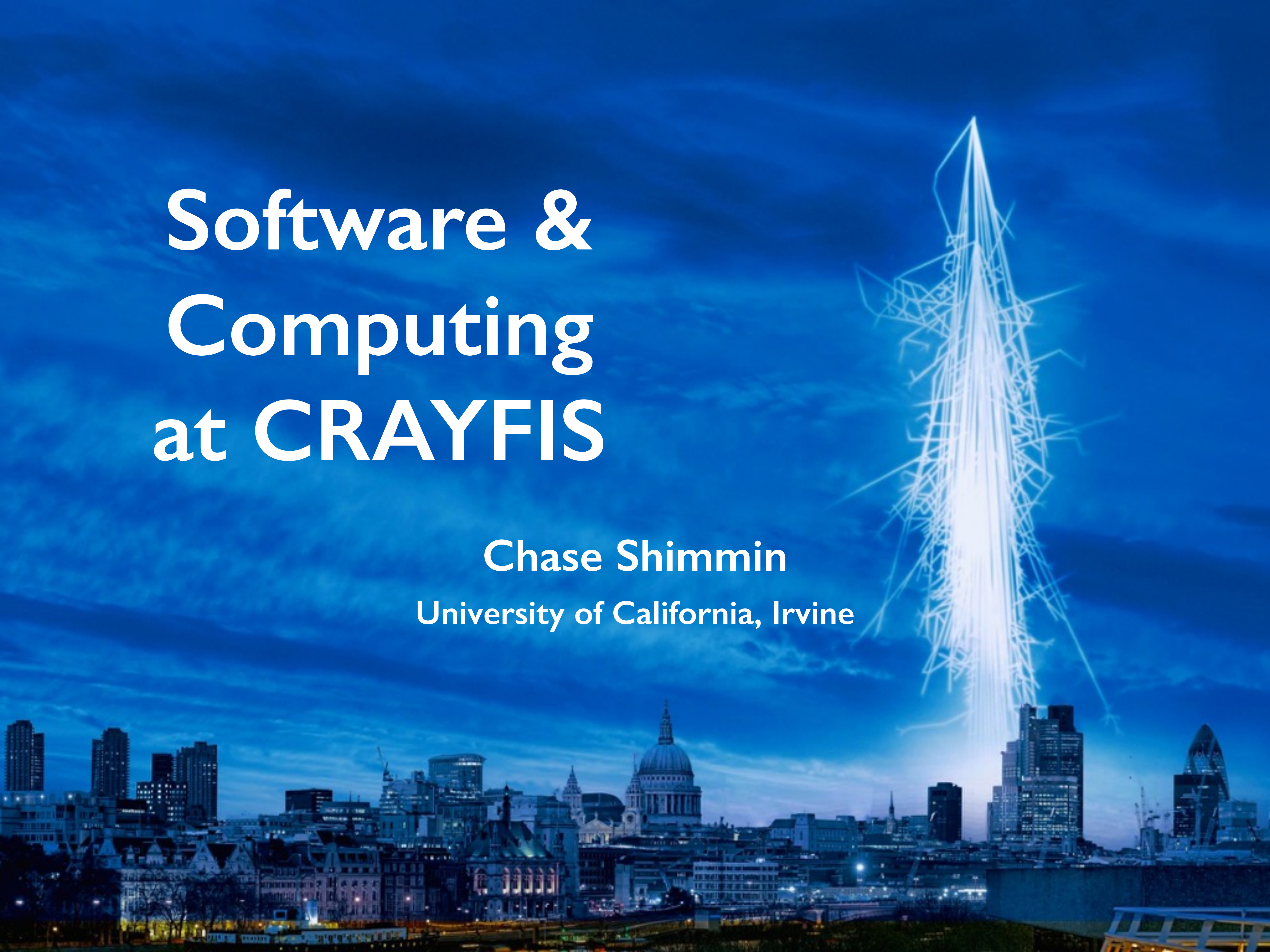
Software & Computing at CRAYFIS

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

UC DAVIS
UNIVERSITY OF CALIFORNIA

Mulhearn
Burns
Buonacarsi

NYU

Cranmer

Яндекс
Yandex

Ustyuzhanin
+2 masters st.

NAOC

Deng

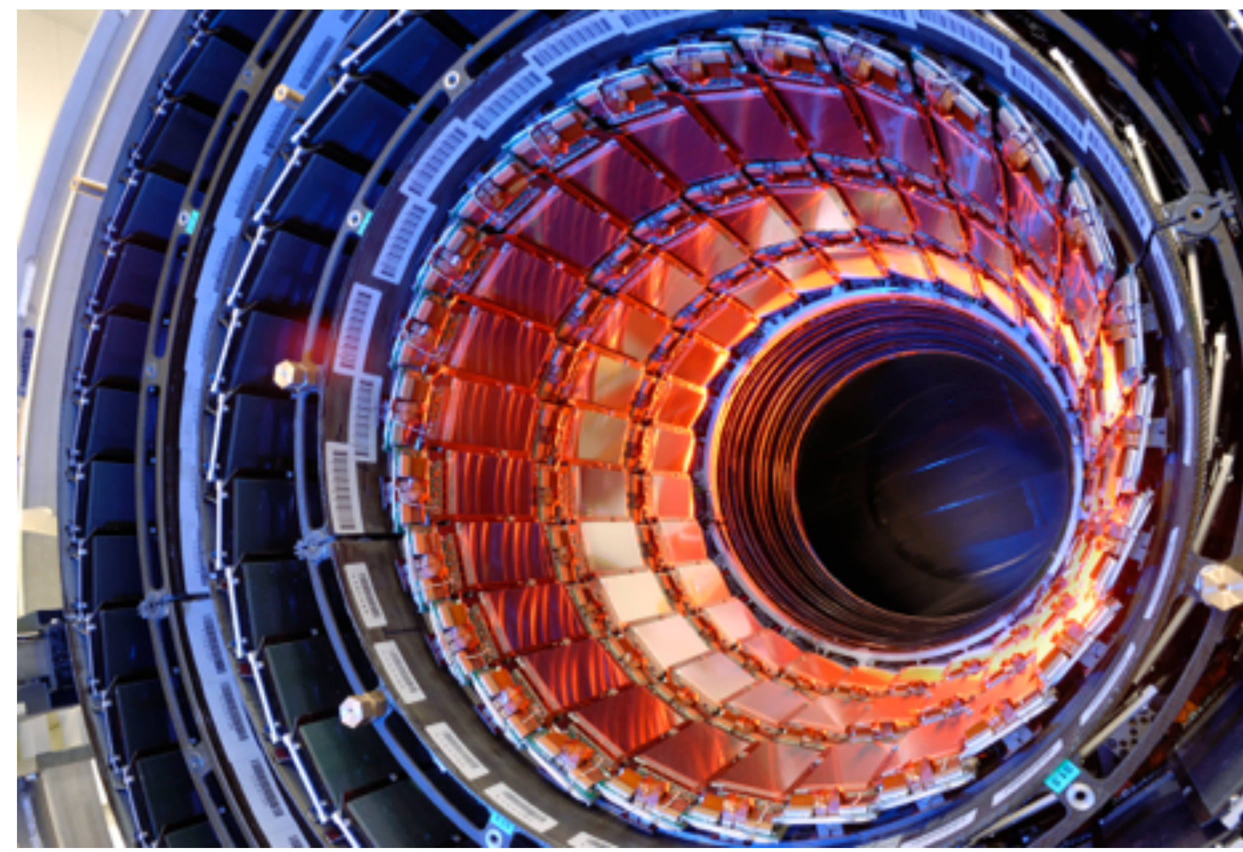
powered by amazon web services

Smartphones are:
(tiny) Particle Detectors

Camera Sensor

(Active area: ~0.3 cm^2)

