

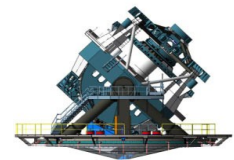


# Alertsim - Serbian contribution to the LSST

**Darko Jevremović**  
**Astronomical Observatory Belgrade**

**LSST@EUROPE2**  
**Belgrade, June 22 2016**





# Belgrade group



- We got involved fairly -early - interest in '09/'10
- MOA's for 4 PI in 2013
- Veljko Vujčić (CS – CEP/esper); Miodrag Malović (Walsh, period-shape detection); Yana Khusanova obs\_sofi; Vladimir Srećković (databases&services); Jovan Aleksić (astronomy);
- Background stellar atmospheres, stellar flares, use of HPC in astronomy + VO and astroinformatics



# LSST movie



- [/home/darko/enigma\\_1189\\_n25\\_n3001.mp4](/home/darko/enigma_1189_n25_n3001.mp4)

Thx to Lynne Jones and Peter Yoachim – LSST Simulation group



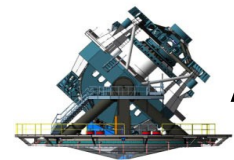
- Alerts - Level 1 (nightly) data products
- Anything that changes on sky goes in alert
- Release 60 seconds after the visit to the world wide community
- Expect many alerts (10k/visit)
- 2-4 public brokers(due to bandwidth constraints)
- Simulator necessary for validating & testing brokers (Mario identified)



# AlertSim - requirements



- Generating realistic streams of LSST transient alerts
- Simulating various failures or exceptional/extreme operations modes:
  - Unexpectedly large numbers of spurious detections
  - Large number of detections (dense fields)
  - Disruptions of event stream
  - Corruption of event stream
  - Network connectivity interruptions



# AlertSim - requirements



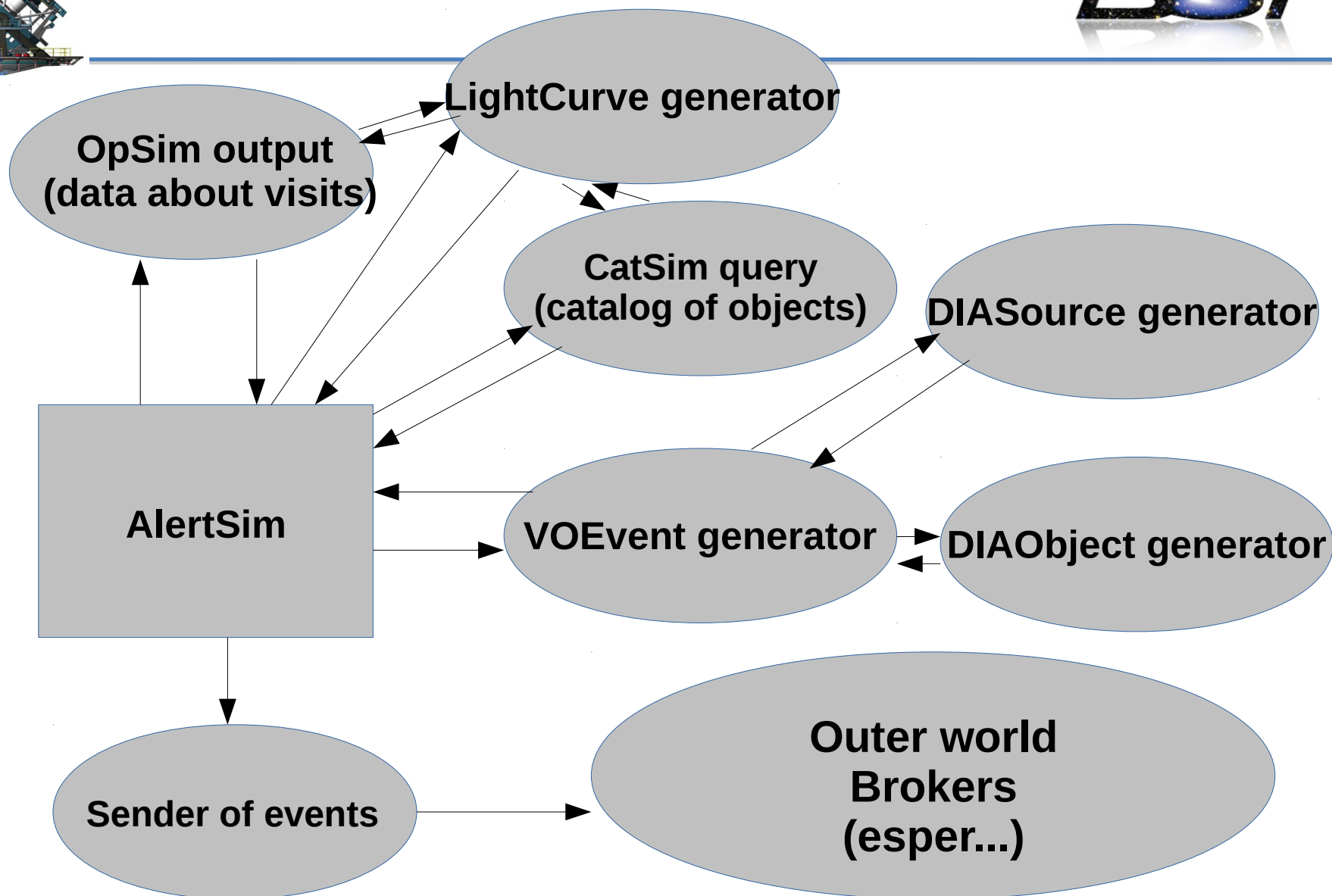
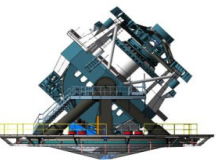
- Provide facilities to ease troubleshooting problems with broker end-points
- configurable, automated and capable of keeping provenance
- Written following LSST software standards, conventions and development processes and executable on LSST Data Access Center hardware
- Developed in coordination with capabilities provided by LSST Simulations group and DM team



