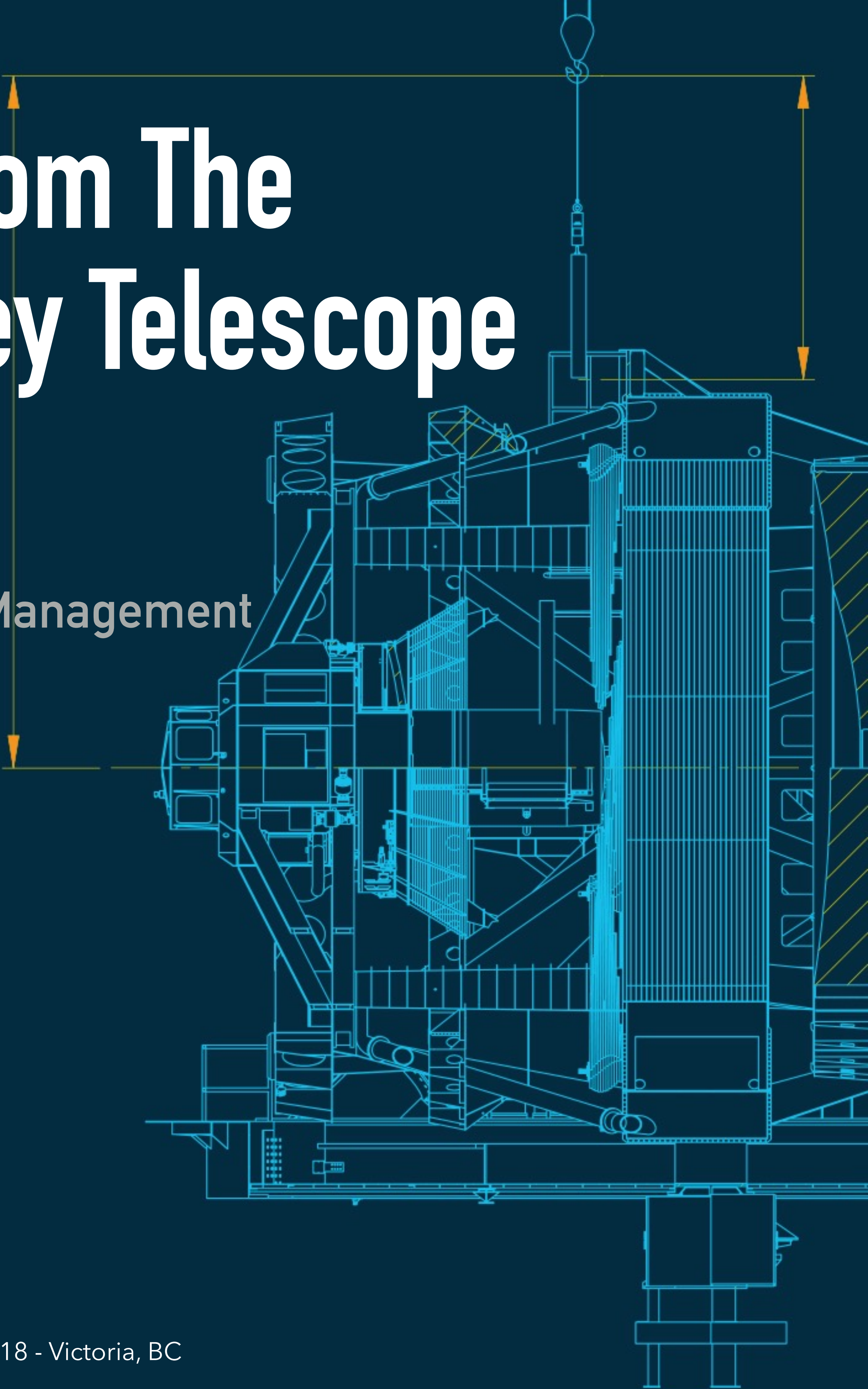
Distributing Alerts From The Large Synoptic Survey Telescope

Eric Bellm
University of Washington
Alert Production Science Lead, LSST Data Management

