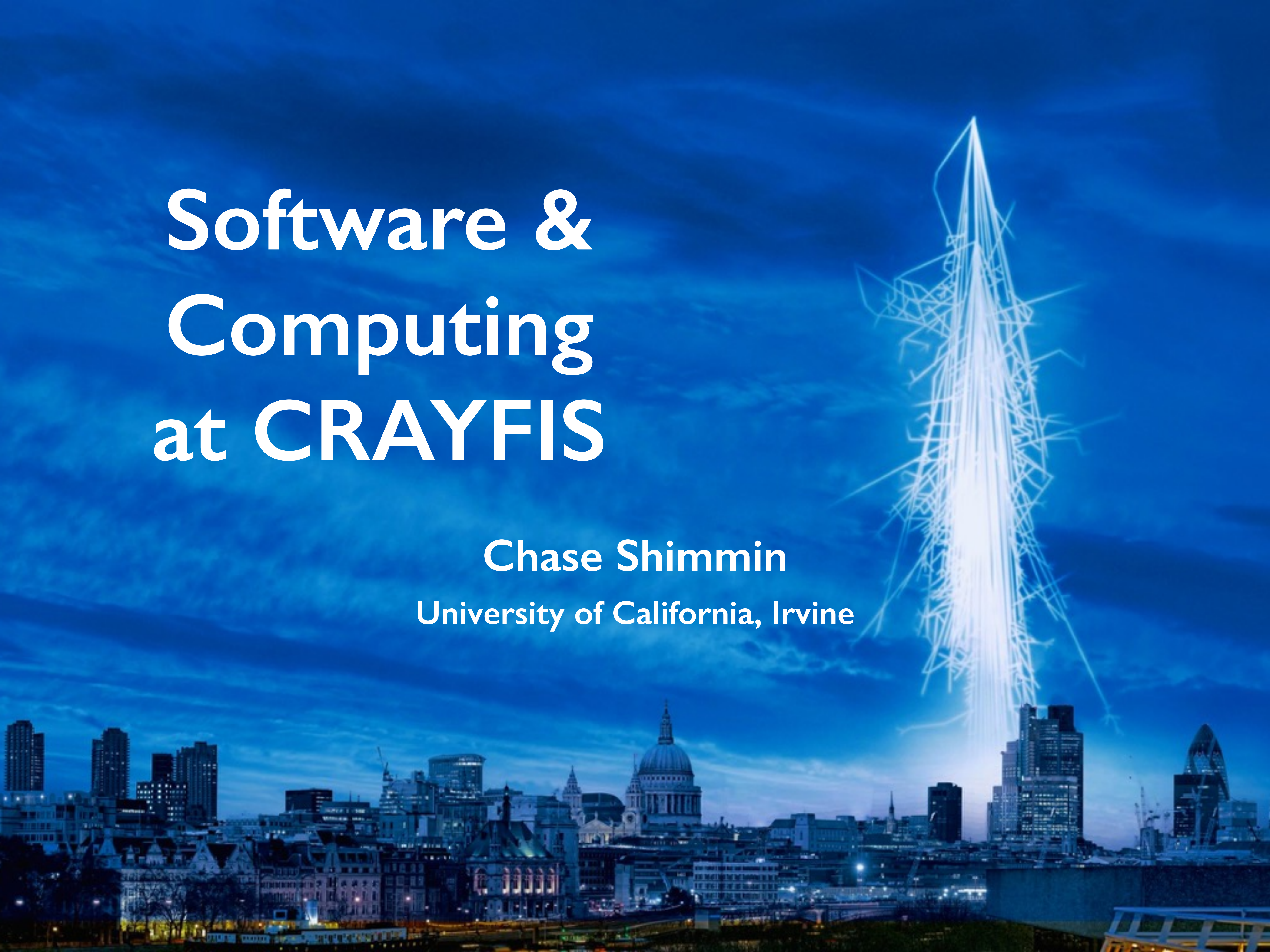


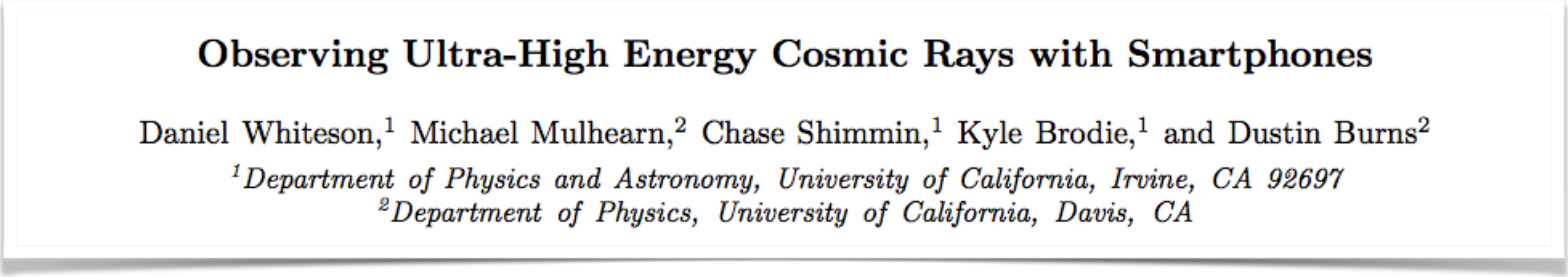
Software & Computing at CRAYFIS

Chase Shimmin

University of California, Irvine

CRAYFIS

cosmic rays found in smartphones



Observing Ultra-High Energy Cosmic Rays with Smartphones

Daniel Whiteson,[1] Michael Mulhearn,[2] Chase Shimmin,[1] Kyle Brodie,[1] and Dustin Burns[2]

[1] *Department of Physics and Astronomy, University of California, Irvine, CA 92697*
[2] *Department of Physics, University of California, Davis, CA*

arXiv:1410.2895

CRAYFIS

cosmic rays found in smartphones

Whiteson
Shimmin
Strong
Brodie
Goddard
Porter
Sandy

NYU

Cranmer

Яндекс
Yandex

Ustyuzhanin
+2 masters st.

UCDAVIS
UNIVERSITY OF CALIFORNIA

Mulhearn
Burns
Buonacarsi

NAOC

Deng

powered by
amazon
web services

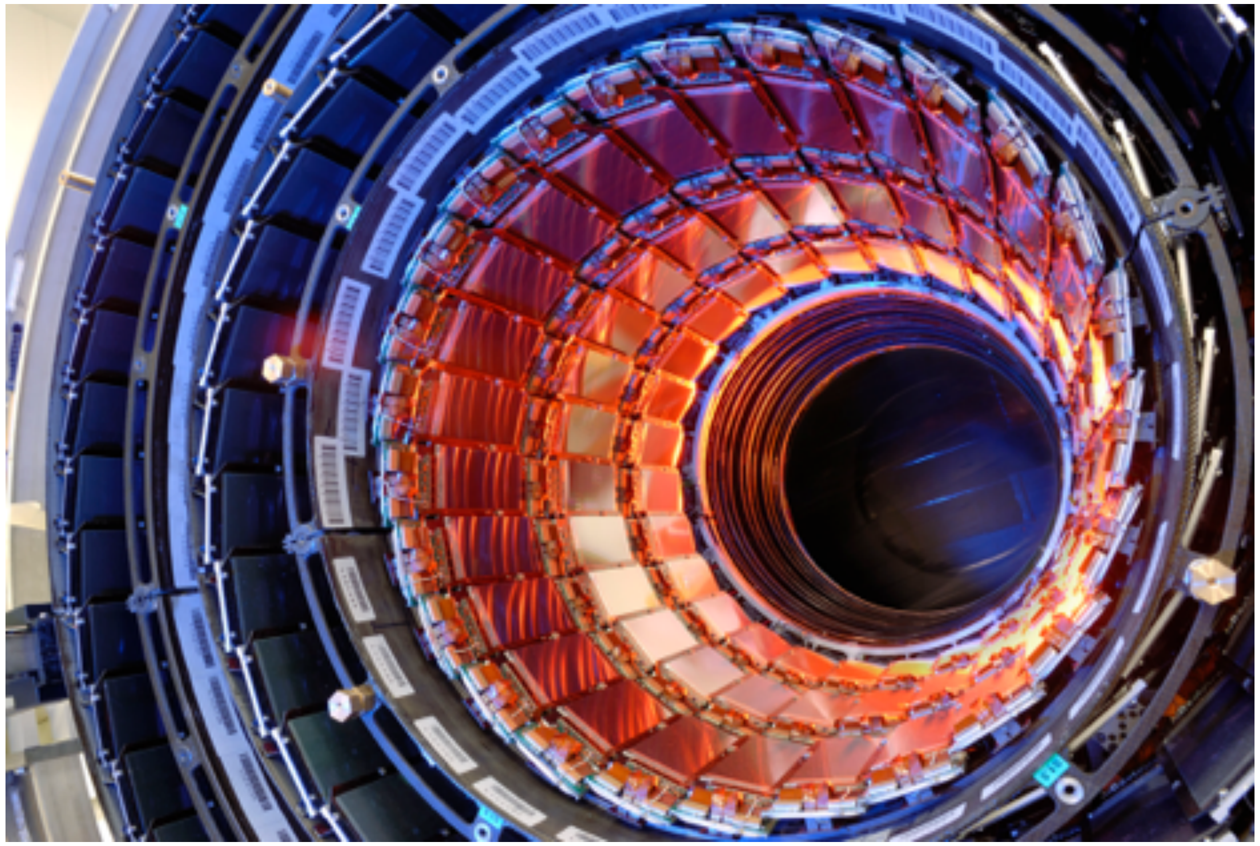
Smartphones are:

(tiny) Particle Detectors

Camera Sensor

(Active area: ~0.3 cm^2)