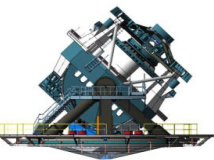
# AlertSim - prototype



- First 'alert' end 2014
- Python based using low level functions from socket (allows not only TCP/IP but multicast or similar – idea is to use streaming capabilities for alert stream)
- Query to opsim output and catsim database and generate alerts
- Pack alerts to VOEvent (requirement about standard – could be changed )
- VOEvent XML shema... very impractical
- Make it a service...







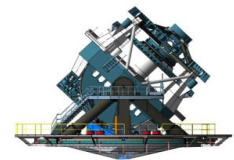
- Basic idea is to provide brokers (Antares etc.) heads up playground
- (we had to calculate too many light curves for Antares...)
- Django based (experience with VAMDC!)
- Easily adaptable for machine queries (requests)!
- Simple form to fill to choose:
  - ip address and port
  - Different local or remote opsim databases
  - Different catsim tables
  - Parameters for querying databases



# AlertSim - service



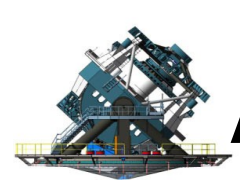
- History calculated from opsim and variability mixin inside catsim (slow, problems with single magnitude...)
- Pack history(ical light curves) in diasource chunks (or emit complete diasources)
- Necessary connection to UW databases (is it wise to have it as a service??)
- For how long we want service to run?
- Note that we connect mainly to the stellar stuff – galaxies & sso in the future



# AlertSim - problems

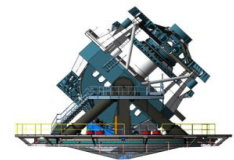


- Lot of hacks to achieve service – problems with environment variables and setting up eups from outside the shell
- Queries may return huge outputs
- should we make lcg a separate package? Problems with distribution sky for particular types of objects....
- Older routers may cut messages(need to repeat or divide)
- Proxy problems – ports are closed
- VOEvent/XML does not like mages
- Possible to encode images in XML – but consider separate mechanism
- (problems are good – force us to think outside of the box)



# AlertSim and DM end-to-end simulations

- Connection with results of end-to-end simulations
- Idea is to generate simulated images out of what is in catsim (using phosim or galsim and some level of noise)
- Process those images with LSST Stack (imdiff on simulated or other images)
- DM Stack should in (not too distant) future become capable of generating alerts or at least DIASources for detected features
- If DMstack sends us DIASource(s) we are able to pack it in VOEvent format and forward it further...



# AlertSim - todo



- Cutouts – i.e. force galsim to do small patches
- Esper – playing with aggregates and other functionalities (control of what is happening)
- Collect engineering simulated data (through opsim or otherwise) and use in decision making process or at least simulate and test several byte quality stamp??
- Detecting readiness of clients (at the moment we ignore – just emit)
- Parallelization (brute force vs. clever)
- Build DSL (and classifiers) on top of esper which will be understandable to astronomers and make their life easy in LSST era



# AlertSim - demo of service

---



- Control parameters
- We use RRLyrae/allstars
- sending and receiving xml
- esper



# Few things to remember

---



- Alerts sim is (will be) capable to provide near realistic service of LSST alert stream
- Good starting point to train different brokers, classifiers...
- More functionalities will come with time