The Case for Comprehensive Diagnostics

Musings on Cloud Diagnostics

Chas DiFatta (chas@cmu.edu)
Jim Gargani (jgargani@cmu.edu)
Mark Poepping (poepping@cmu.edu)

Carnegie Mellon

Diagnostics…?

You discover your car has a flat tire...

- You fix it you move on

It's flat again a week later...

- Valve problem?
- Nails in the driveway?
- Is Majd playing with my head?

Can you check all failure possibilities?

- Might help if you knew when air started leaking

Cars… Computers…

You discover your job-tracker daemon crashed...

- You restart it and you move on

It crashes again a day later...

- Configuration problem?
- Performance or resource problem?
- New bug or integration problem with VM provisioning system?
- Security vulnerability? Is it really my tracker running or a rogue daemon?

Why Diagnostics?

- Things break, in complicated, partial ways – and it matters
- Systems built to ‘get it working’, not to be ‘fixed’
  - Meter/maintain/fix after installation?
  - The maintainer learns how… but it’s a struggle
- Software reuse and layered infrastructures create dynamic dependencies
  - Diagnostic data may not be available at all
  - Certainly doesn’t follow service path
  - Minimally ‘out of band’, often ‘out of question’
- Service Plane + Management Plane + *Diagnostic Plane*

Who are the Diagnosticians?

In IT (lots of other diagnostic domains):

- Applications Support Personnel
- Systems/Cloud Administrators
- Network Support Staff
- Security Response Teams
- Managers of Computing Infrastructure
- Help Desk
- End Users

Who are the Diagnosticians?

In IT (lots of other diagnostic domains):

- Applications Support Personnel
- Systems/Cloud Administrators
- Network Support Staff
- Security Response Folks
- Managers of Computing Infrastructure
- Help Desk
- End Users

Everybody

Carnegie Mellon

Banes of Diagnosticians

Validated through Interviews

- Limited access to slices of diagnostic data
- Discovering valuable information in a sea of data
- Correlating different diagnostic data types
- Providing evidence for non-repudiation of a diagnosis
- Finding time to create tools to transfer diagnostic knowledge to less skilled organizations and/or individuals (automation)

An Illustration

Someone reports the Hadoop application seemed slow at 2pm

- You look around, but it seems fine and you move on.

Someone else reports the problem again a week later...

- Configuration or task-tracker problem?
- Downstream congestion problem caused by large file transfers?
- Networking problem?
- How many potential failure scenarios?

An Illustration (2)

What's Involved?

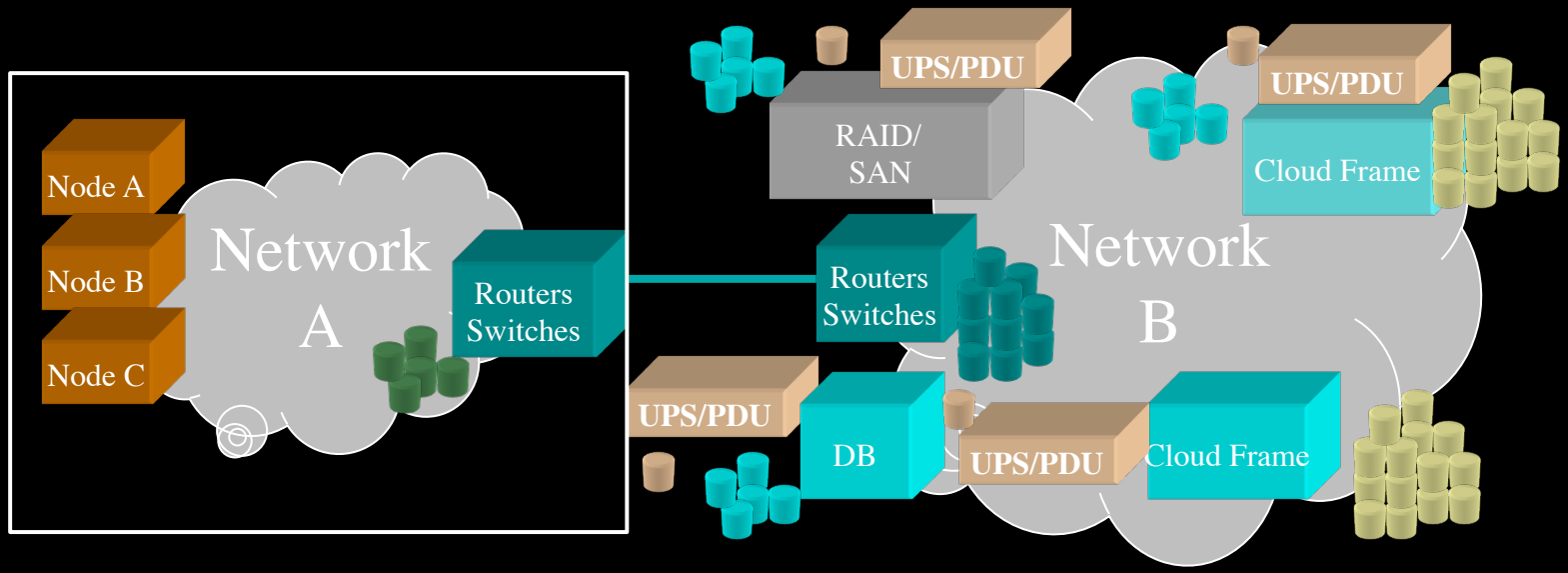
- Peer network routers/switches
- RAID and SAN devices
- Application servers
  - Task-trackers, job-trackers, end node cloud application
- Maybe:
  - Configuration problem
  - Resource contention
  - Intermittent device failure