=



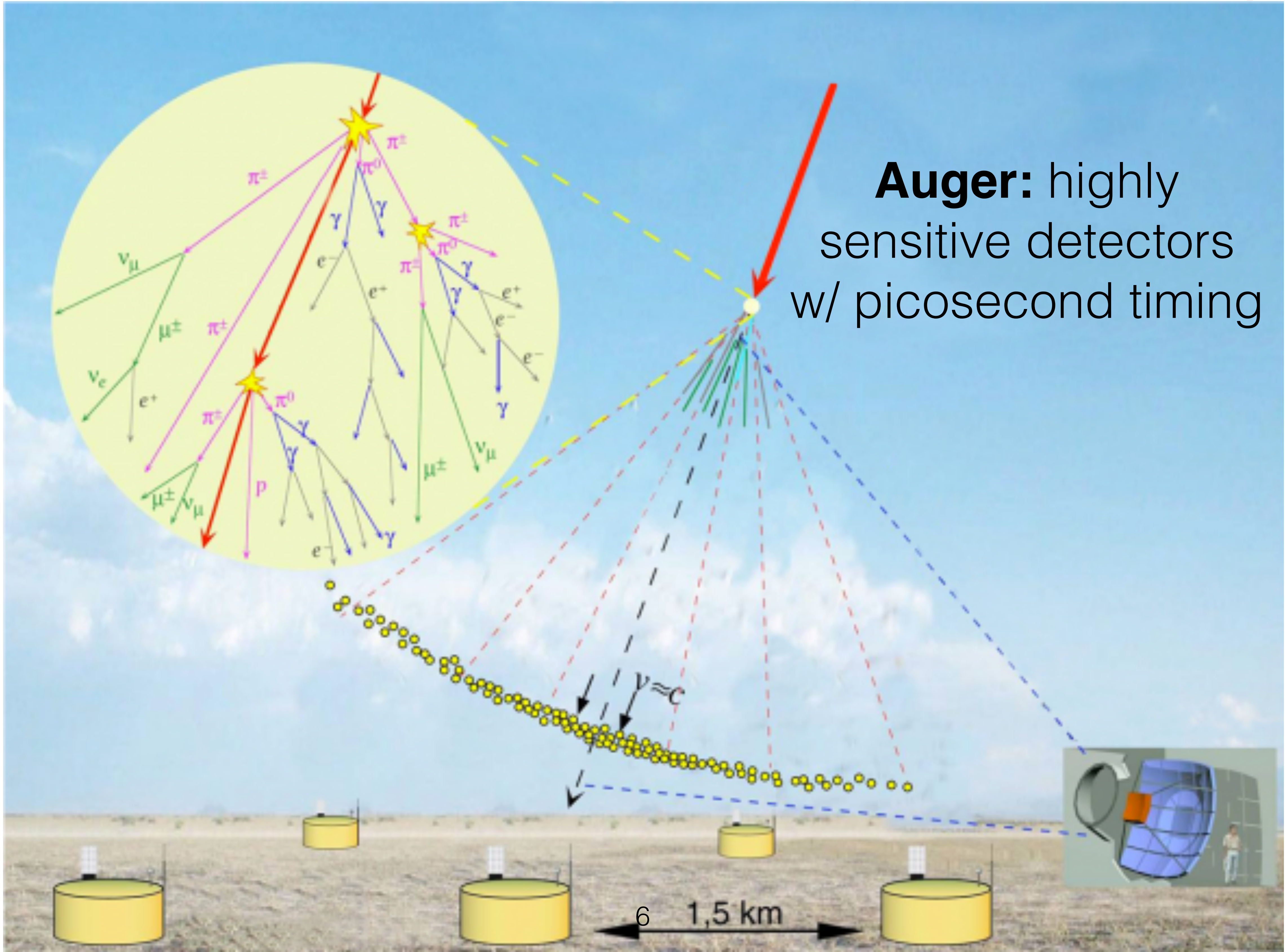
Smartphones are:
Mobile Laboratories

GPS

Wi-Fi

Shower Reconstruction

(State of the Art)

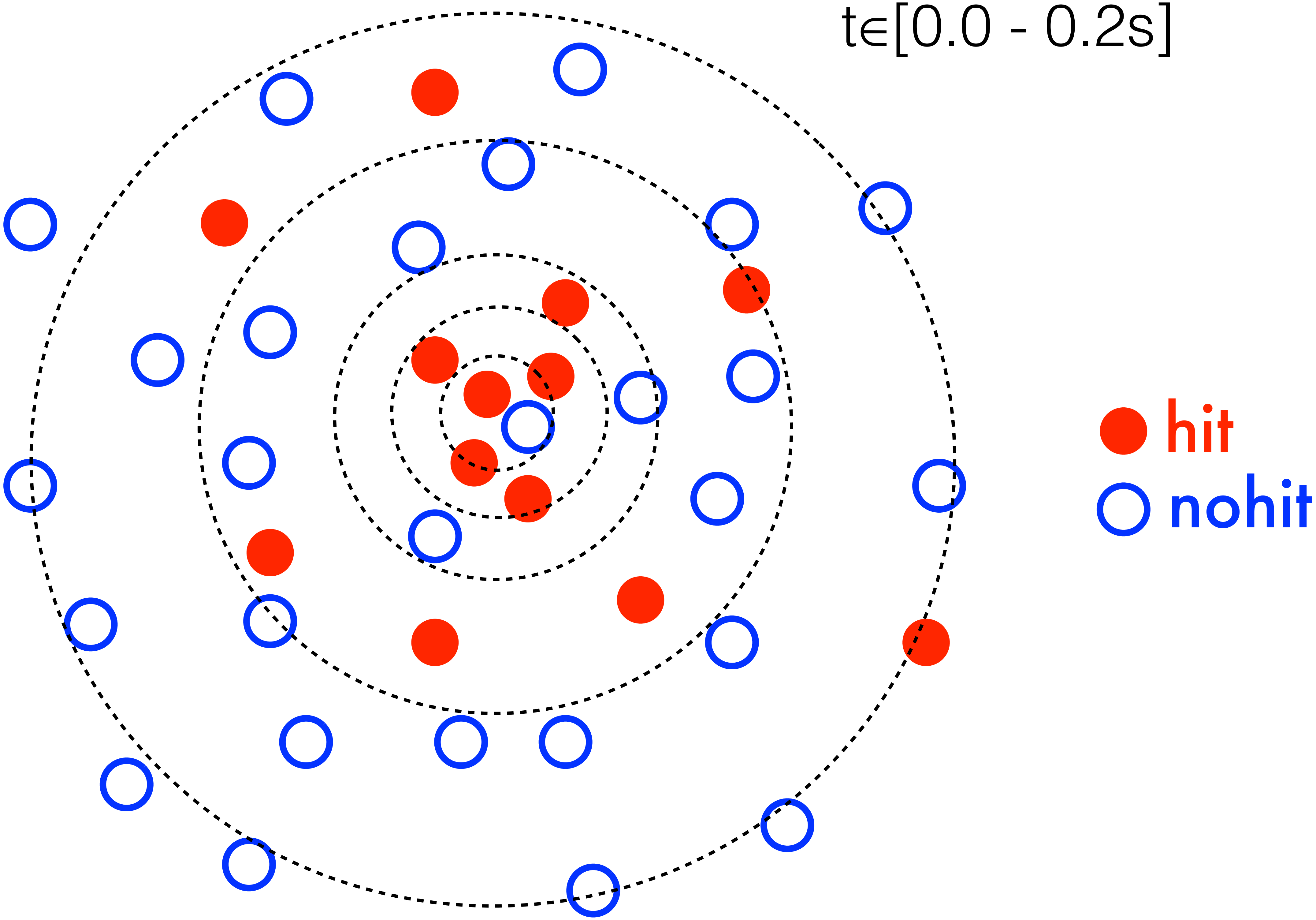


Auger: highly sensitive detectors w/ picosecond timing

Shower Reconstruction

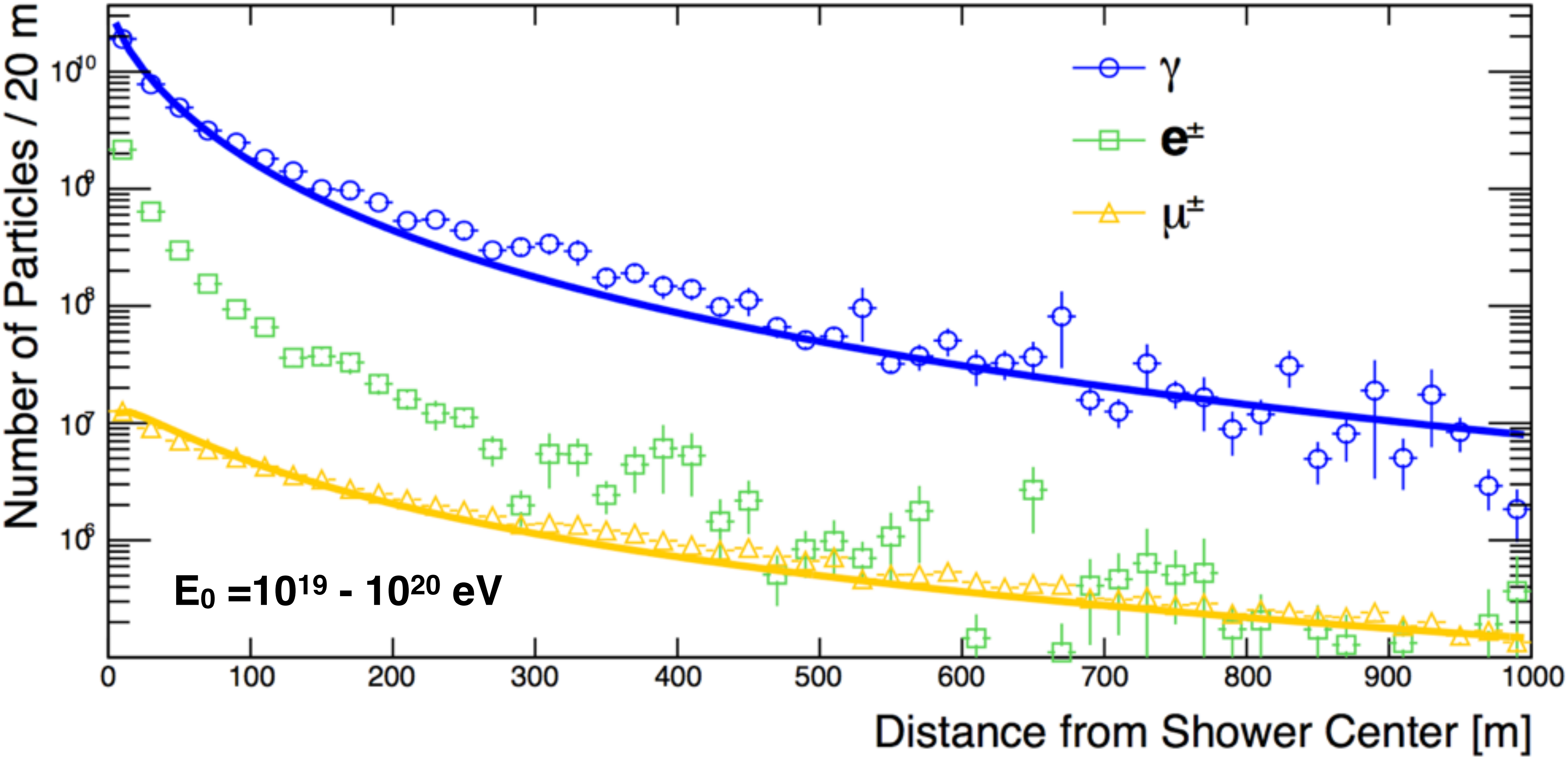
(w/ CRAYFIS)

t∈[0.0 - 0.2s]