=



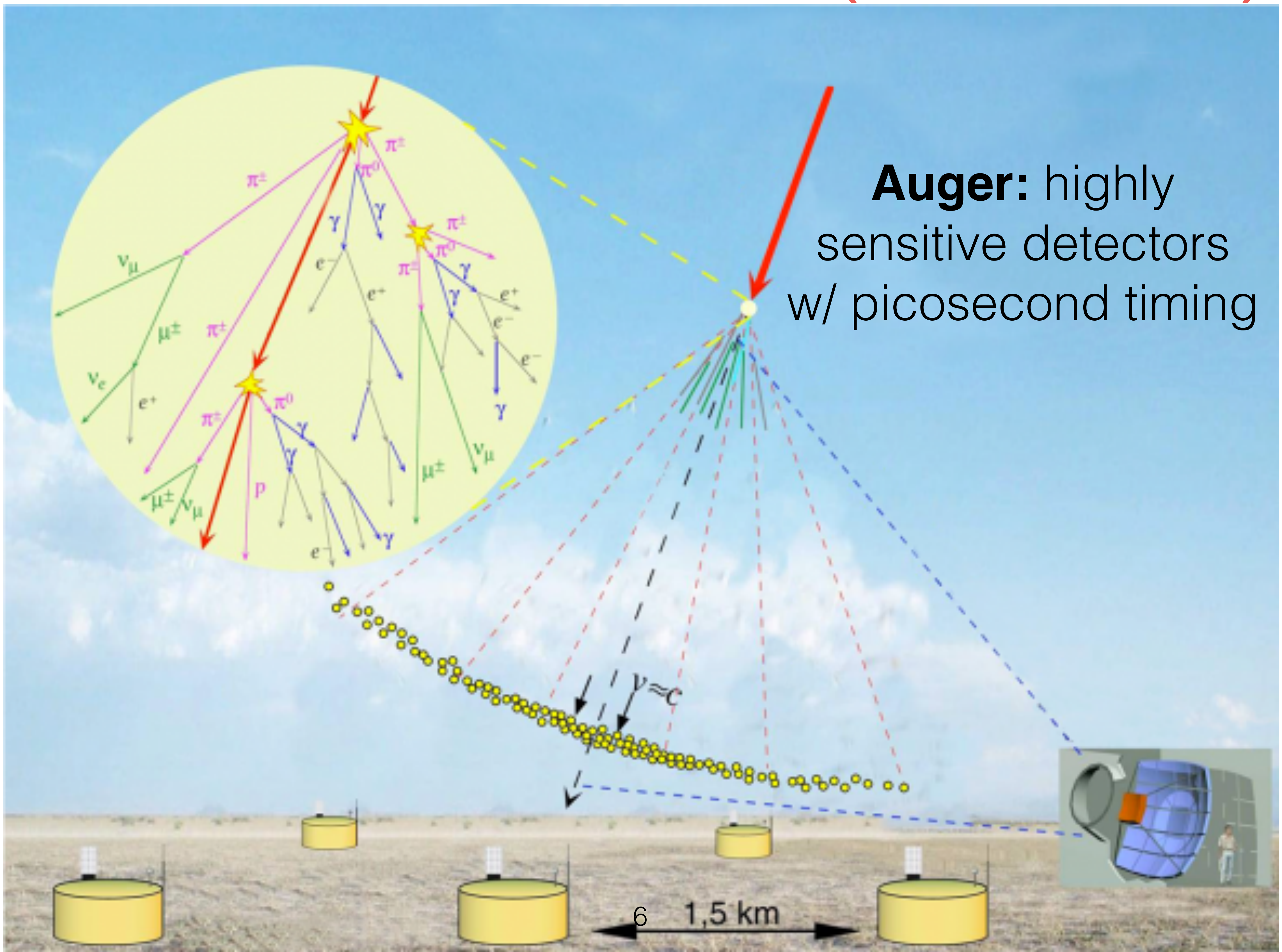
Smartphones are:
Mobile Laboratories

GPS

Wi-Fi

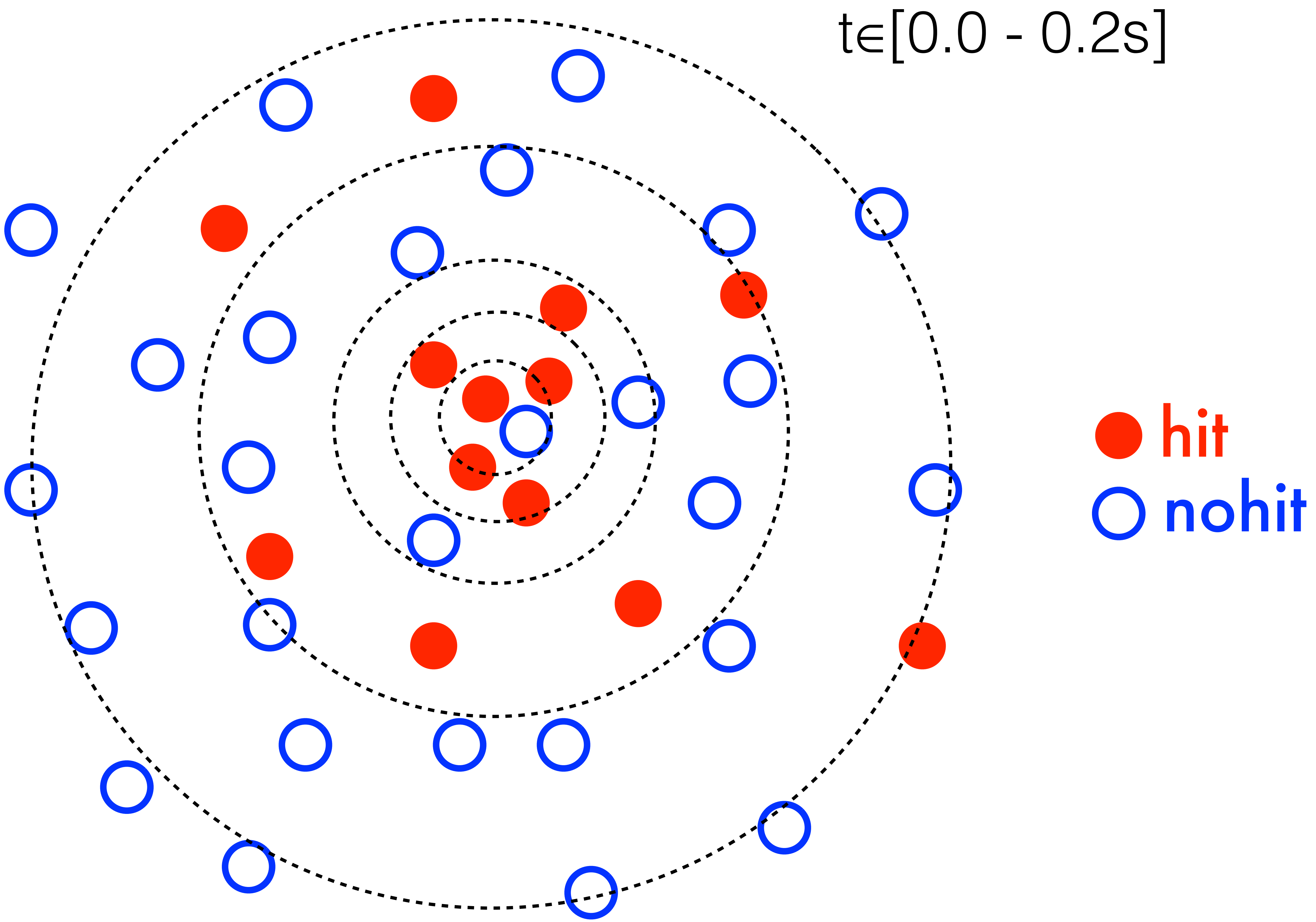
Shower Reconstruction

(State of the Art)



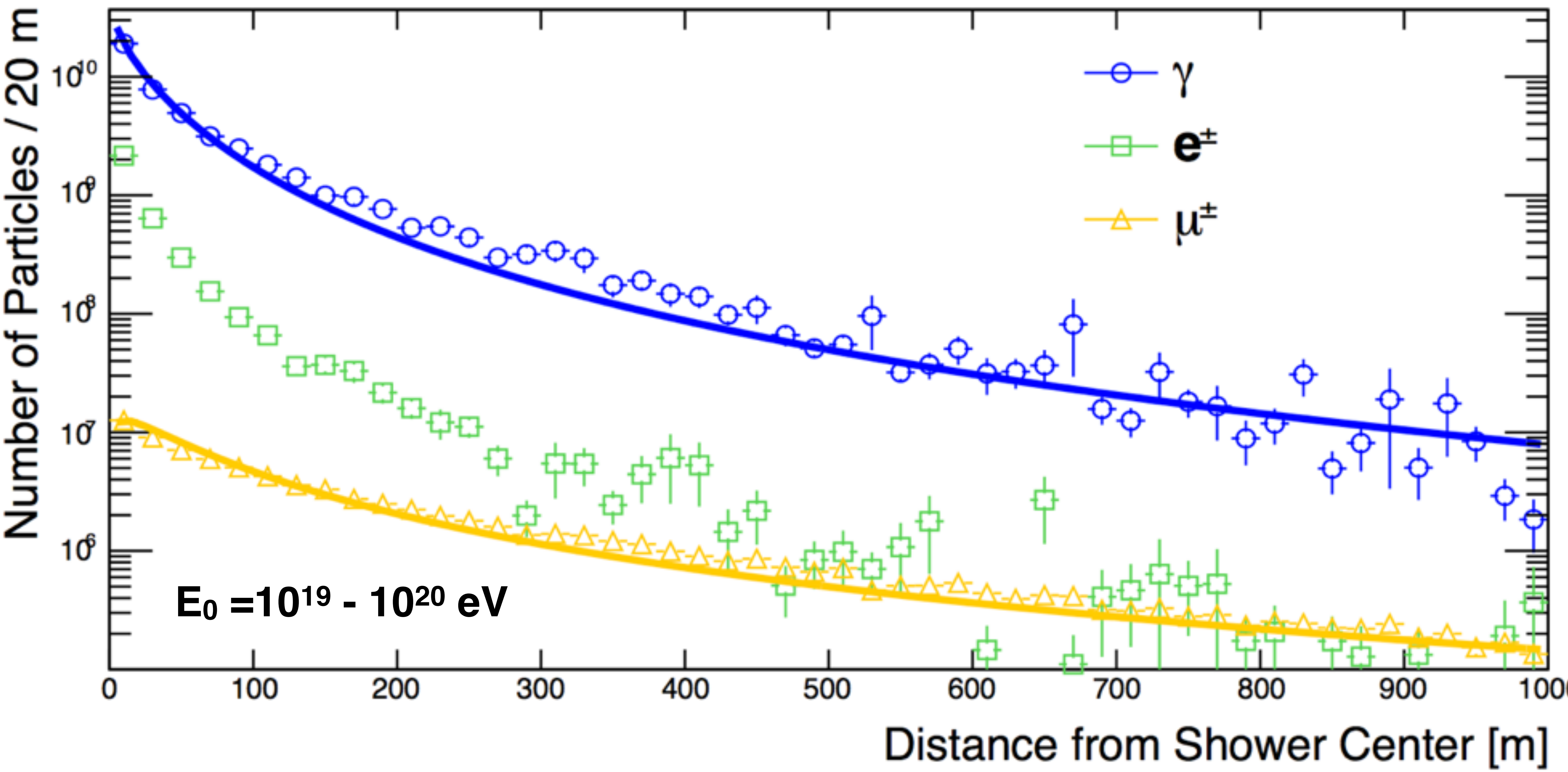
Shower Reconstruction

(w/ CRAYFIS)



Particle Content

Tremendous densities
near shower core

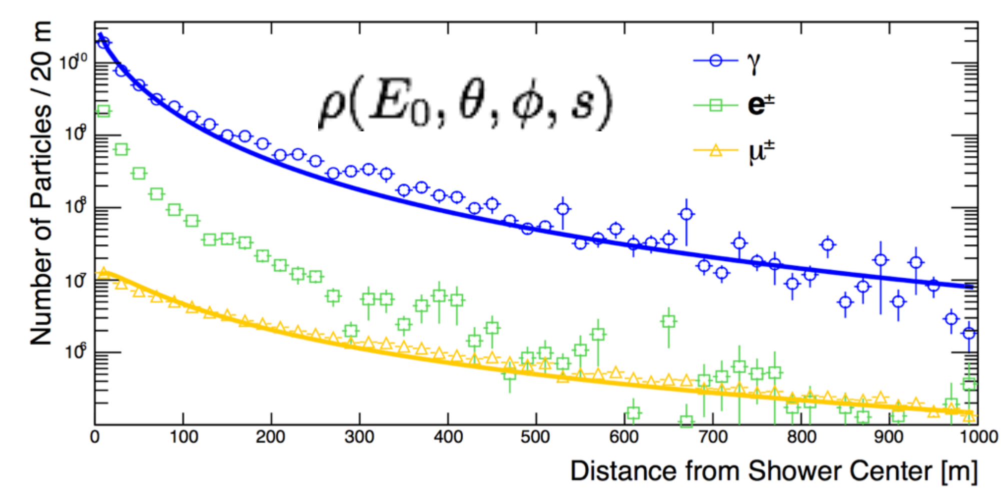


Likelihood

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



Probability of seeing nothing:

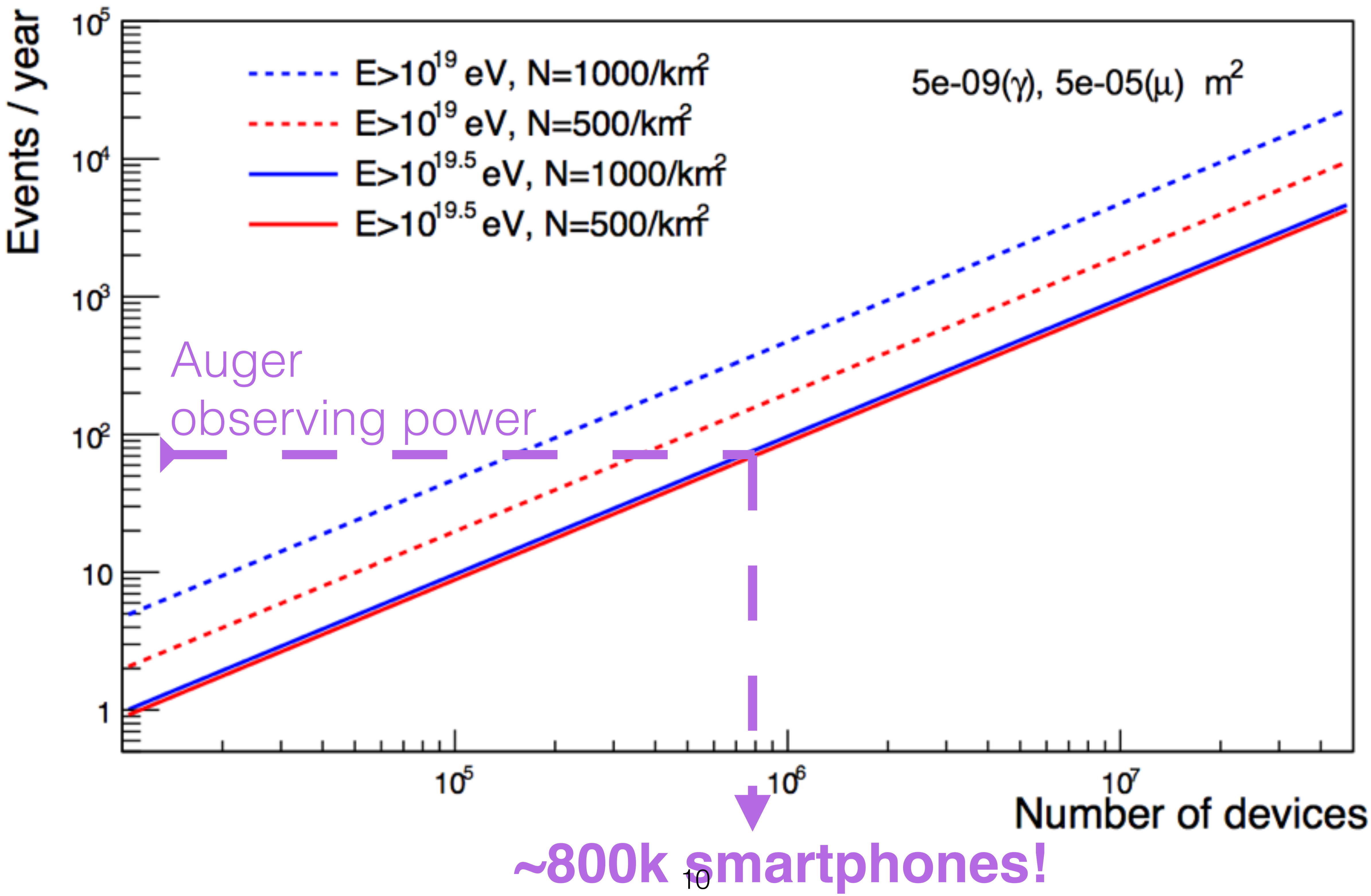
$$P_0(x, y) = e^{-\lambda}$$

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

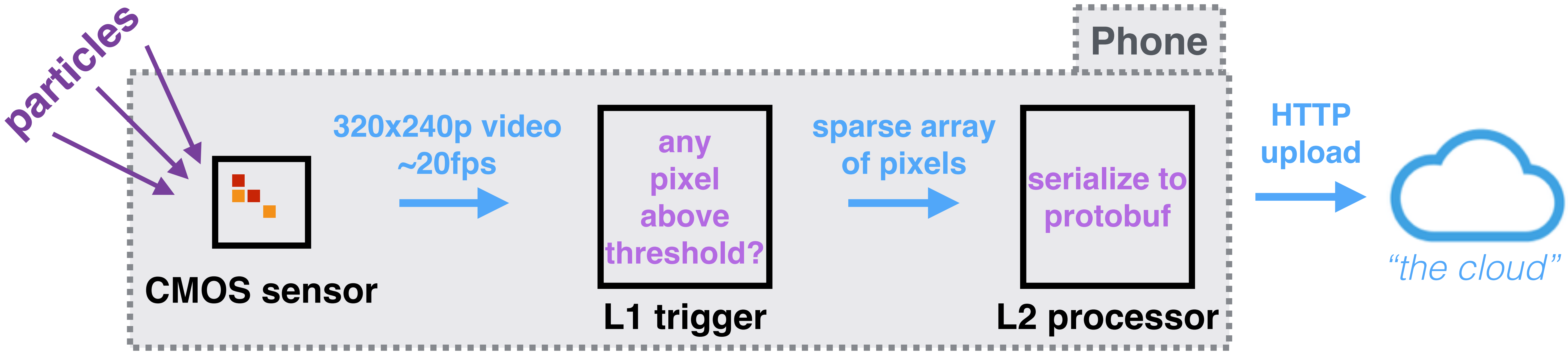
$$L(E_0, \theta, \phi, s) = \prod_i P_0(x_i, y_i) \prod_j (1 - P_0(x_j, y_j))$$

no hit — hit

Keeping up with Auger



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

(DAQ)

Online Computing

CRAYFIS DAQ load:

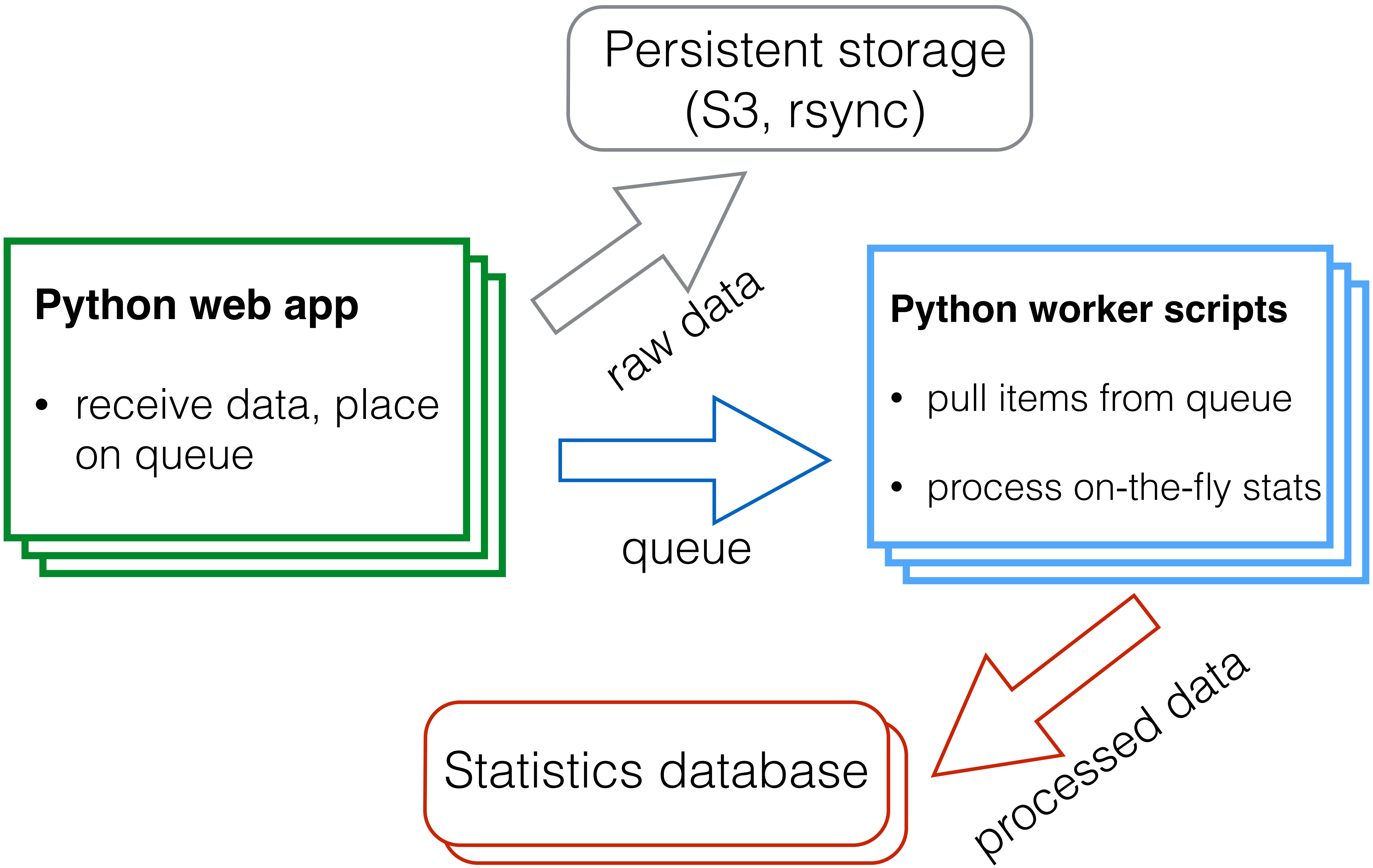
(1M phones) x $(3\text{-}5\ \text{min})^{-1}$ ~ 3-6 kHz