An Illustration (3)

Present day manual process for resolution

- Map DNS/DHCP/IP address/MAC address
- Inspect historical network statistics on all devices in the path
  - Interface information: byte, packet and error counts
  - Device health: CPU, memory, power supply, etc.
  - Network flow records of devices in question
- Manually inspect logs/statistics on server and client applications and middleware/cloud systems

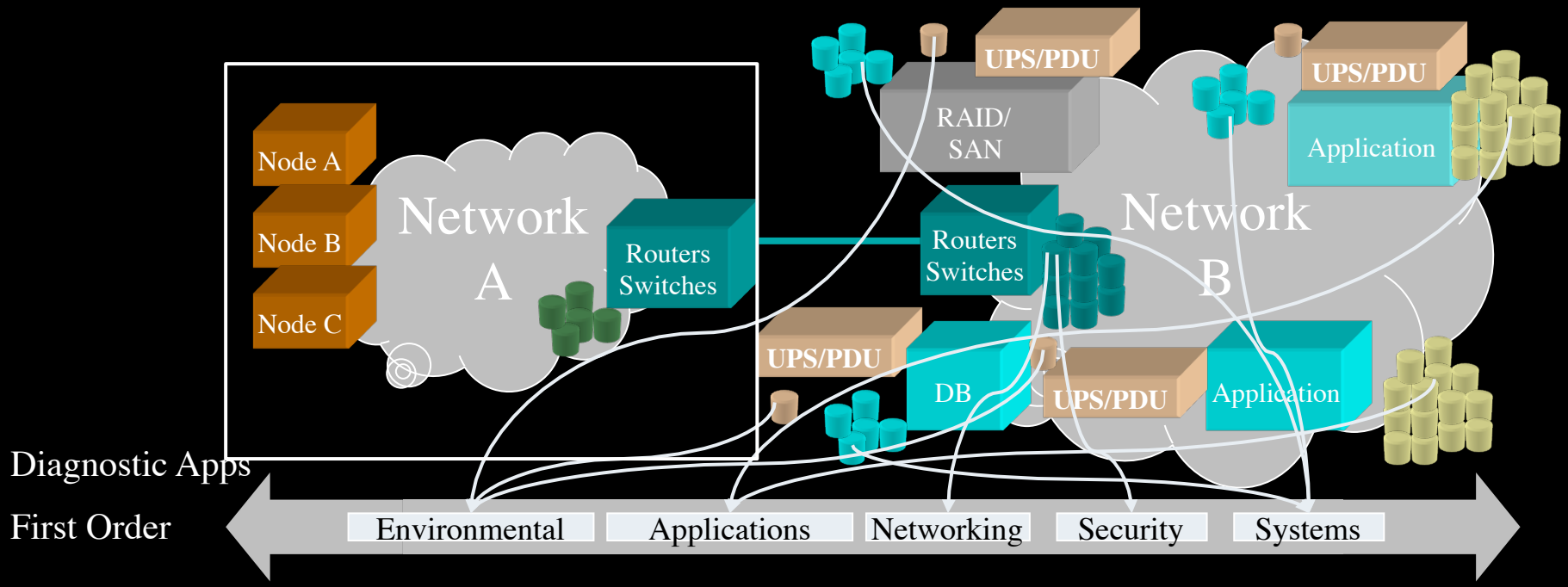
Separate Event Domains



Network Info
Application Info
System Log Info
Environmental Info

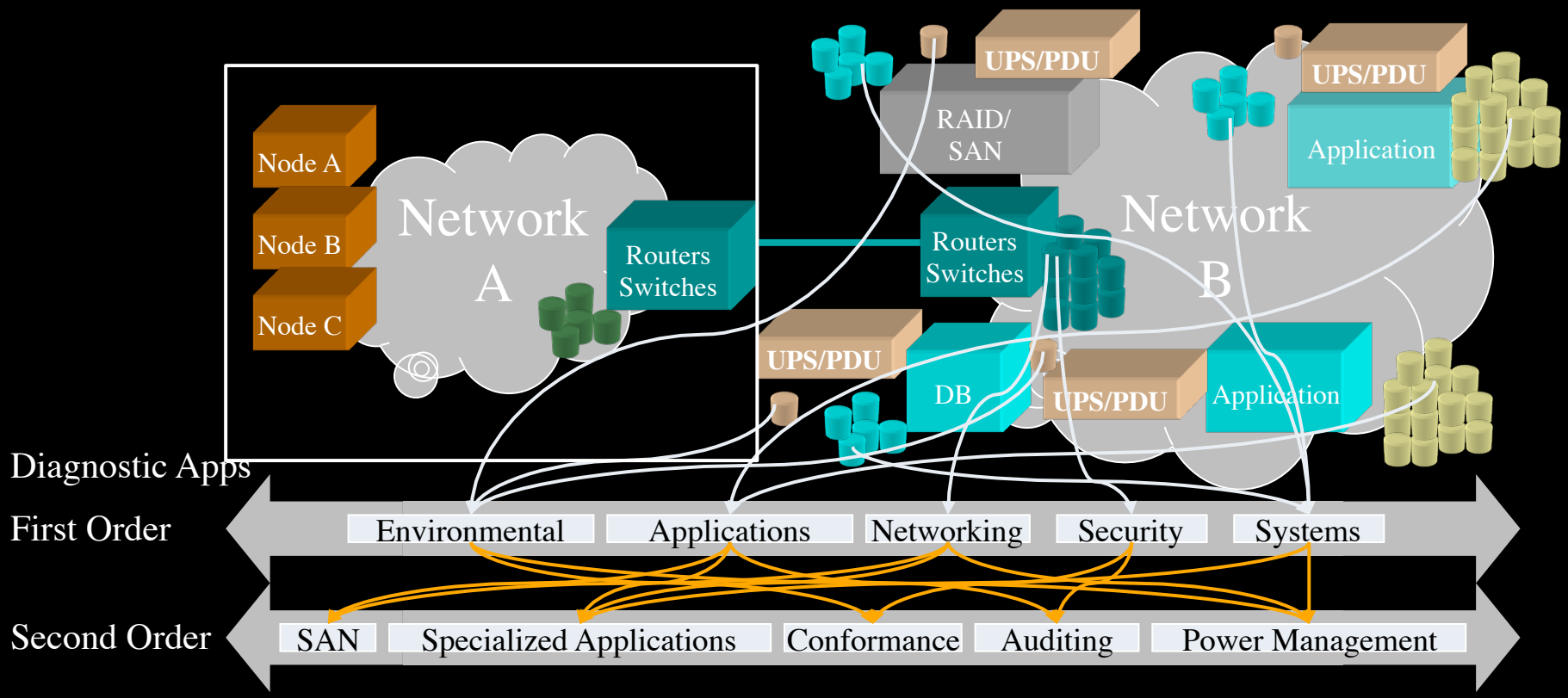
Carnegie Mellon

Collecting Event Domains



Network Info
System Log Info
Application Info
Environmental Info

Integrating Event Domains



Carnegie Mellon

Cloud Administrator Question

- Why was the cloud application slow?
- The redundant power supply failed on the RAID (using the SAN) caused by a PDU failure and the RAID was cycling between write through and write back mode.

Thinking about the Problem

[A Layered Architecture for Diagnostic Infrastructure]

1. Sensing Technology
   - State, transaction info, whatever…the ability to collect anything
3. Diagnostic Data Orchestration
   - Data acquisition/normalization/transport, getting the:
     - Instrumentation data you want
     - In the format that you need
     - Where you want it
5. Diagnostic Information
   - Generic translation and statistical methods
   - Simple event correlation, visualization, longitudinal pattern analysis
   - Data Lifecycle (must be policy driven)
7. Domain-specific Diagnostic Analytics
   - Detailed analyses, situational diagnosis, specialized UI's