Particle Content

Tremendous densities
near shower core

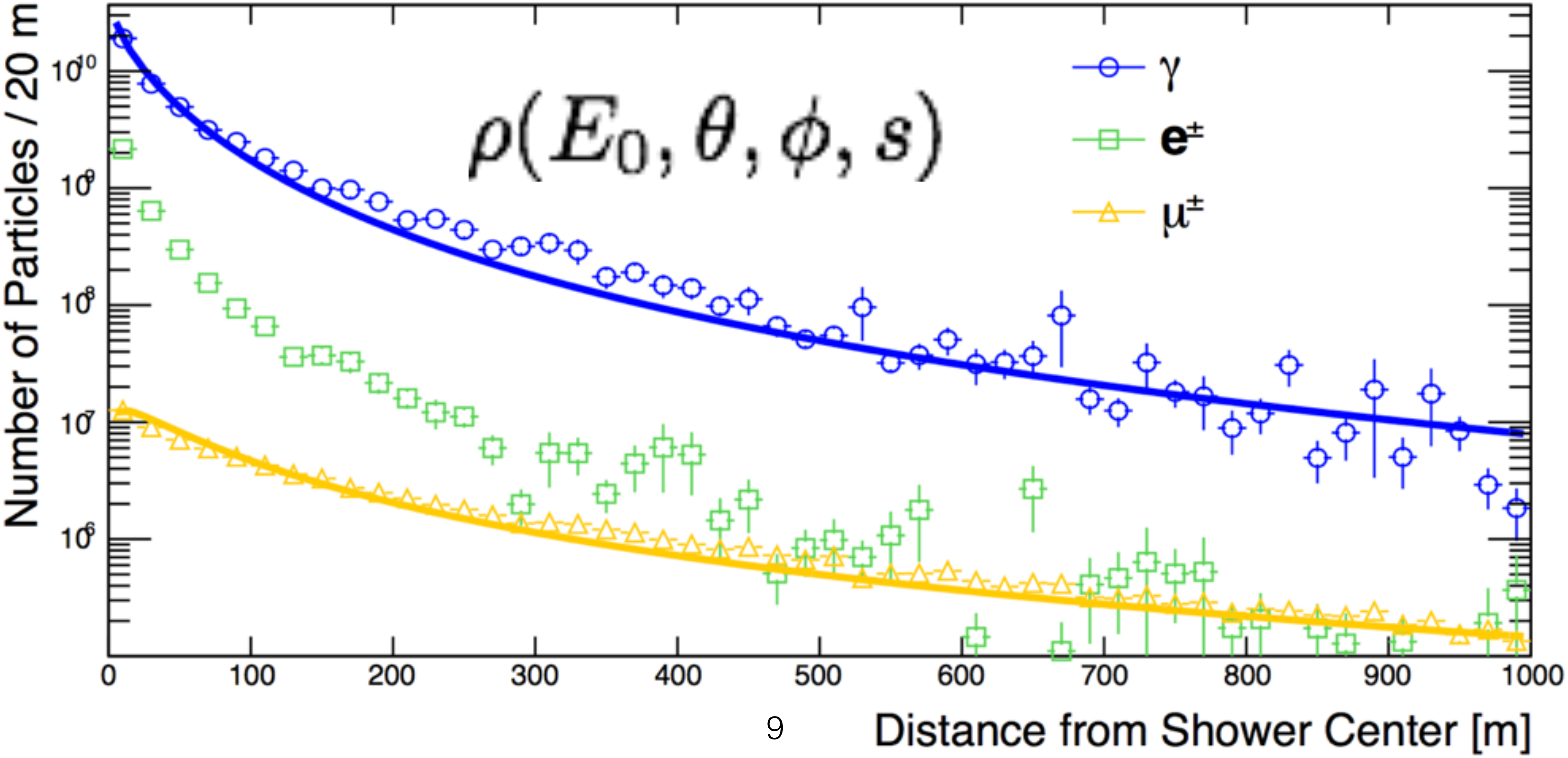


Likelihood

During a shower event, the expected number of particle hits is:

$$\lambda = A\epsilon \cdot \rho(x, y) + \eta$$

A — active area
ϵ — detection eff.
ρ — LDF [particles/m^2]
η — noise term



Likelihood

During a shower event, the expected number of particle hits is:

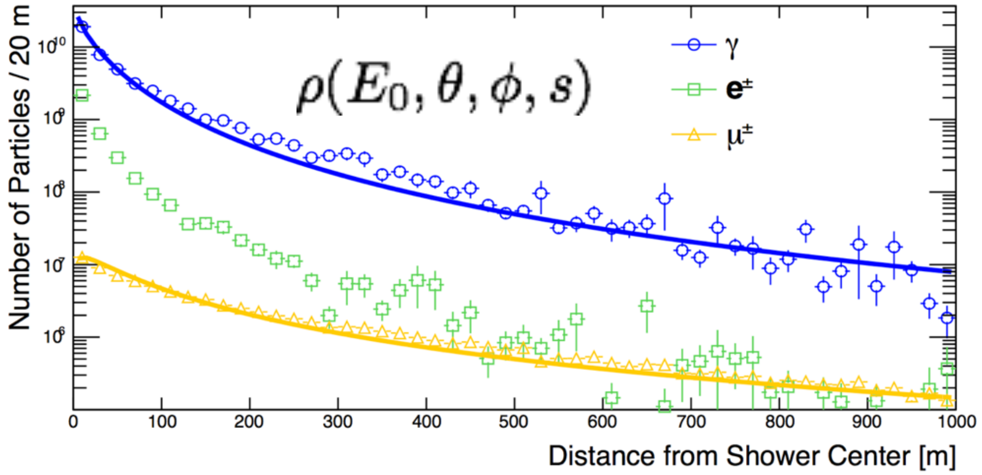
$$\lambda = A\epsilon \cdot \rho(x, y) + \eta$$

A — active area

ϵ — detection eff.

ρ — LDF [particles/m^2]

η — noise term



Probability of seeing nothing:

$$P_0(x, y) = e^{-\lambda}$$

Likelihood

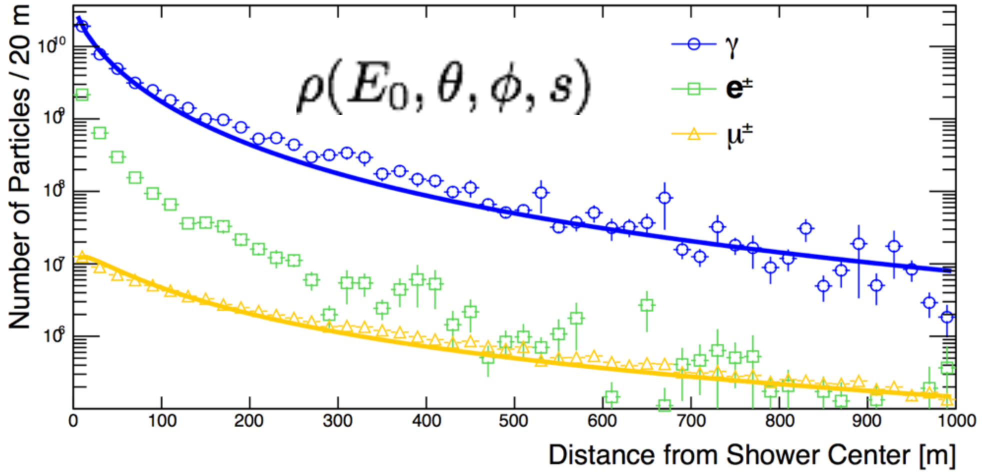
During a shower event, the expected number of particle hits is:

$$\lambda = A\epsilon \cdot \rho(x, y) + \eta$$

A	— active area
ϵ	— detection eff.
ρ	— LDF [particles/m^2]
η	— noise term

Probability of seeing nothing:

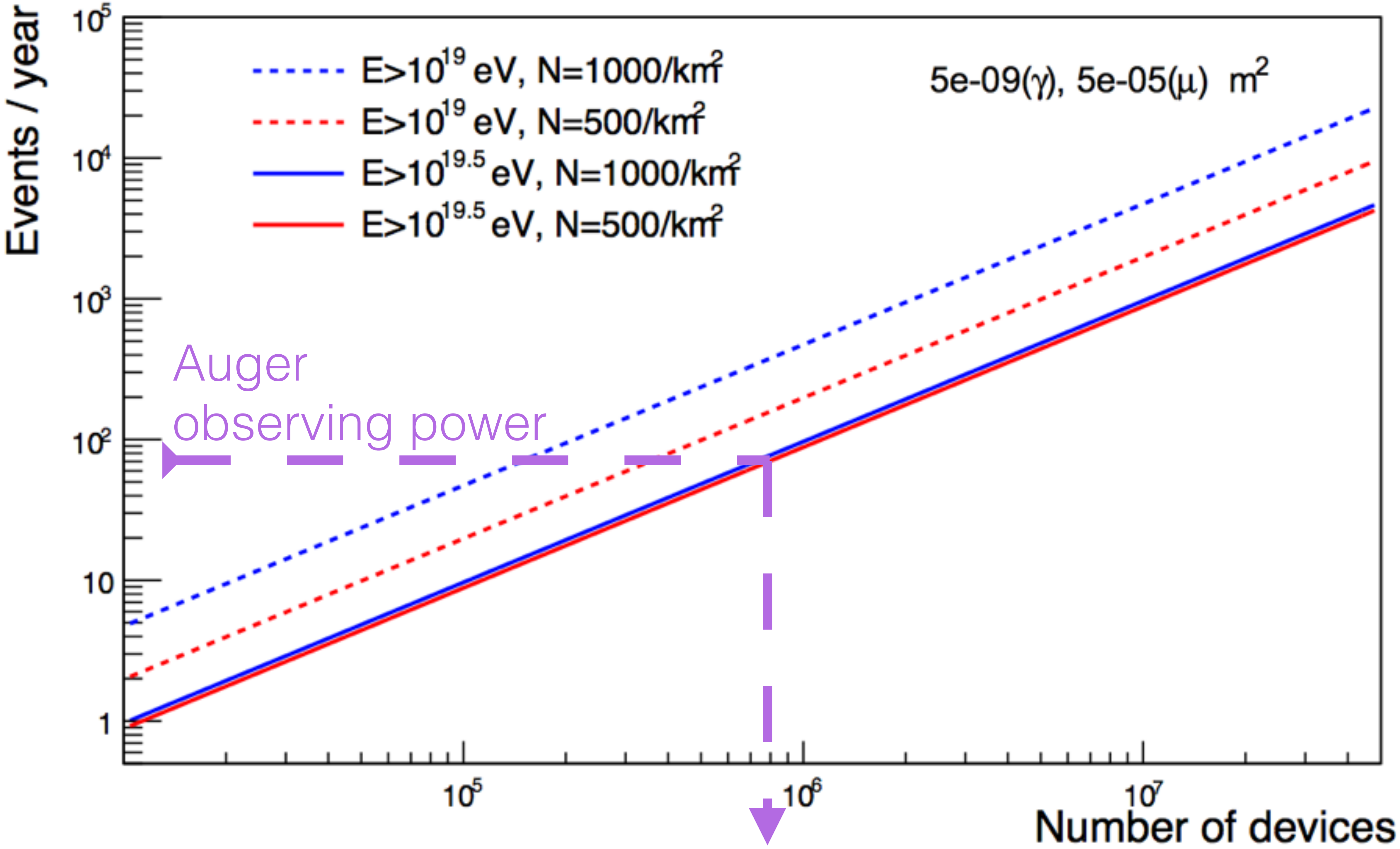
$$P_0(x, y) = e^{-\lambda}$$



Likelihood function, given phones that were / weren't hit:

$$L(E_0, \theta, \phi, s) = \underbrace{\prod_i P_0(x_i, y_i)}_{\text{no hit}} \underbrace{\prod_j (1 - P_0(x_j, y_j))}_{\text{hit}}$$