Google searches: 40 kHz

Reddit pageviews: <100 Hz

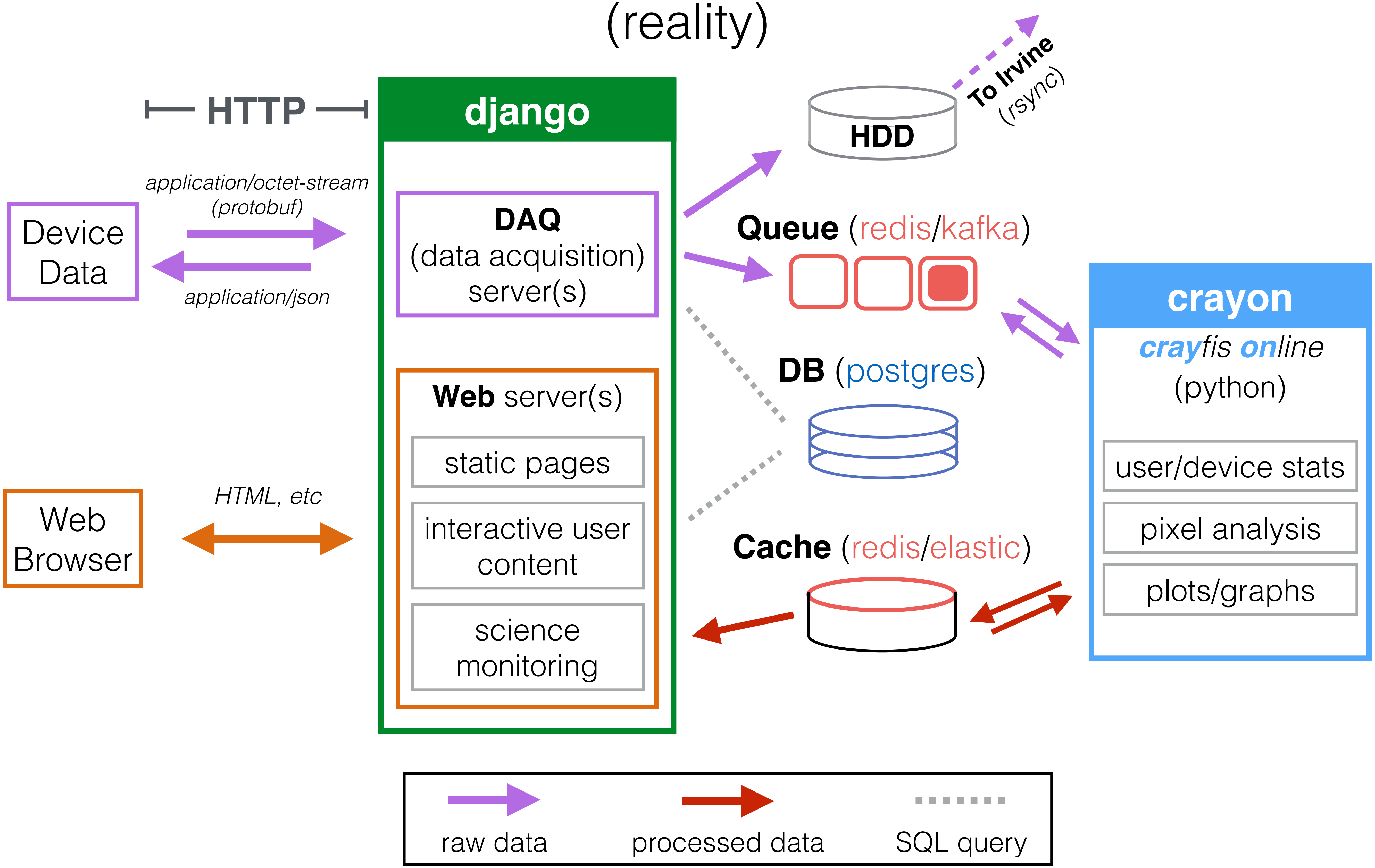
- Phones generate datablocks every ~3-5 min
- Each one needs to be received, saved, and processed by us!

Online Backend
(simplified)



Online Backend

(reality)



Online Frontend

Website: http://crayfis.io

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

Network Map

National Ranking

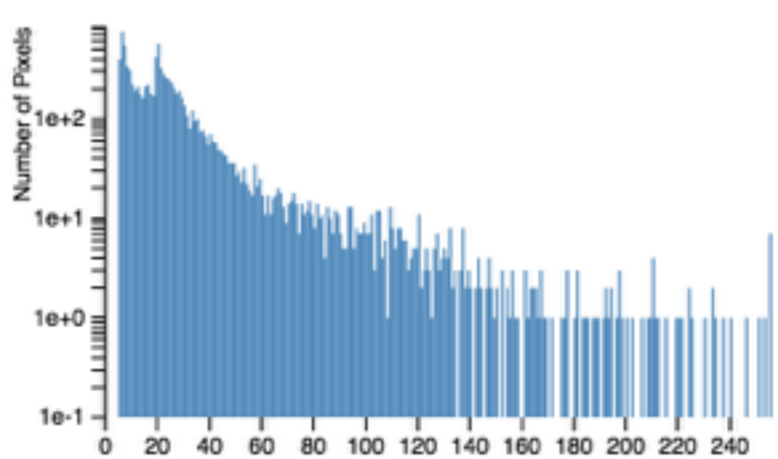
Rank	Country	Score
1	USA	187,760,499
2	NLD	37,744,624
3	GBR	34,409,785
4	BEL	24,051,870
5	CHN	21,498,151
6	ESP	5,286,500
7	FRA	5,026,540
8	DEU	3,909,537

