completeness checking operations easier for brokers.

The most pressing concerns are those of data access. Alert packet contents are world public; this would include both the lightweight alerts and the full-sized alerts. However, access to databases and services within the Rubin Data Facility is restricted to data-rights holders [RDO-013]. If the same consideration applies here, only individual users with data rights and selected community brokers could subscribe to the lightweight stream and access the full-sized alerts directly[5]. As in the current baseline, users without data rights would rely on community alert brokers to access the public alerts. Since the lightweight alerts could be much more widely distributed to individual users, this situation would highlight more starkly the distinction between data rights and data access to the (world-public) alerts, which might create confusion or frustration among scientists.

Both the lightweight alert stream and the packet retrieval service would require authentication and authorization to confirm data rights and set appropriate rate limits. Individuals may find rate limits frustrating if their alert retrieval needs don't match their rate limits and may pressure the project for greater access rather than adapting their workflow to use a community broker.

The lightweight alerts contain a URI that points to the corresponding full alert packet. If a user without data rights is unable to access the packet at that location, having a copy of the world-public lightweight alert is not very useful. While community brokers are responsible for providing public access to alerts even in the baseline, they would need to provide their own redirection of the provided URI that to a publicly-accessible copy of the full alert, which could create confusion. From a pure user-experience perspective this suggests making access to the full alerts world-public, even if the rate limits for non-data-rights holders are extremely low.

Referential integrity of the lightweight and full alerts could be hard to maintain. Ensuring that the full alert is present for retrieval before publishing the lightweight alert could be guaranteed but may impact users' ability to rapidly retrieve the full alerts.

Because brokers must now actively retrieve the full alert packets, this inhibits architectures where the brokers provide only stateless real-time filtering or forwarding of alerts. This pattern does not appear common among the precursor brokers operating on ZTF, however.

A broad concern is whether lightweight alerts are beneficial to community brokers. We believe from a technical standpoint that brokers will benefit from being able to retrieve the full-size

[5] We do support investigating whether true world-public *access* to the lightweight alerts and full alerts with very low rate limits could be supported.

alerts when desired and in parallel, rather than being forced to seek through each alert to get to the next one. But for the top five selected brokers this benefit is probably not large. The greater advantage is to community brokers that might not otherwise be able (or want) to access the full alert stream, either directly from the project or from an “upstream” community broker. It is difficult to quantify this impact at present; the SAC’s evaluation of the full broker proposals will be useful here. And there are other alternatives for increasing the number of community brokers served (§7).

As discussed in §5, the biggest beneficiaries of the lightweight alert stream are likely to be single science users or small groups, who can now receive the complete lightweight alert stream and perform more sophisticated analysis than is possible with the baselined Alert Filtering Service. It may be argued that filtering and/or redistribution of alerts with fewer contents can or will be performed by the brokers themselves, and Rubin’s efforts are thus duplicative (and may even hinder broker teams’ ability to obtain funding). We expect that community brokers will be the dominant way that users will access LSST alerts thanks to their scale, ease of use, and integration with other facilities. However, the Project aims to maximize the scientific return from the survey; making lightweight alerts broadly available will not hinder the use of community brokers and are likely to enable fresh new ideas and applications. Additionally, a Rubin-provided service has the advantage of guaranteed stability and longevity over the lifetime of the survey.

A Possible lightweight alert packet contents

Table 1 provides a list of candidate fields that could be included in a lightweight alert and enable substantial pre-filtering by science users before they needed to retrieve the full alert packets corresponding to a subset of alerts.

B References

[RDO-013], Blum, R., the Rubin Operations Team, 2020, *Vera C. Rubin Observatory Data Policy*, Data Management Operations Controlled Document RDO-013, NSF-DOE Vera C. Rubin

Field	Type	bytes
alert contents URI	varchar(80)	80
diaSourceId	unit64	8
filterName	unit8	1
programID	unit16	2
diaObjectId	unit64	8
ssObjectId	unit64	8
midPointTai	double	8
ra	double	8
dec	double	8
psFlux	float	4
psFluxErr	float	4
totFlux	float	4
totFluxErr	float	4
trailLength	float	4
extendedness	float	4
spuriousness	float	4
number of previous detections	int16	2
time of most recent observation	double	8
totFluxMean	float	4
totFluxSigma	float	4
distance to nearest star	float	4
distance to nearest galaxy	float	4
parallax/PM	3 floats	12
other timeseries or SSObject features	float	TBD

TABLE 1: A potential set of fields for a lightweight alert packet of ~200 bytes.

Observatory, URL https://ls.st/RDO-013

[LSE-29], Claver, C.F., The LSST Systems Engineering Integrated Project Team, 2017, *LSST System Requirements (LSR)*, Systems Engineering Controlled Document LSE-29, NSF-DOE Vera C. Rubin Observatory, URL https://ls.st/LSE-29

[LSE-61], Dubois-Felsmann, G., Jenness, T., 2019, *Data Management System (DMS) Requirements*, Systems Engineering Controlled Document LSE-61, NSF-DOE Vera C. Rubin Observatory, URL https://lse-61.lsst.io/, doi:10.71929/rubin/2587200

[DMTN-102], Graham, M.L., Bellm, E., Guy, L., et al., 2024, *LSST Alerts: Key Numbers*, Data Management Technical Note DMTN-102, NSF-DOE Vera C. Rubin Observatory, URL https://dmtn-102.lsst.io/, doi:10.71929/rubin/2997858

[LSE-163], Jurić, M., Axelrod, T.S., Becker, A.C., et al., 2023, *Data Products Definition Document*, Systems Engineering Controlled Document LSE-163, NSF-DOE Vera C. Rubin Observatory, URL https://lse-163.lsst.io/, doi:10.71929/rubin/2587118

Patterson, M.T., Bellm, E.C., Rusholme, B., et al., 2019, PASP, 131, 018001 (arXiv:1902.02227), doi:10.1088/1538-3873/aae904, ADS Link

[DMTN-093], Patterson, M.T., Bellm, E.C., Swinbank, J.D., Nelson, S., Smart, B.M., 2024, *Design of the LSST Alert Distribution System*, Data Management Technical Note DMTN-093, NSF-DOE Vera C. Rubin Observatory, URL https://dmtn-093.lsst.io/, doi:10.71929/rubin/2586493

C Acronyms

Acronym	Description
Avro	is a row-oriented remote procedure call and data serialization framework developed within Apache's Hadoop project
DMTN	DM Technical Note
DOI	Digital Object Identifier
DPDD	Data Product Definition Document
GB	Gigabyte
HTTP	HyperText Transfer Protocol
KB	KiloByte
LMC	Large Magellanic Cloud

LSE	LSST Systems Engineering (Document Handle)
LSST	Legacy Survey of Space and Time (formerly Large Synoptic Survey Telescope)
NEO	Near-Earth Object
PM	Project Manager
RDO	Rubin Directors Office
SAC	Science Advisory Committee
TBD	To Be Defined (Determined)
ZTF	Zwicky Transient Facility
kbps	kilobits per second