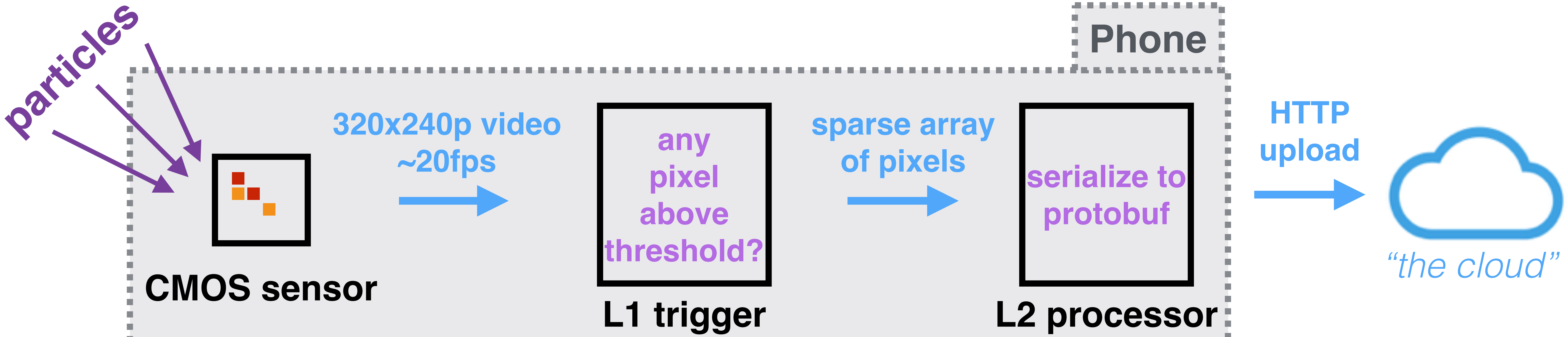
Keeping up with Auger



Auger observing power

~800k smartphones!

The App: Internals



Basic Idea:

Scan video for bright pixels.
Upload any hits to our server.

Data Challenges

Online/DAQ:

- Acquire data from many endpoints all over the world
- Robustly store data to persistent site
- Aggregate and process analytics in realtime

Offline/Analysis:

- Generate/index calibration for 1M+ unique sensors
- Cluster events in space +time
- Convolve array with shower MC to determine acceptance

Online Computing

CRAYFIS DAQ load:

(1M phones) x $(3\text{-}5\ \text{min})^{-1}$ ~ 3-6 kHz

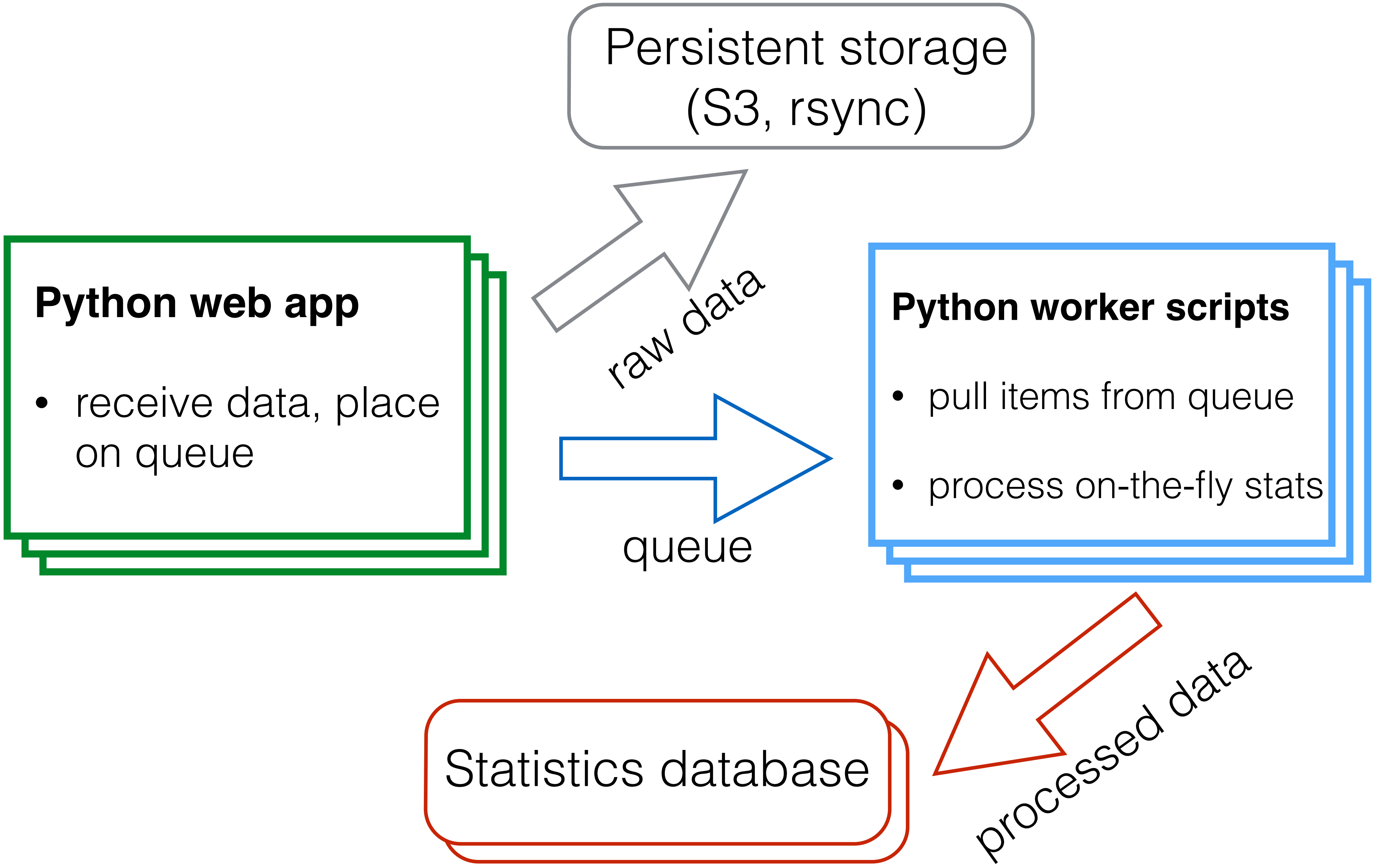
Google searches:
40 kHz

Reddit pageviews:
<100 Hz

- Phones generate datablocks every ~3-5 min
- Each one needs to be received, saved, and processed by us!

Online Backend

(simplified)



Persistent storage (S3, rsync)

raw data

Python web app

- receive data, place on queue

queue

Python worker scripts

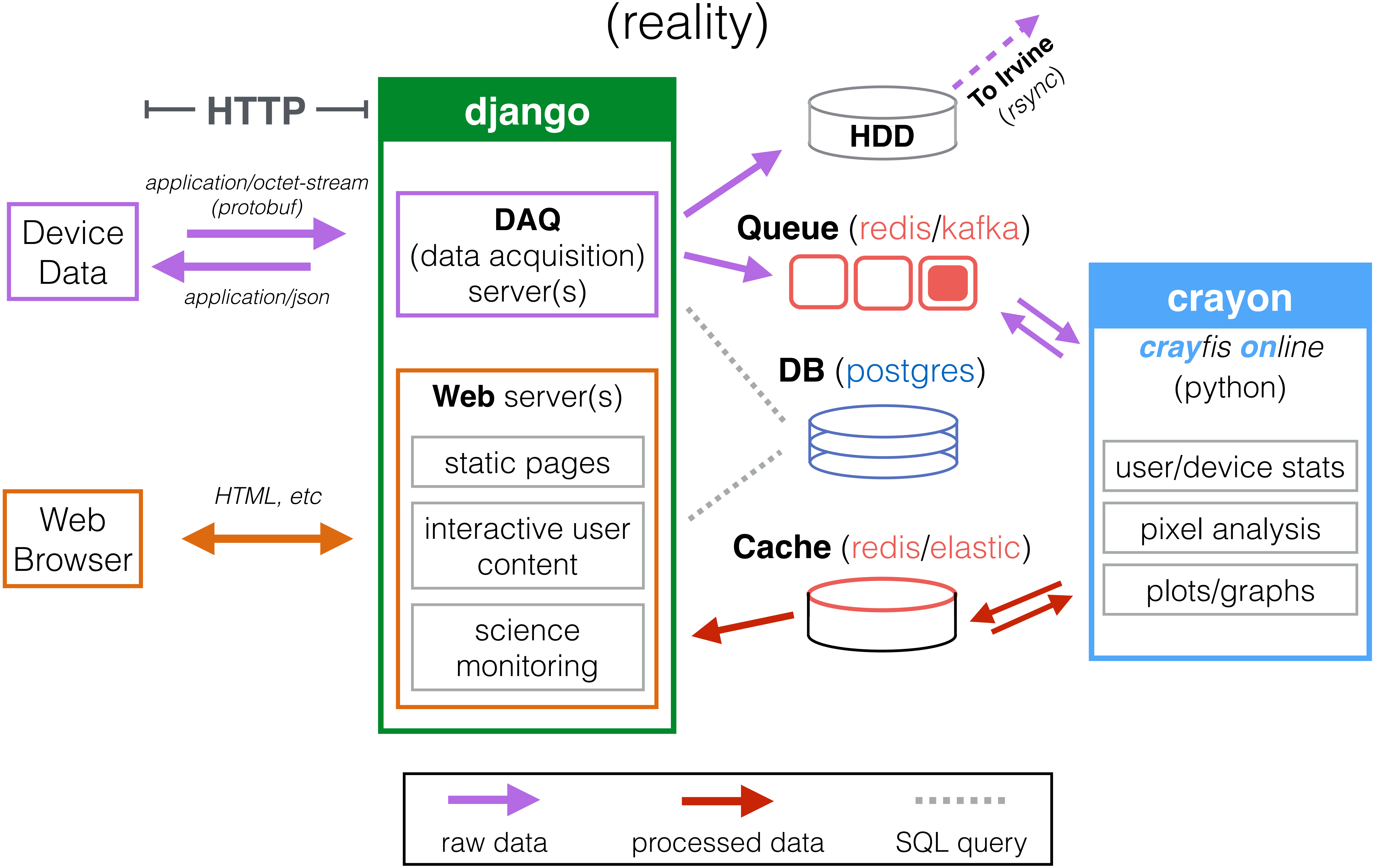
- pull items from queue
- process on-the-fly stats

processed data

Statistics database

Online Backend

(reality)



HTTP

application/octet-stream (protobuf)

application/json

HTML, etc

Device Data

Web Browser

django

DAQ (data acquisition) server(s)

Web server(s)