CRAYFIS Cosmic RAYs Found In Smartphones Project Login

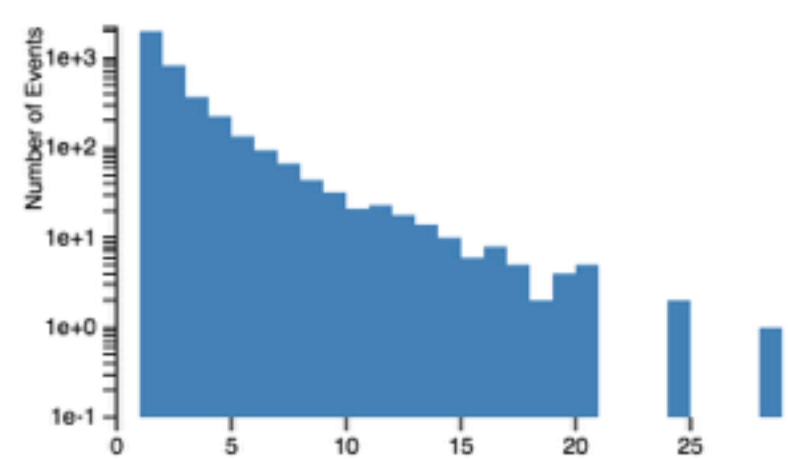
PEACE 4 SPACE

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

Pixel values



Counts of pixels per event

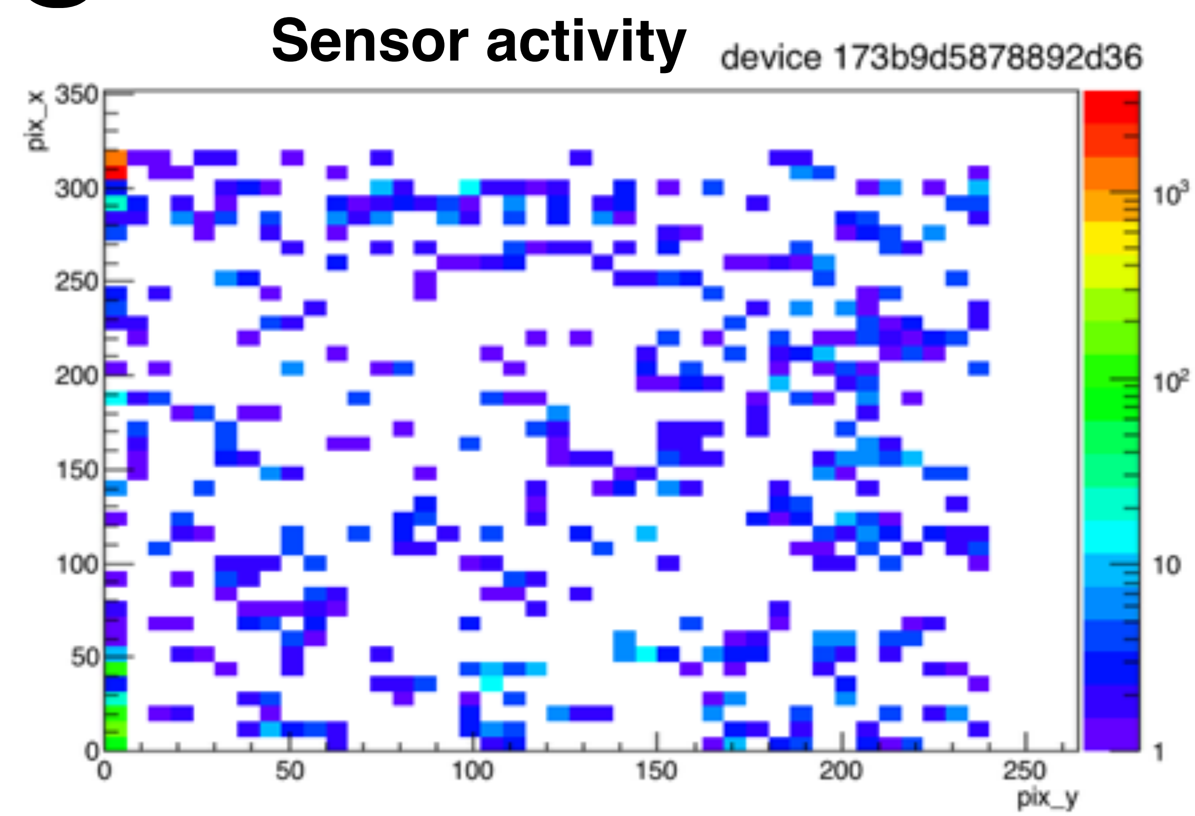


Offline Computing

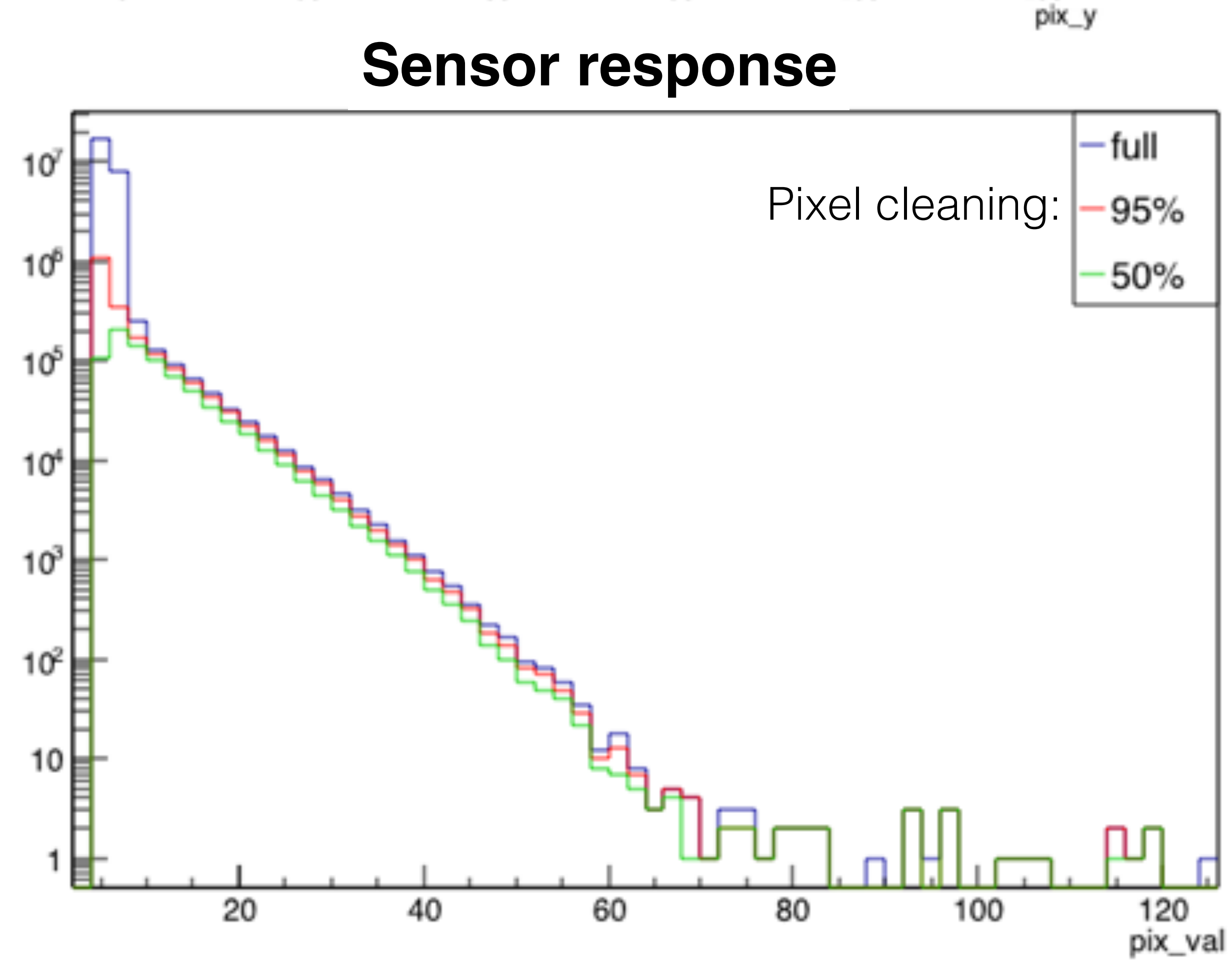
(analysis)

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**
    - Can also index phones based on activity / location



Offline Computing: Array Exposure

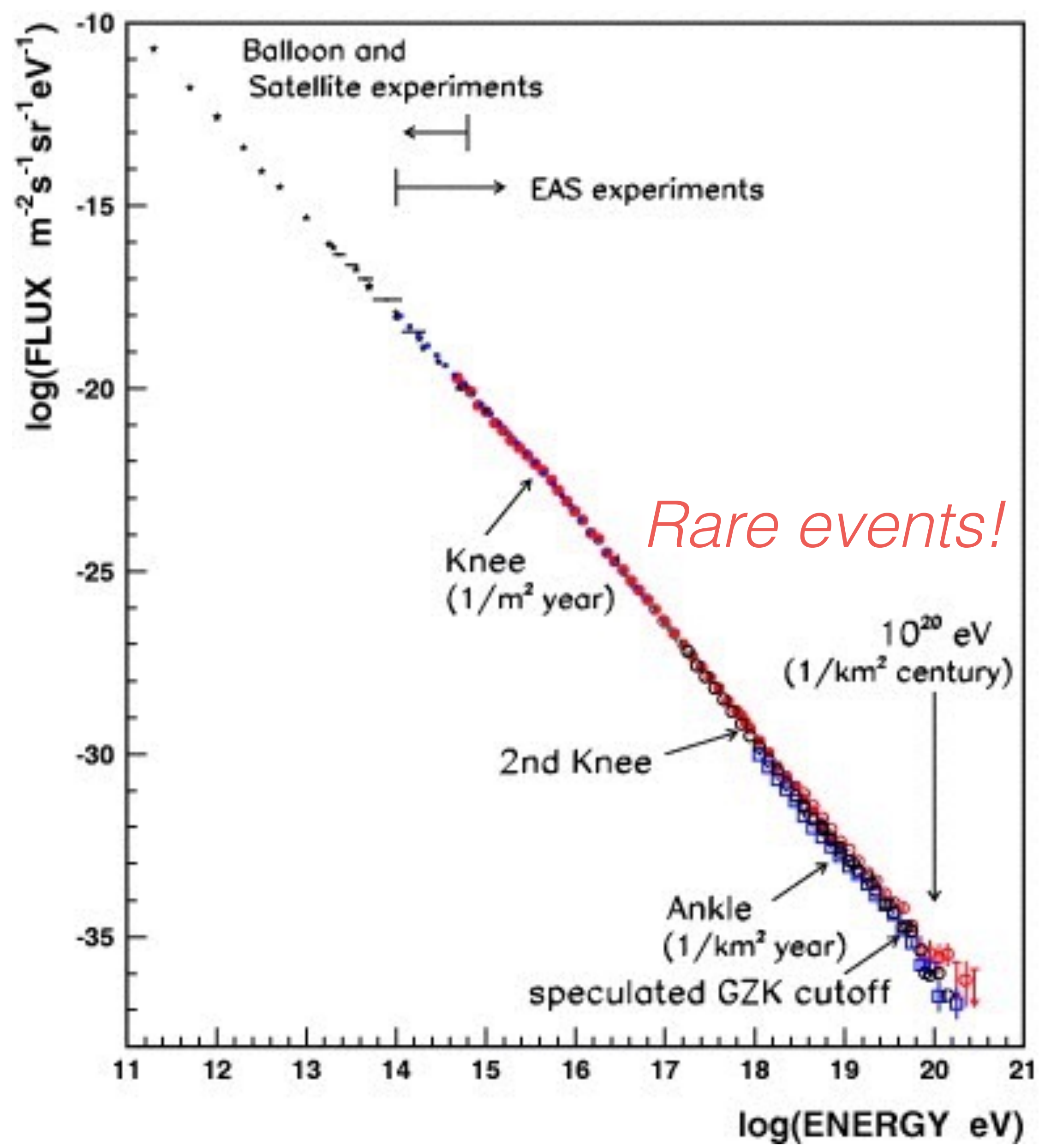
LHC:Luminosity :: UHECRs:Exposure

Exposure function:

(Detection efficiency) x (array coverage)

$$\mathcal{E}(E) = \int_T \int_\Omega \int_{S_{\rm gen}} \varepsilon(E, t, \theta, \phi, x, y) \cos\theta \, dS \, d\Omega \, dt,$$

where $d\Omega = \sin\theta d\theta d\phi$ and Ω are respectively the differential and total solid angles, θ and ϕ are the zenith and azimuth angles and $dS = dx \times dy$ is the horizontal surface element.



Auger efficiency: ~100% (above 10^{18} eV)
Auger size: $3\text{x}10^3$ km^2

But: can't get better **or** bigger!

Offline Computing: Array Exposure

Basic approach:

- Simulate detector response to many showers vs. energy, angle, etc.
- Multiply by size of array

Challenge: Need instantaneous acceptance

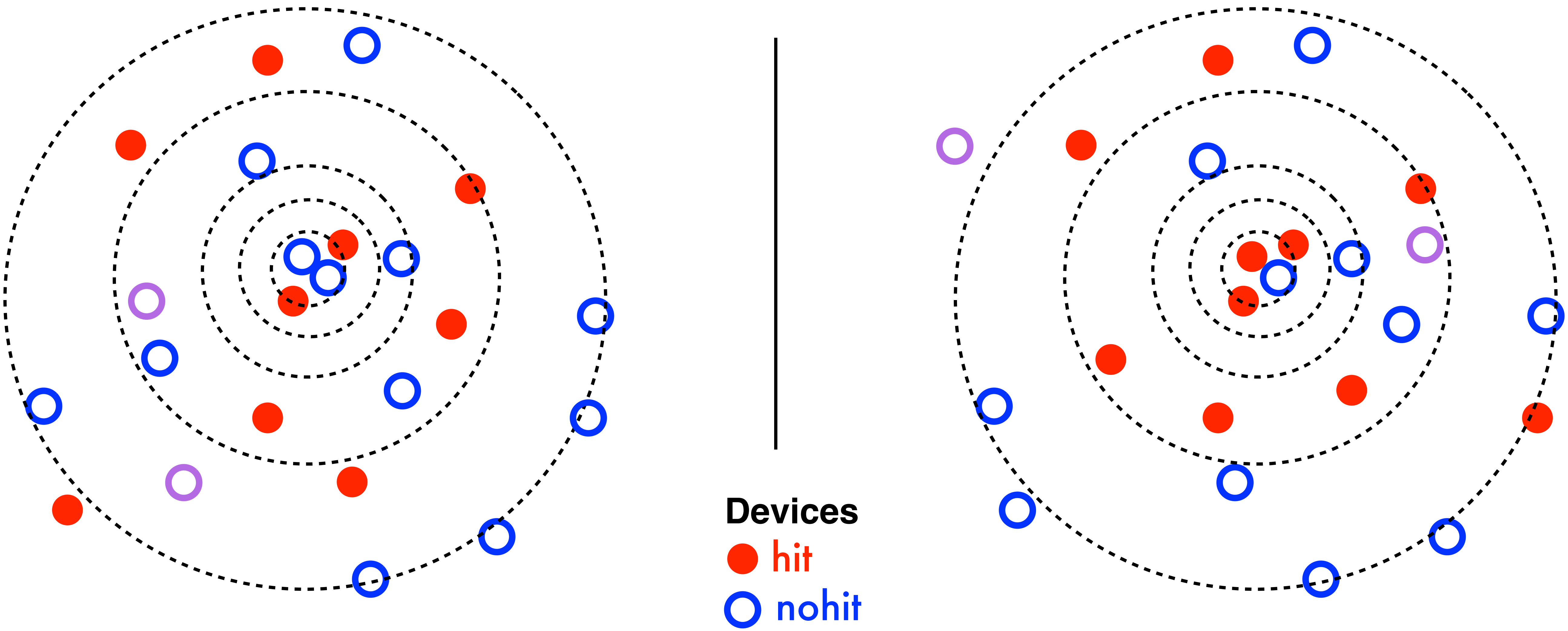
- Unlike Auger, our array changes constantly!
- Intrinsically global calculation
- Must be able to *cluster all events* in space + time