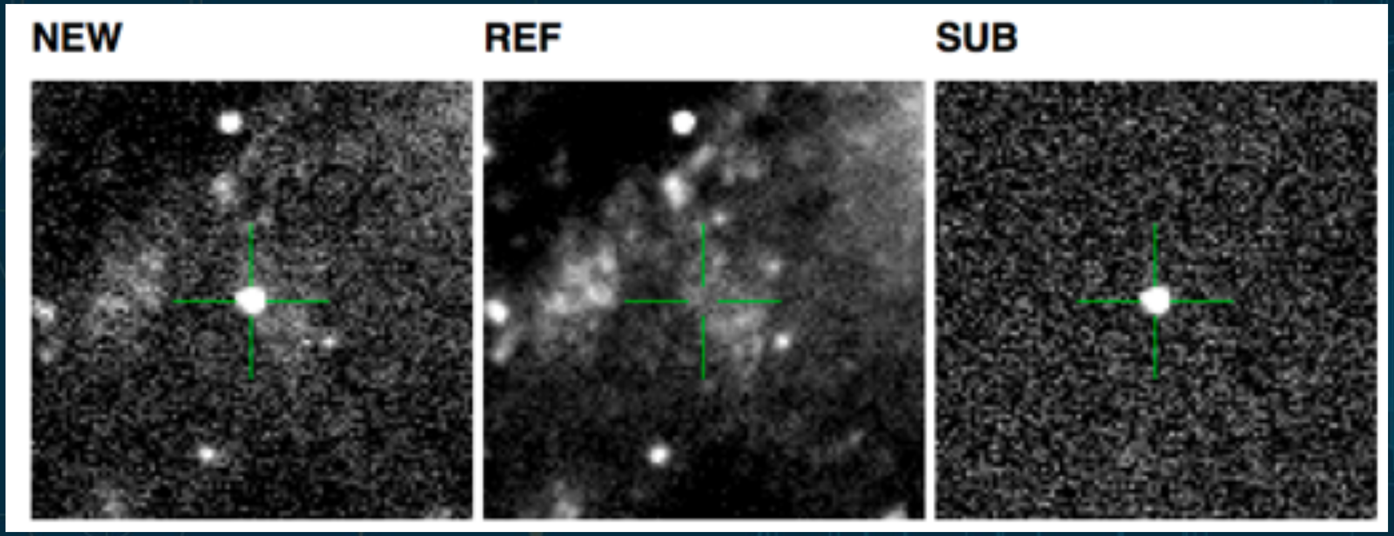


Large Synoptic Survey Telescope

IVOA InterOp - 29 May 2018 - Victoria, BC

The LSST alert stream will enable cutting edge time-domain science.

Streams *all* difference image detections (~10M/night)



Alerts are world-public and packaged with contextual information

Source measurements, a one year lightcurve, timeseries statistics, crossmatches to the data release, and image cutouts

Delivered within 60 seconds of shutter close!

The alert stream has two destinations.

Community Brokers

A small number (~4) of brokers will receive the full alert stream

limited by bandwidth out of the datacenter

Brokers perform crossmatches to external catalogs, provide classifications, and provide public access to the alert contents.

The LSST Simple Filtering Service (“mini-broker”)

Users with LSST Data Rights may access a simple, limited-capacity alert filtering service

Applies user-supplied filters that evaluate alert packet contents only

VOEvent 2.0 & VTP will not satisfy LSST's requirements.

Semantic content

No major concerns.

Serialization

LSST is required to embed image cutouts in the packets.

XML serialization is bandwidth inefficient.

Transport

VTP is too chatty for 10M+ alerts/night (greater than lifetime 4 Pi Sky archive)

We'd also like brokers to be able to recover/replay alerts if needed.

We have prototyped open-source technologies that meet LSST requirements for bulk transport.