- static pages
- interactive user content
- science monitoring

HDD

To Irvine (rsync)

Queue (redis/kafka)

DB (postgres)

Cache (redis/elastic)

crayon

***cray**fis **on**line*

(python)

- user/device stats
- pixel analysis
- plots/graphs

raw data

processed data

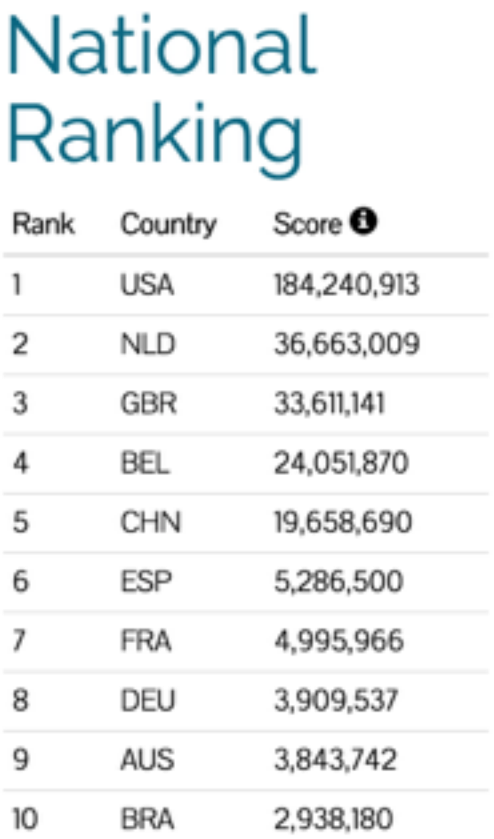
SQL query

Online Frontend

Website: http://crayfis.io

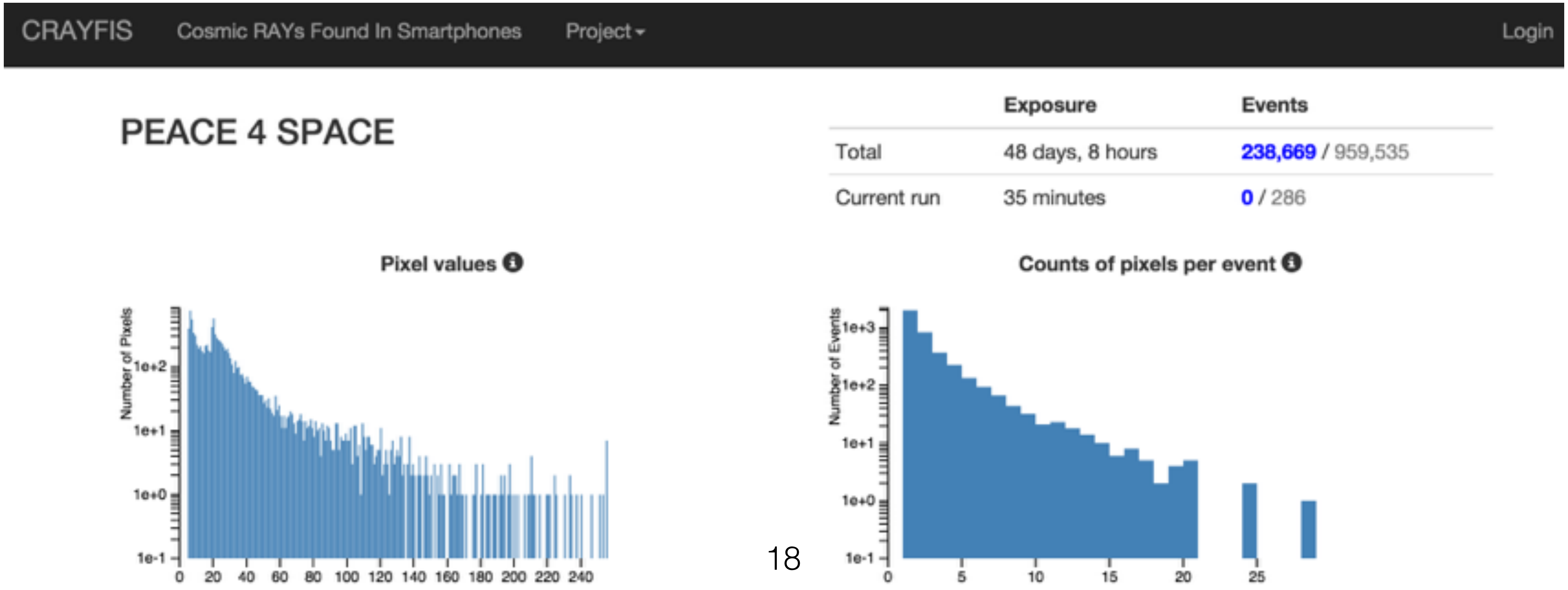
- Stats and maps
- Plots from each device
- All updated in realtime

Network Map



National Ranking

Rank	Country	Score
1	USA	184,240,913
2	NLD	36,663,009
3	GBR	33,611,141
4	BEL	24,051,870
5	CHN	19,658,690
6	ESP	5,286,500
7	FRA	4,995,966
8	DEU	3,909,537
9	AUS	3,843,742
10	BRA	2,938,180



PEACE 4 SPACE

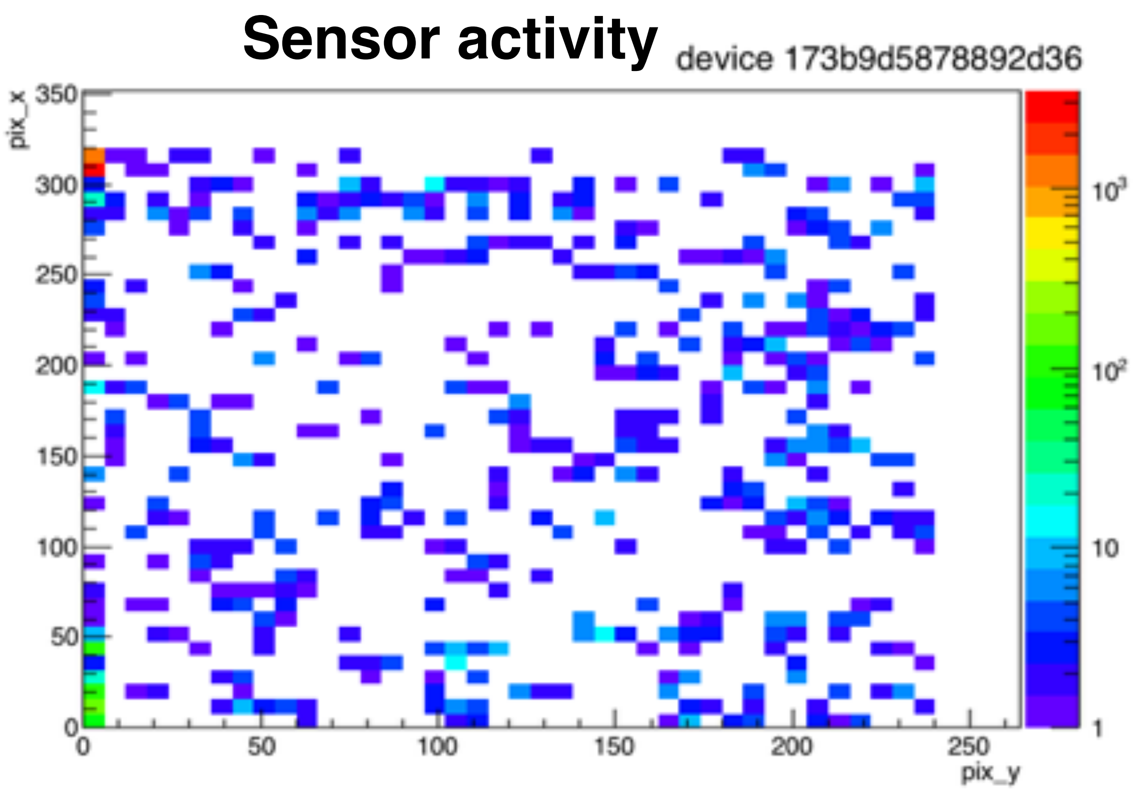
	Exposure	Events
Total	48 days, 8 hours	238,669 / 959,535
Current run	35 minutes	0 / 286