Offline Computing:
Array Acceptance

Different detector array every instant!

t∈[0.0 - 0.2s]

t∈[0.1 - 0.3s]

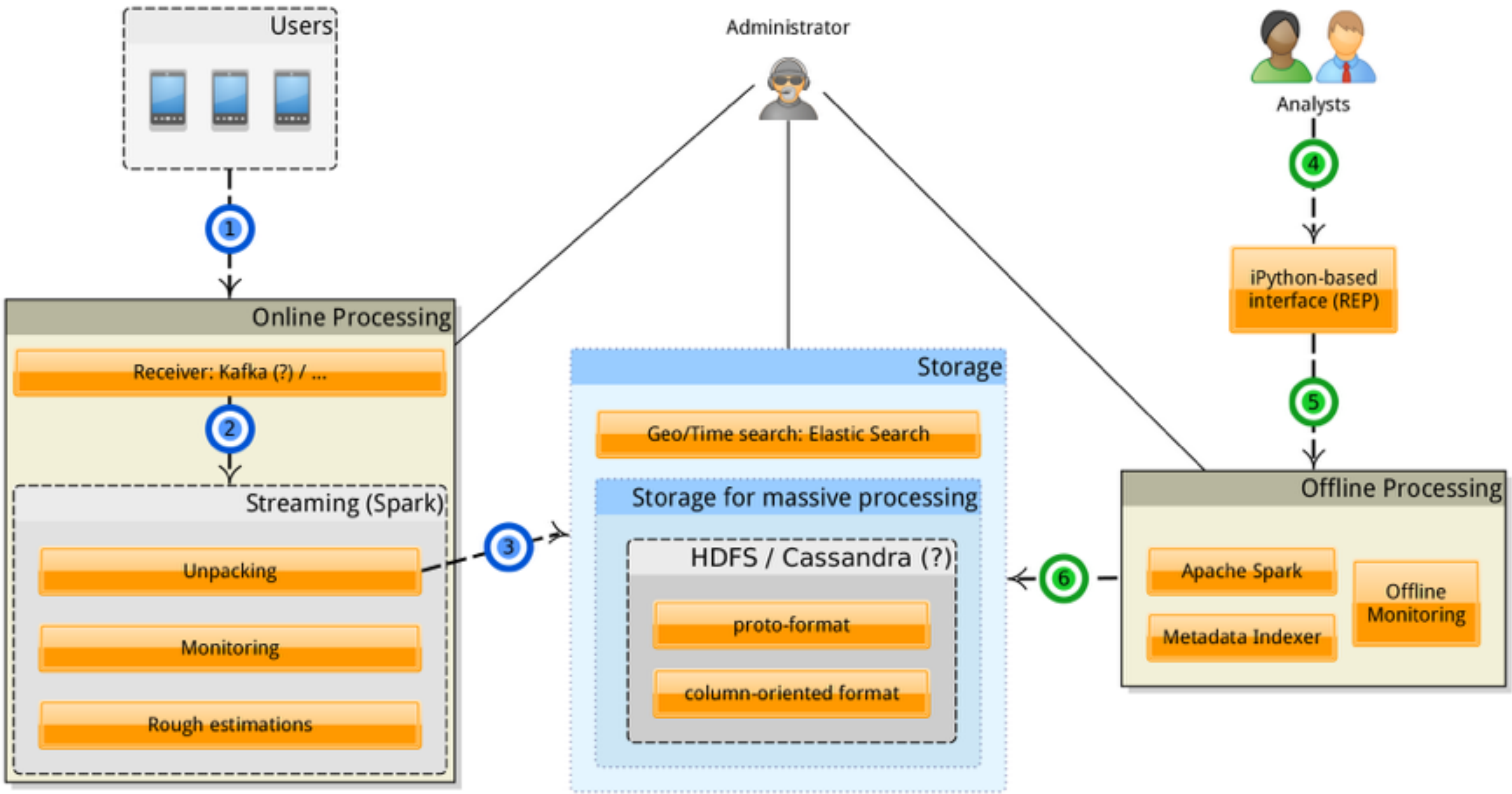


Offline Computing

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

Spark
+
cassandra

Infrastructure Overview



Next Steps

✓ **Online processing:** pretty much done!

- ➡ Working w/ volunteers to create new features (plots, interactive items, etc)

✦ **Offline processing:** new territory

- Developing cleaning/calibration algorithms
- Implementing spark+Cassandra stack
- Locating cluster resources and/or grant **$$$**

✦ **Full-scale release:**

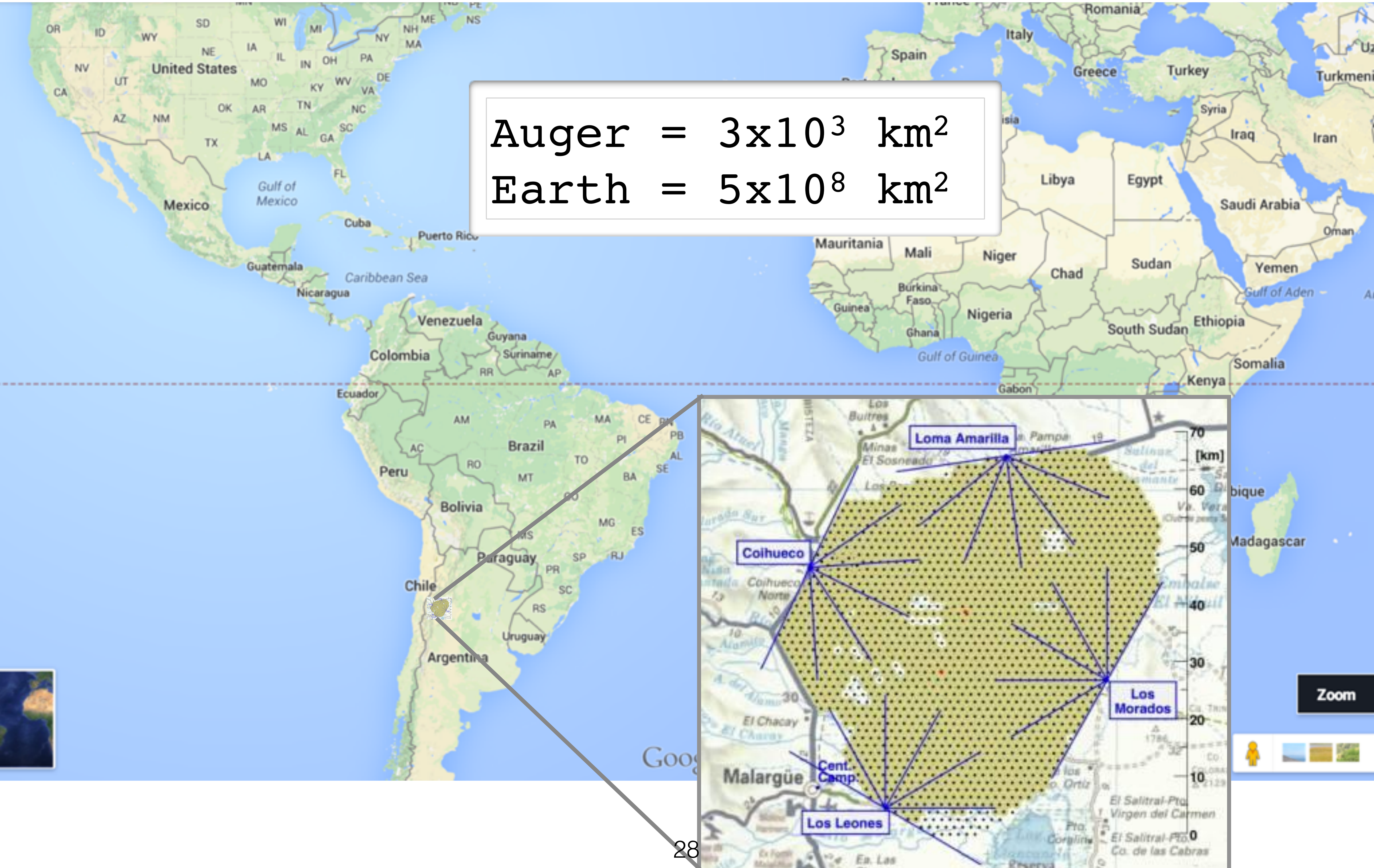
- 80k+ emails on our beta invite list!
- Pending **calibration**, iOS/android development

Thank you!