Serialization: Apache Avro

Binary, strictly-schemaed serialization format

6x smaller than equivalent XML & 40x faster (de)serialization (Maeda 2012)

prototype: https://github.com/lsst-dm/sample-avro-alert

Transport: Apache Kafka

Distributed real-time message queue

Scales to trillions of daily messages

Stream consumers can replay topics as desired

prototype: https://github.com/lsst-dm/alert_stream

LSST-scale benchmarks: https://dmtn-028.lsst.io/

We expect to use this bulk-transport protocol to send full streams to community brokers.

Have to be able to demonstrate to our funders that we can meet requirements.

The few community brokers are chosen by a proposal process;
choice of serialization format and transport protocol is an internal issue,
part of that negotiated agreement.

ZTF has adopted a fork of this stack (cf. previous talk by M. Graham).
We are monitoring the ZTF experience with these tools closely
& will incorporate lessons learned & community feedback.

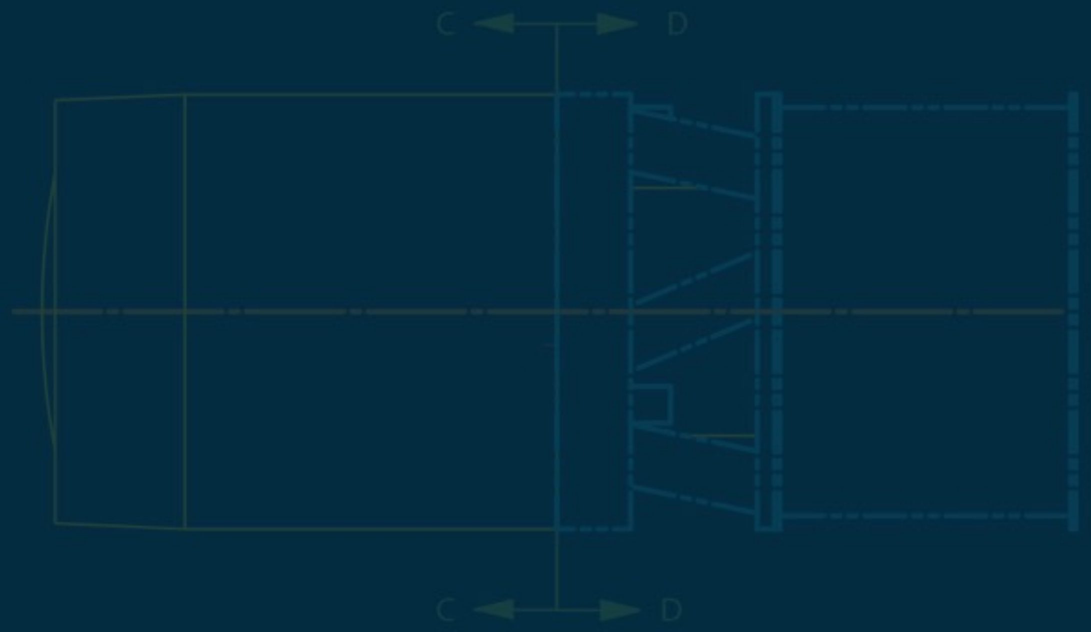
We expect the mini-broker to provide VOEvent.

As a community-facing resource, mini-broker should also be "VO-first."

Clear use cases where users will want streaming output and interoperability.

Sharply reduced streams (10k->20 alerts per visit) means VOEvent 2.0 + VTP are workable as an output channel.

We are still resolving many details surrounding the technical implementation of the mini-broker system; internally it is likely to use Avro+Kafka.



We are interested in contributing to a VOEvent 3 that would be closer to our bulk transport format.

Semantic content

Defining a data model independent of the XML schema.

Adjust LSST alert formats to correspond more closely to VOEvent content.

Serialization

Providing reference implementations of the Avro serialization.

Transport

We are using “vanilla” Kafka—not clear this requires standardization.

Conclusions

1) LSST has a format and transport mechanism that meets its requirements for bulk transport to community brokers.

2) As part of our commitment to VO interfaces we expect the mini-broker to speak VOEvent (whether v2 or some evolution) to the community.

3) We are interested in contributing to discussions that could bring #1 and #2 closer together, both in serialization and transport.