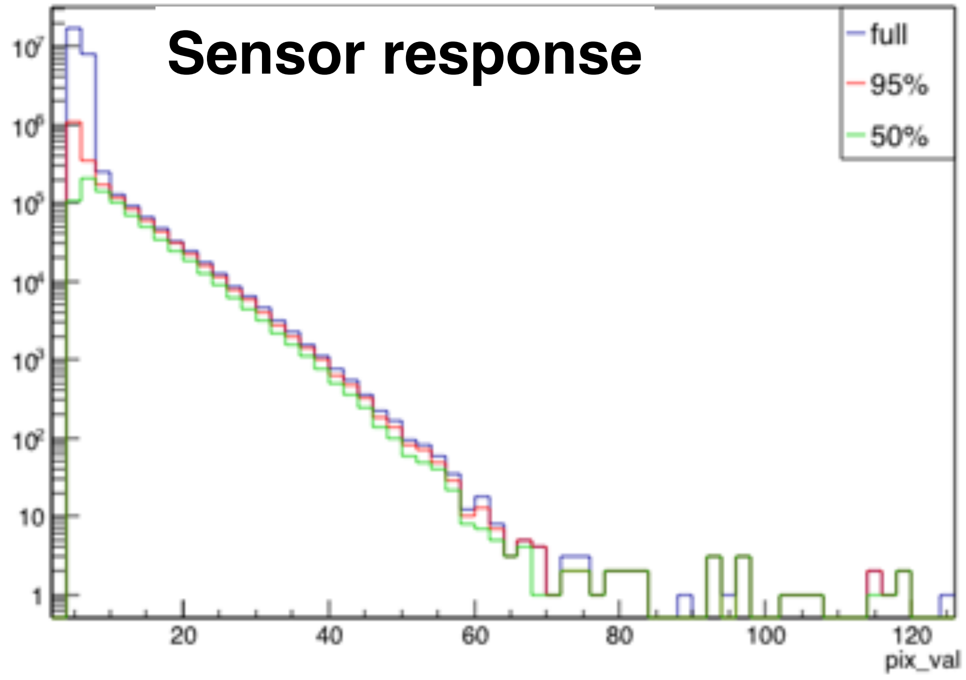
Pixel values

Counts of pixels per event

Offline Computing: Calibration

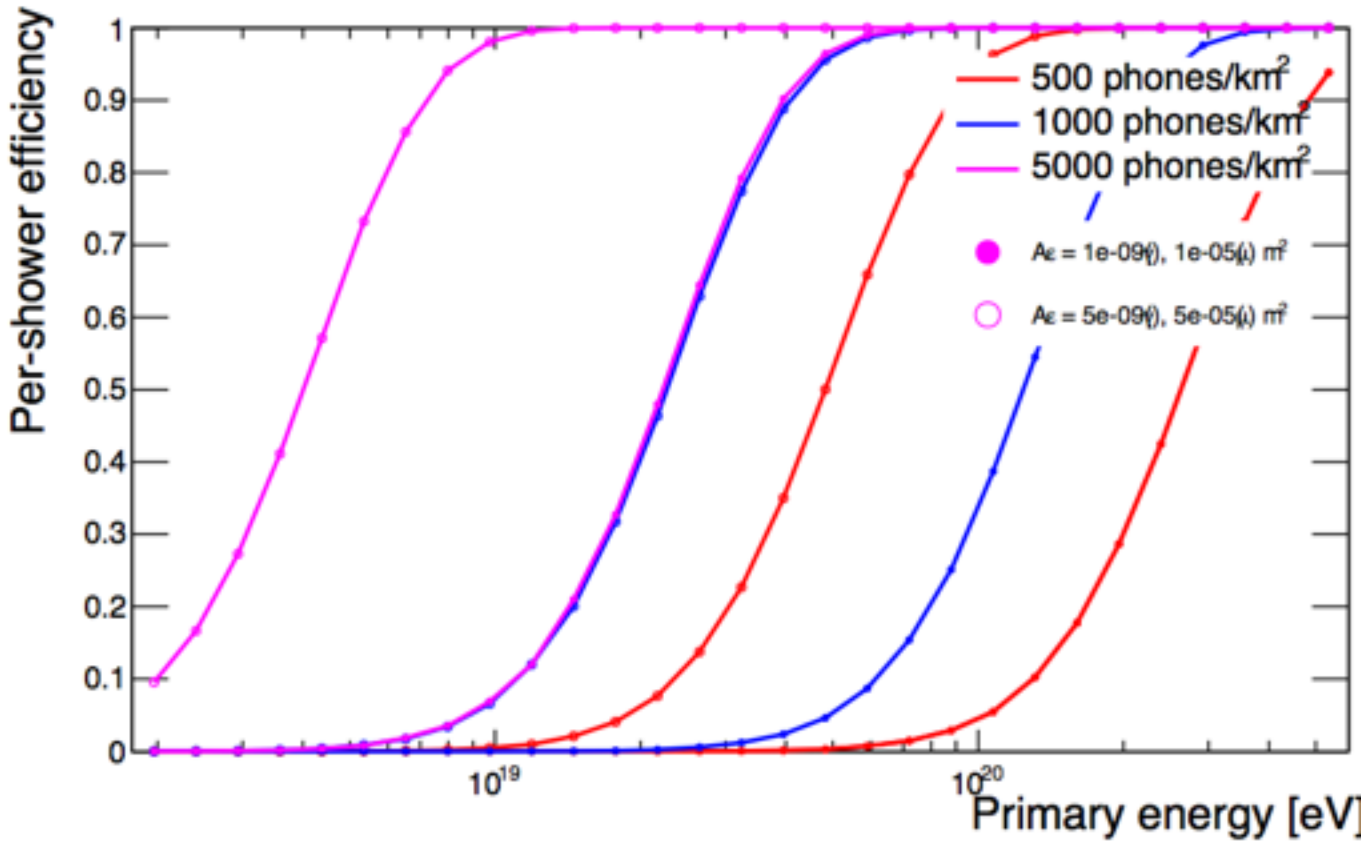
- **Challenge**: need to characterize 1M+ phones
  - Hot cell removal
  - Backgrounds (intrinsic + extrinsic)
  - Sensitivity to shower particles
- **Approach**:
  - Streaming processing whenever possible
  - Save + index device metadata with **elasticsearch**


Sensor activity


Sensor response

Offline Computing: Array Acceptance

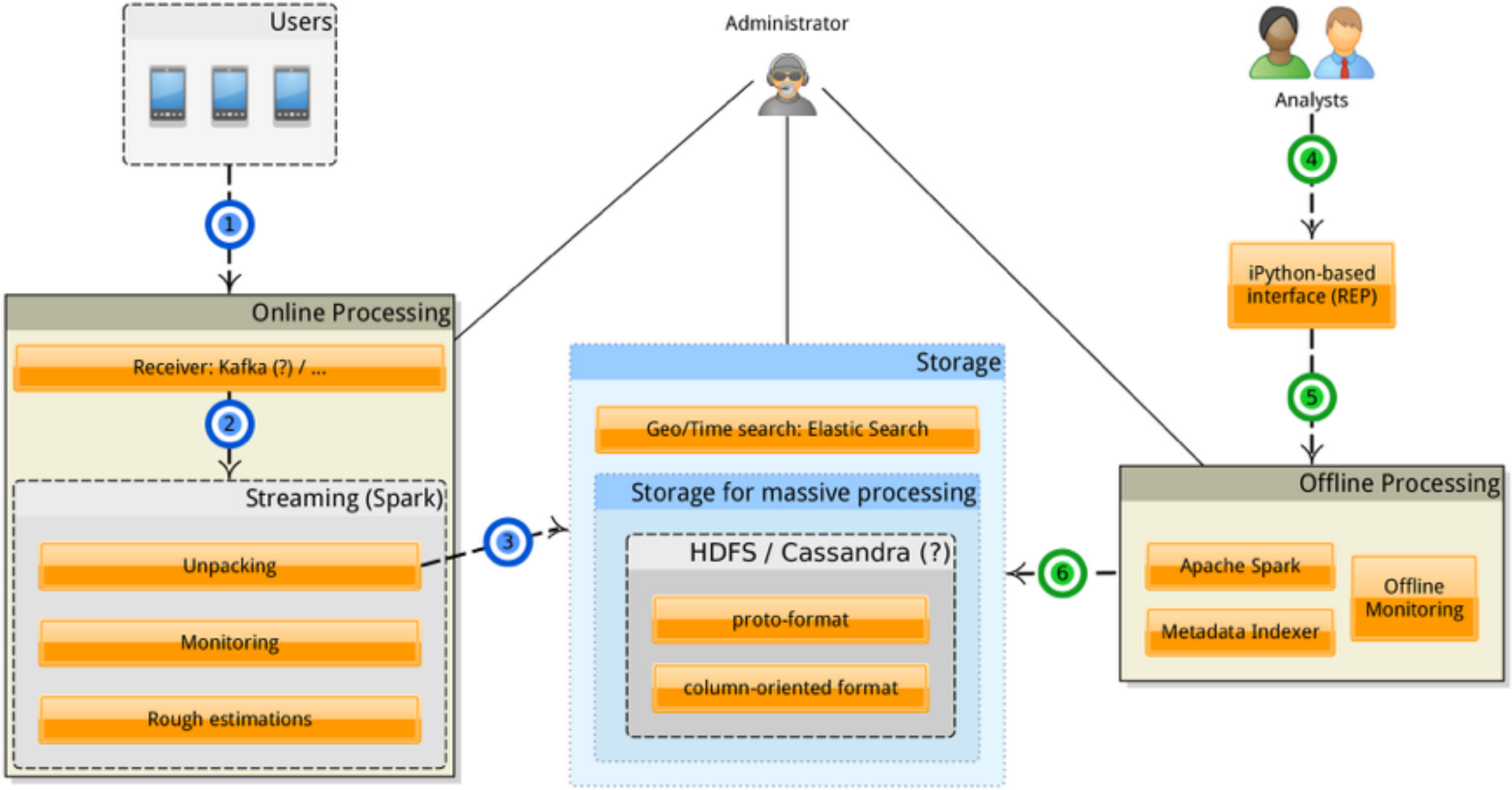
- **Challenge:** Calculate instantaneous acceptance vs. shower energy
  - Intrinsically global calculation
  - Must be able to *cluster all events* in space + time
- **Approach:**
  - Make data available on distributed, index DB (cassandra)
  - Batch processing on homogenous clusters



Offline Computing

- Major hurdles:
  - Limited/unpredictable resource availability
  - Expensive, long computations
  - Stream-process when possible
- Solution: **Apache Spark**
  - functional map-reduce driver
  - runs on anything, integration w/ Hadoop + Cassandra
  - operates in both streaming + batch mode

Offline Computing



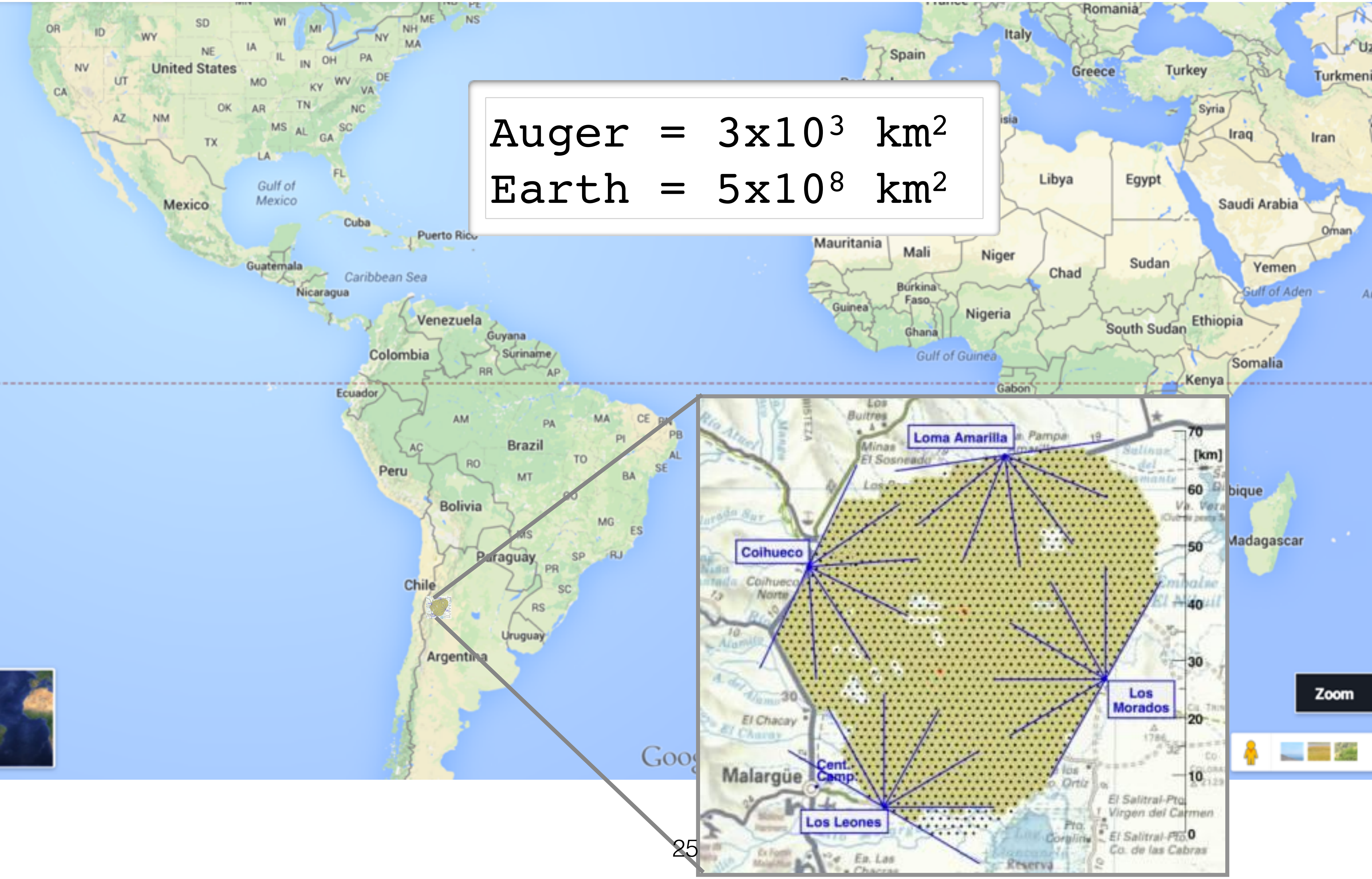
Next Steps

- Online processing is pretty stable
  - Working with our volunteers to create new features (plots, interactive items, etc)
  - Wider release mostly pending iOS/android app updates
- Offline processing: new territory
  - Developing cleaning/calibration algorithms
  - Implementing spark+Cassandra stack
  - Locating cluster resources and/or grant $$$