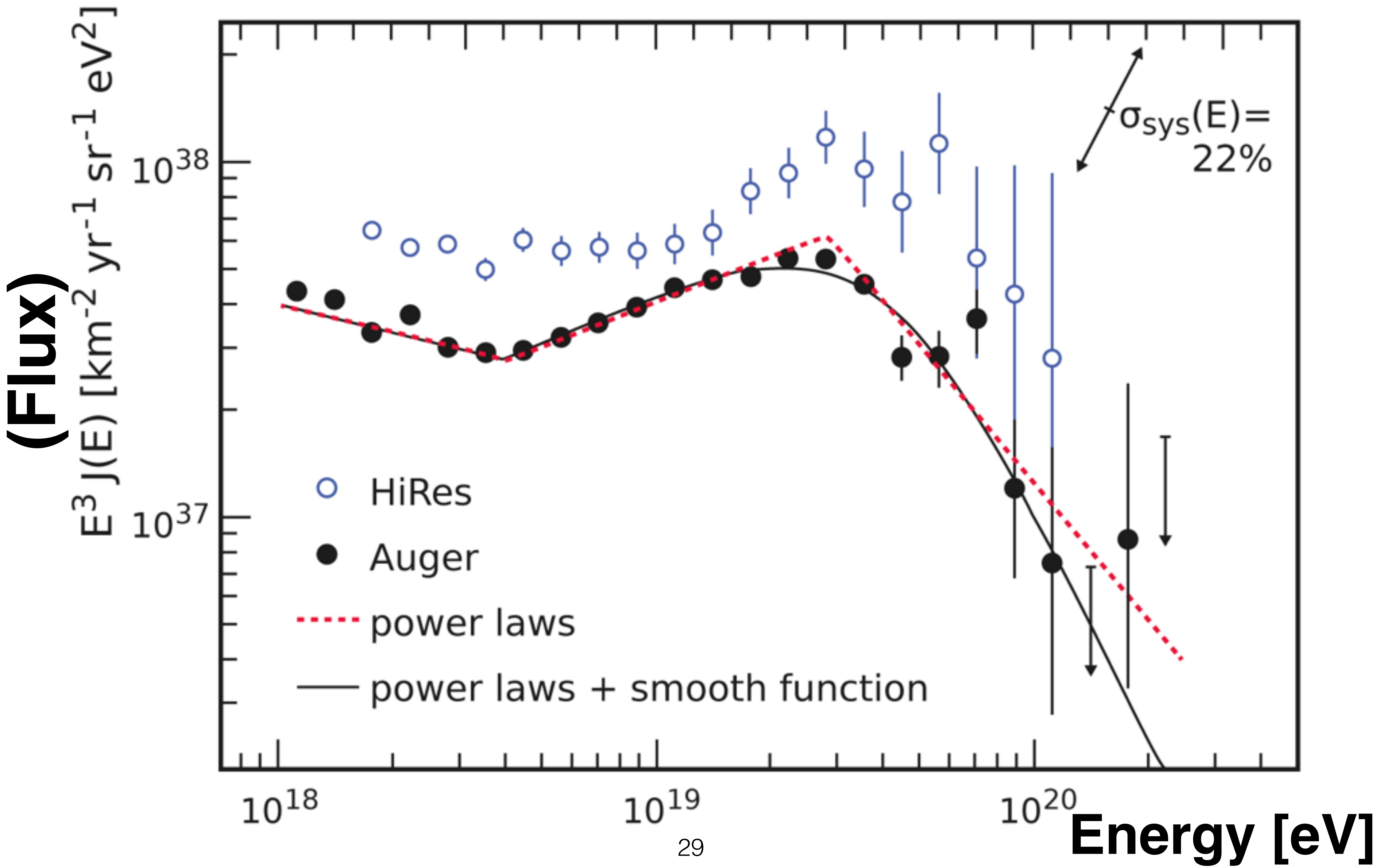
backup

Pierre Auger Observatory

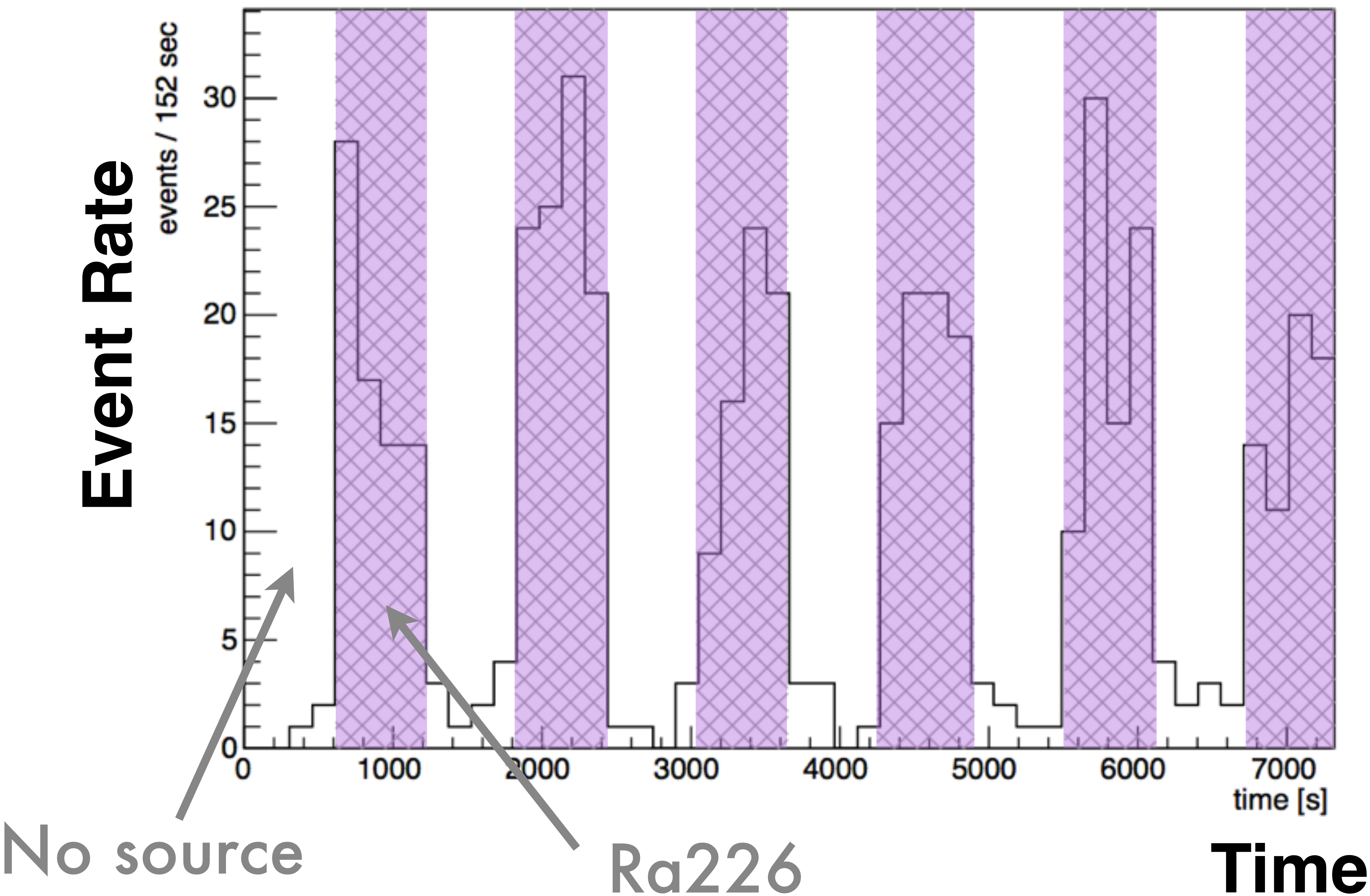
$$\text{Auger} = 3\times10^{3}\ \text{km}^2$$
$$\text{Earth} = 5\times10^{8}\ \text{km}^2$$