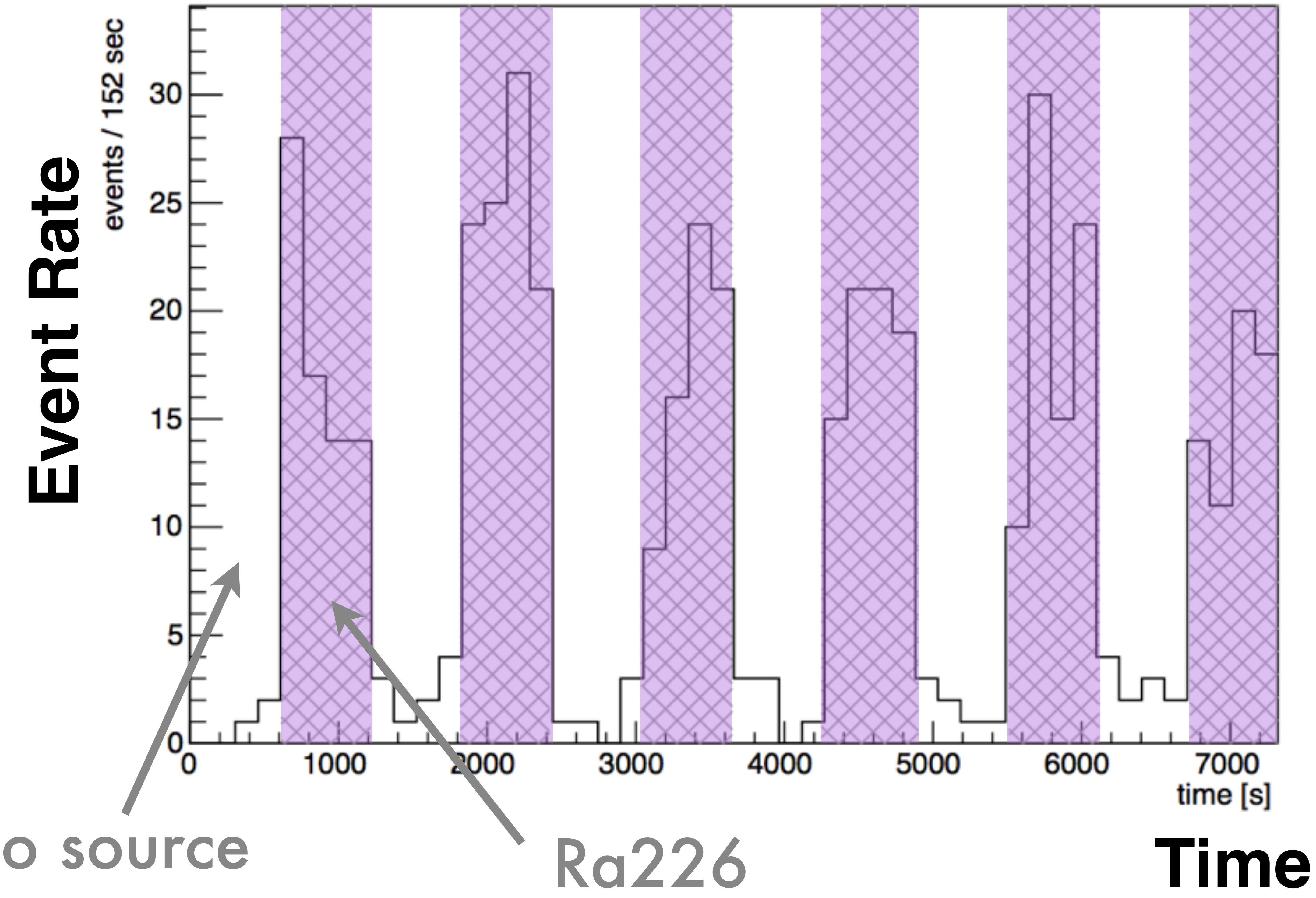
backup

Pierre Auger Observatory

Auger = $3\text{x}10^{3}$ km^{2}
Earth = $5\text{x}10^{8}$ km^{2}



Photon Sensitivity



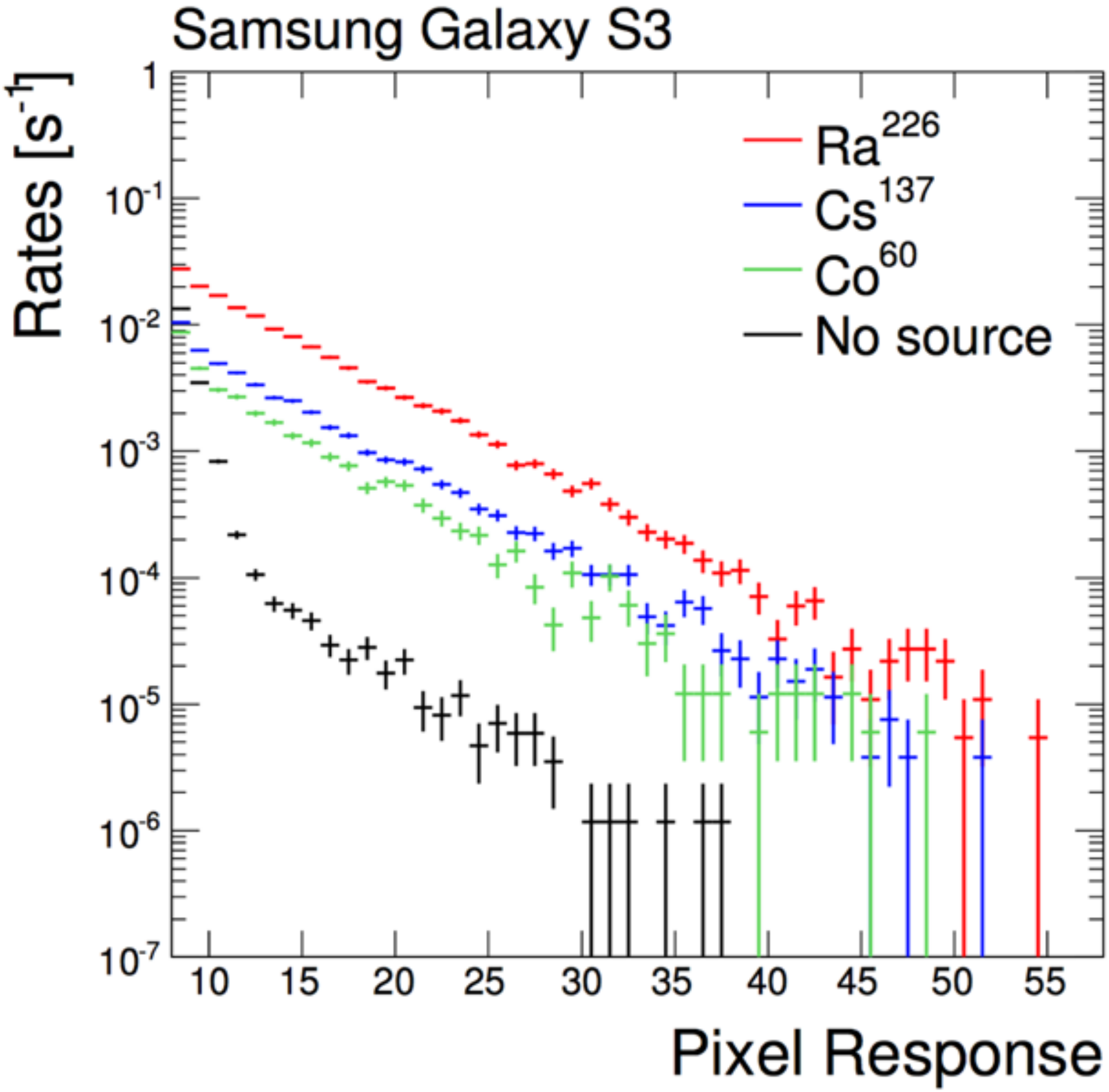
Event Rate

No source

Ra226

Time

Photon Sensitivity



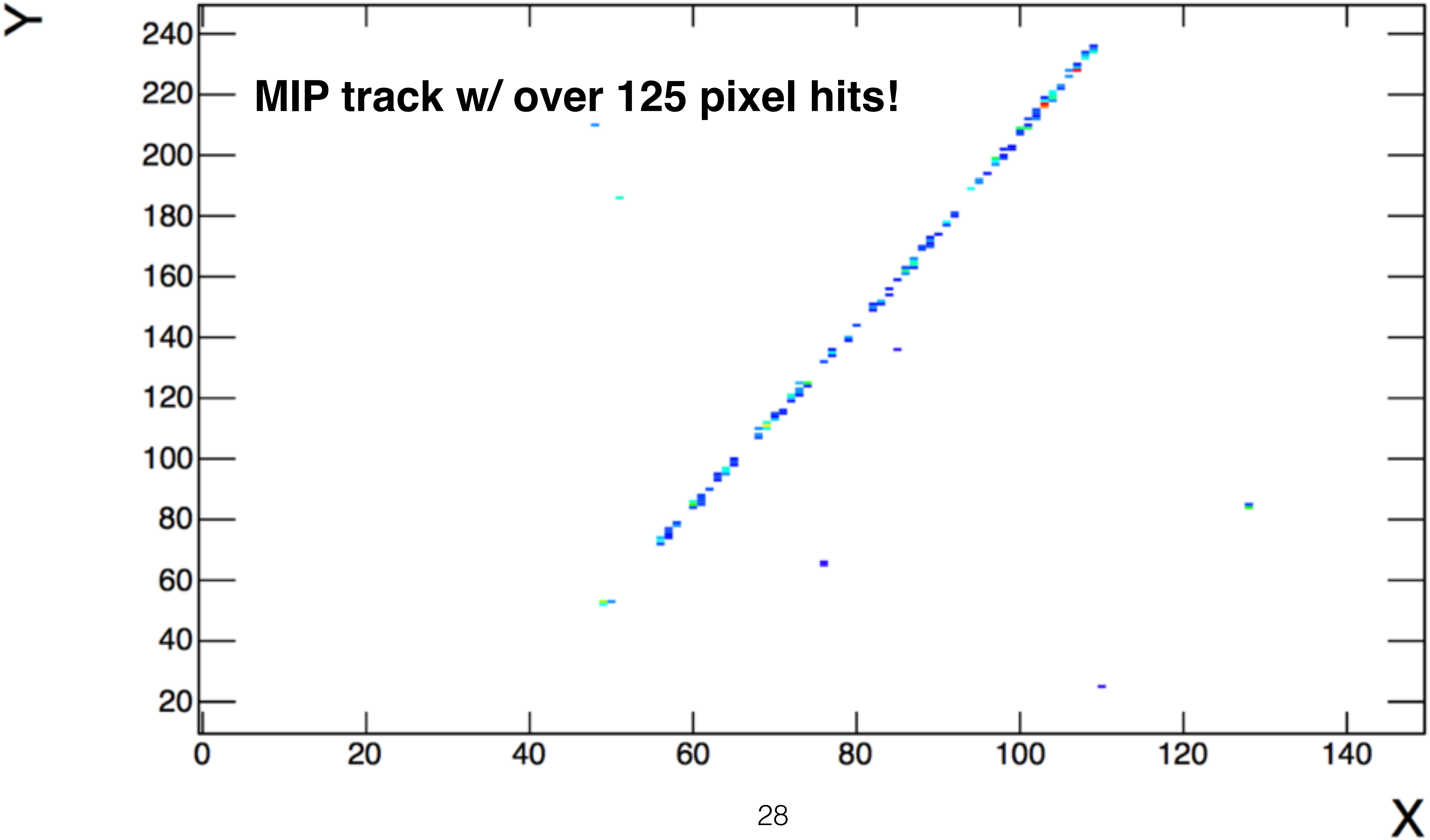
Sources w/ varying activity, energy:

```
Ra226: ~180-600 keV
Cs137:      700 keV
Co60:   1.1/1.3 MeV
```