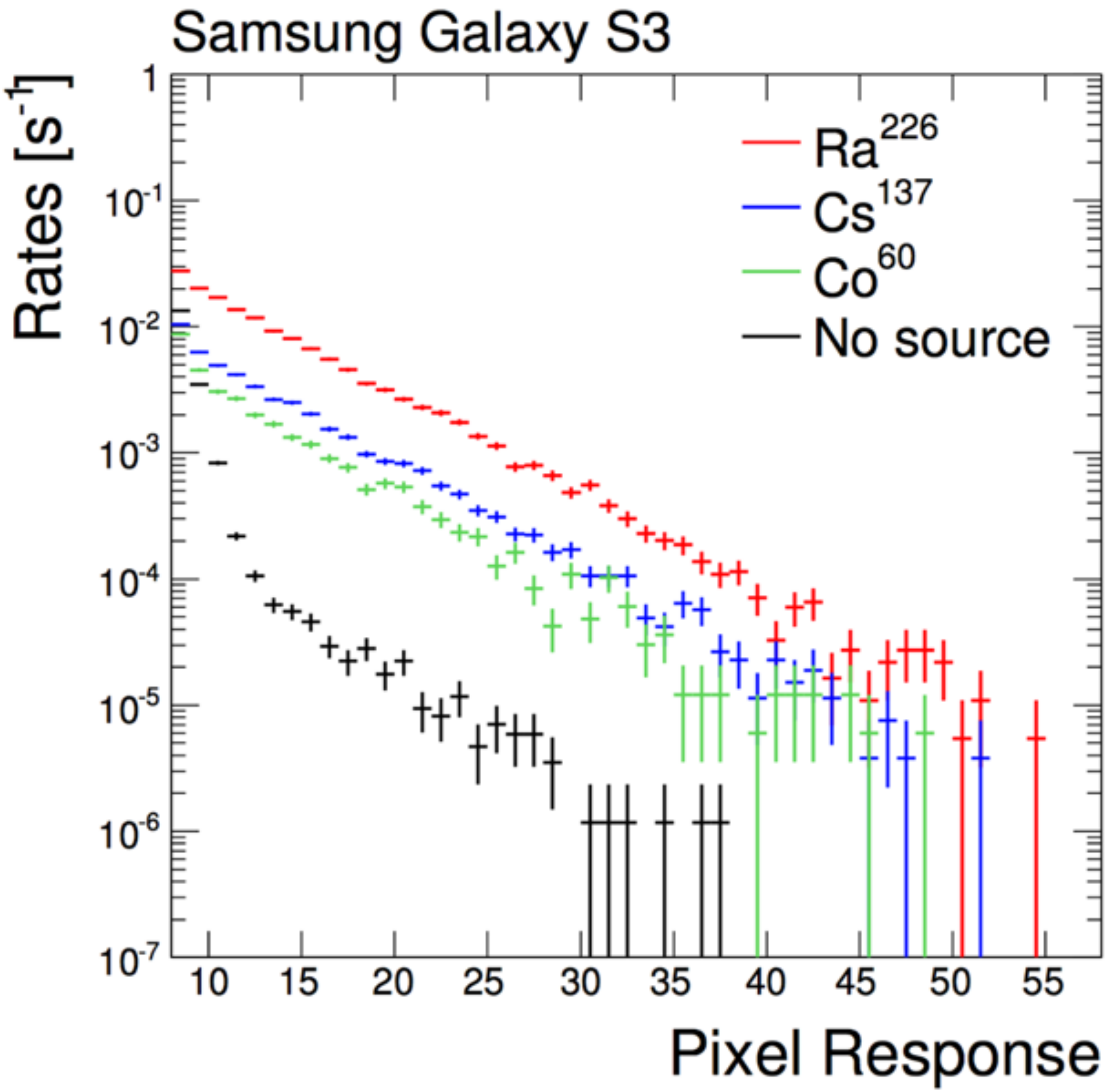
Pierre Auger Observatory



Photon Sensitivity



Photon Sensitivity



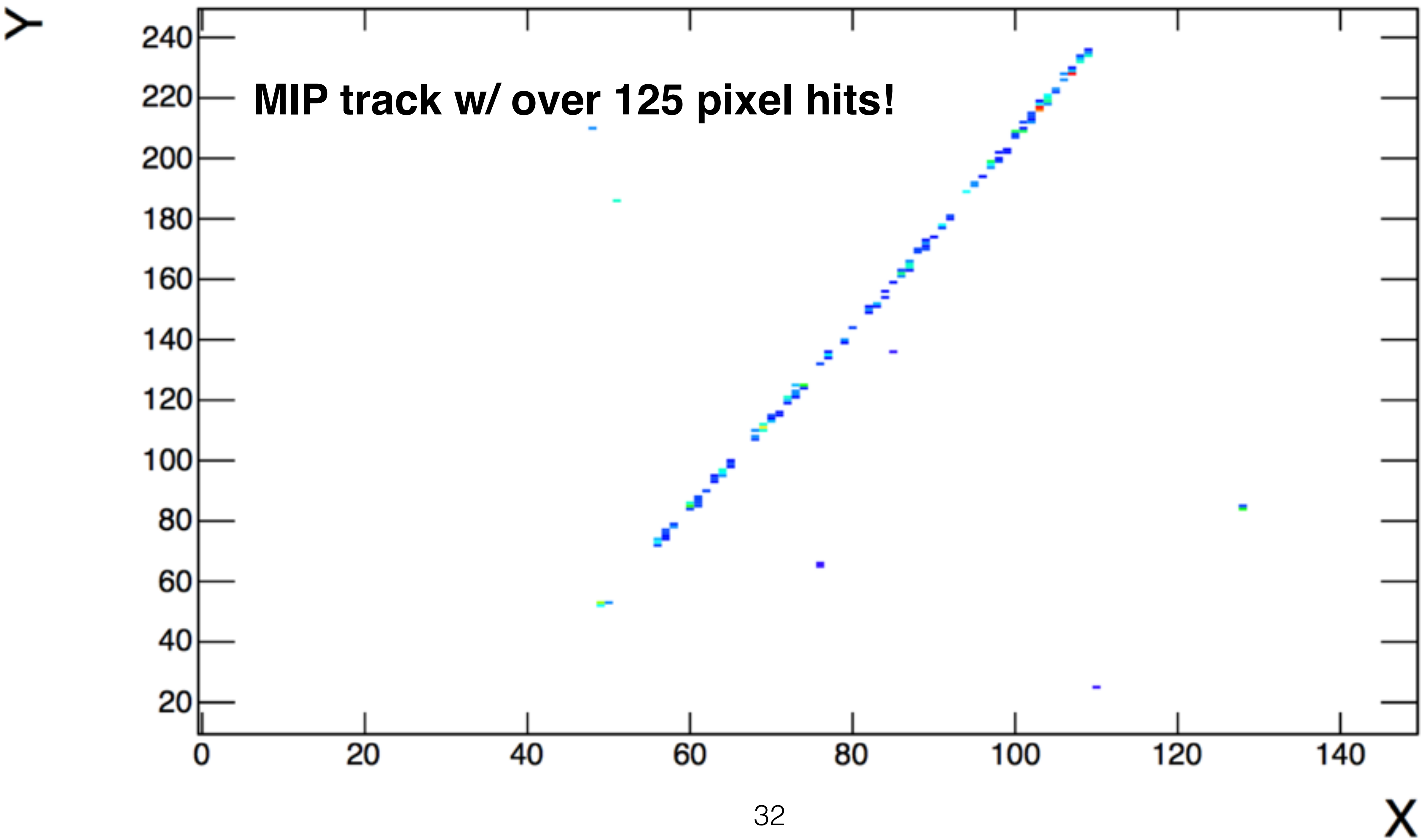
Sources w/ varying activity, energy:

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

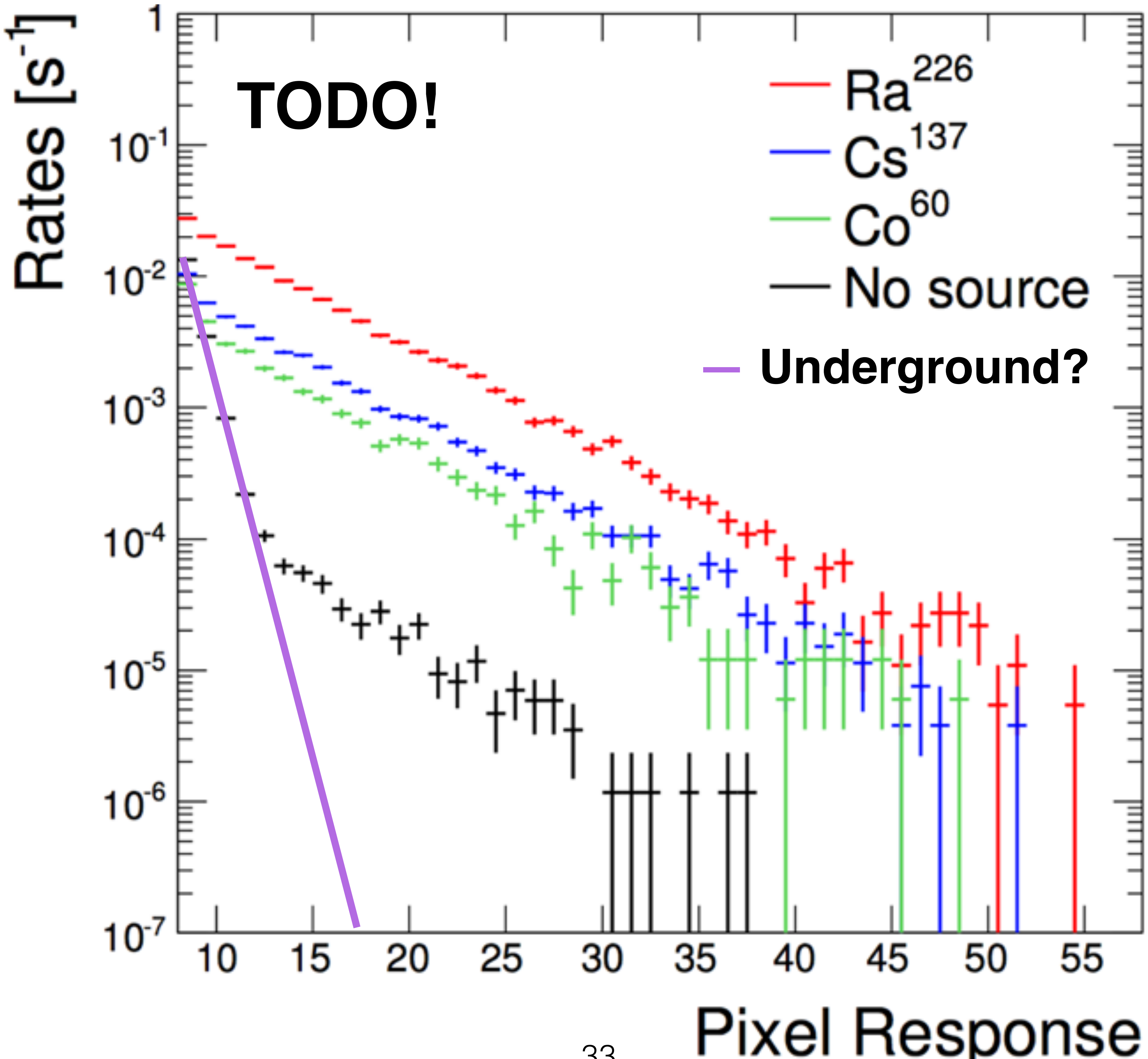
Muon Sensitivity

Get them for free from the sky!

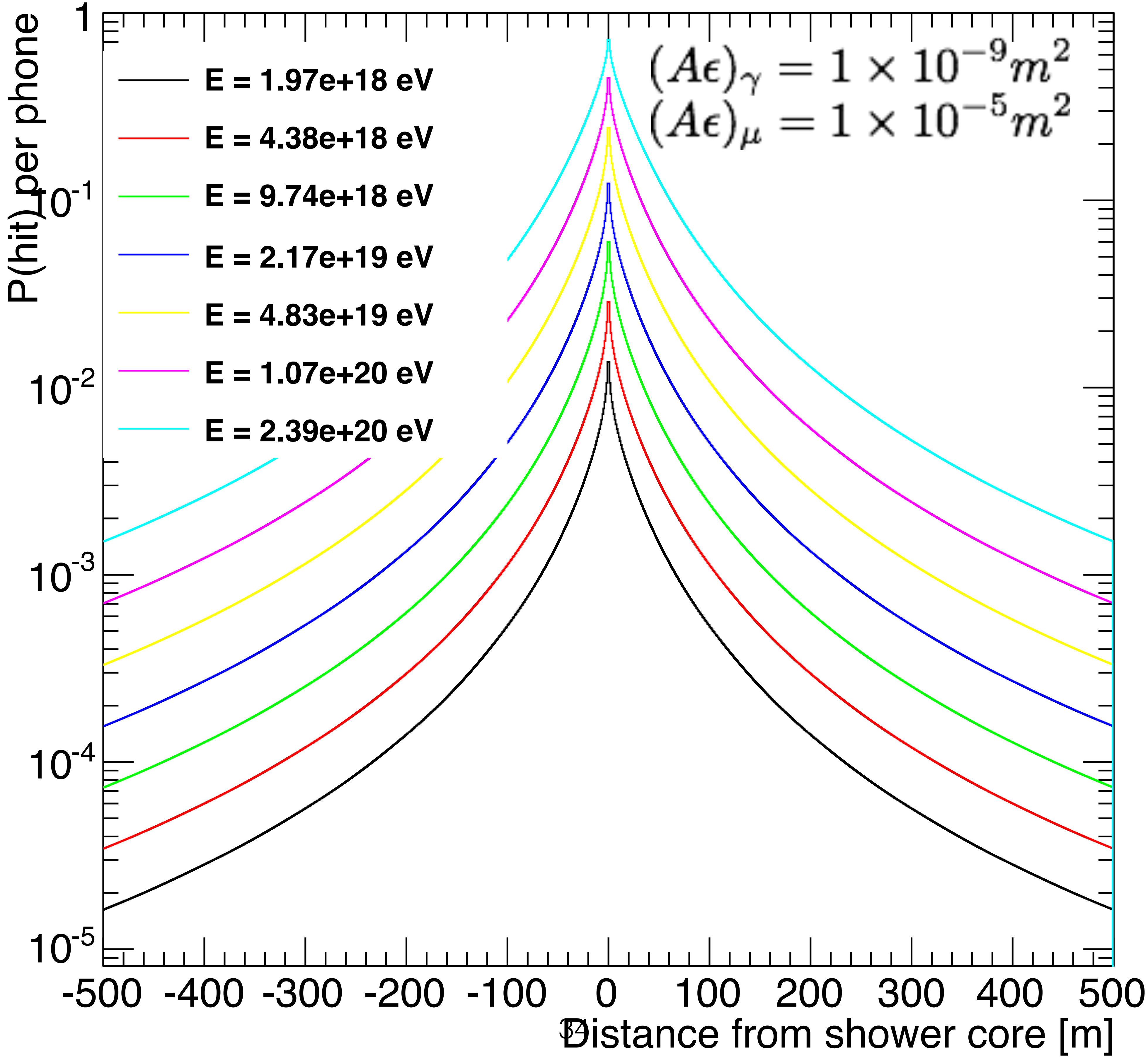
1 muon/cm^2/min $\Longrightarrow$ 1 muon every 4 mins



Muon Sensitivity



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!)