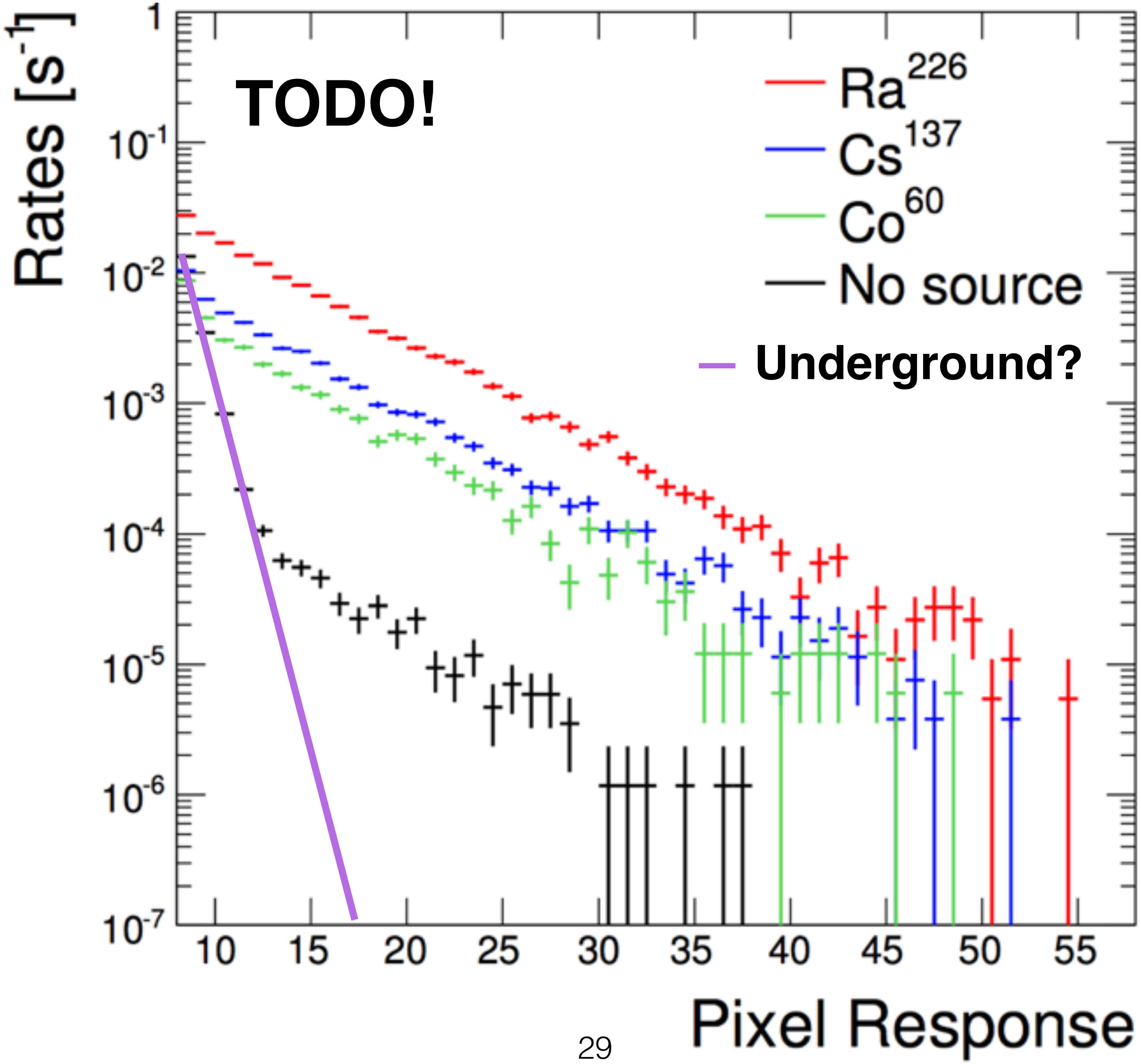
Muon Sensitivity

Get them for free from the sky!

1 muon/cm^2/min ⟹ 1 muon every 4 mins



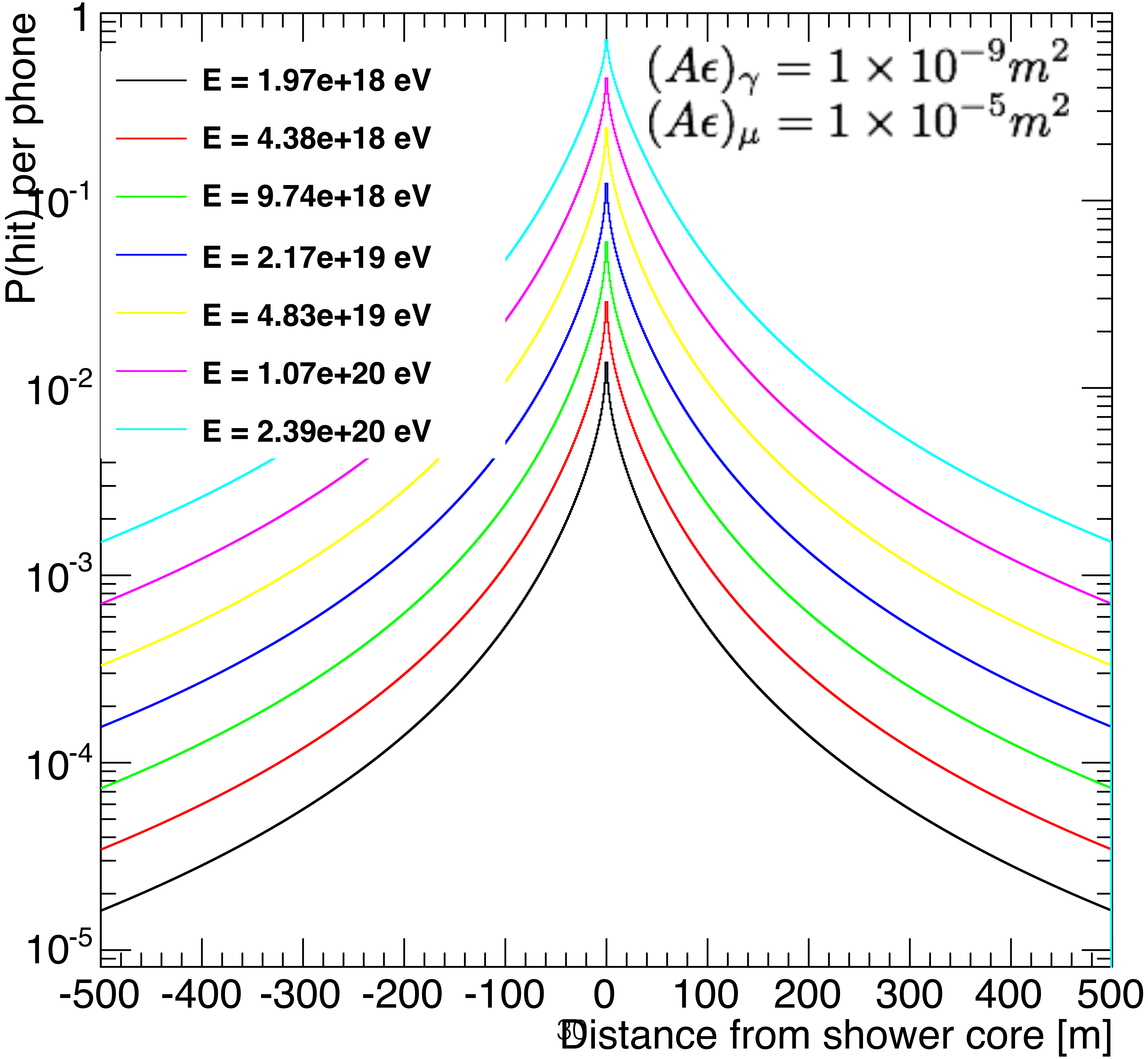
Muon Sensitivity



TODO!

Underground?

Probability of Hit



Technologies

AWS / EC2 / ECS

Pros

- Allows us to scale rapidly according to demand
- Easy to obtain world-wide service coverage
- No capital/up-front costs (pay-as-you-go)
- Possible to pay for “reserved” instances to reduce costs

Cons

- Vendor lock-in can be extreme
- Steep learning curve

Technologies

Docker

Pros

- Surprisingly easy to use
- Simplifies development environment
- Deployment: everything “just works”
- Immutable instance state makes for clean application design

Cons

- Not yet widely adopted
- Significant changes between versions
- Many awesome features are “beta”
- Integrating multiple containers can be challenging

Technologies

Redis

Pros

- Fast & battle tested
- Trivially easy to use
- Multiple functions
  - distributed store
  - messaging queue
  - pub/sub

Cons

- Data must fit in memory
- Cluster support: nacent
- Schemas can become very messy
- Not good for queries/multi-indexing/relational data

Technologies

Elasticsearch

Pros

- Stores & indexes anything
- Extremely powerful query system
- Great for live analytics
- Support for scripting
- Designed for clusters

Cons

- Fairly new technology
- Unclear how well it can scale
- Query DSL is awkward (but powerful!)