   - Significant automation of the domain and implementation autonomics

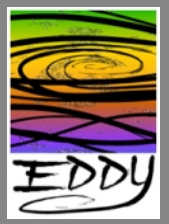
Discussion

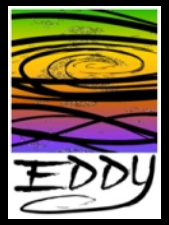
- What are the problems/complexity that you have with using the cloud?
- What diagnostic data would you need?
- What debugging/performance tools would you need?
- What types of visualization/presentation?

Thinking about the Problem

[A Layered Architecture for Diagnostic Infrastructure]

1. Sensing Technology
   - State, transaction info, whatever…the ability to collect anything
3. Diagnostic Data Orchestration
   - Data acquisition/normalization/transport, getting the:
     - Instrumentation data you want
     - In the format that you need
     - Where you want it
5. Diagnostic Information
   - Generic translation and statistical methods
   - Simple event correlation, visualization, longitudinal pattern analysis
   - Data Lifecycle (must be policy driven)
7. Domain-specific Diagnostic Analytics
   - Detailed analyses, situational diagnosis, specialized UI's
   - Significant automation of the domain and implementation autonomics





EDDY Capabilities

[Orchestrate Data and Create Generic Information]

Enable correlation

- Common Event Record (CER) – a way to format event information to make it easier to process
  - TTL, timestamp, observation point, normalizer location, event type, GUID, severity, user defined tags
  - Extensible payload, leverage domain data formats

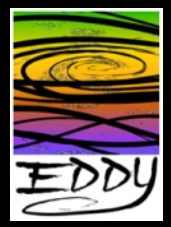
Provide transport

- Diagnostic Backplane – a way to move CERs around to make it easier to automate processing
  - High performance and XMPP

Some simple event orchestration methods

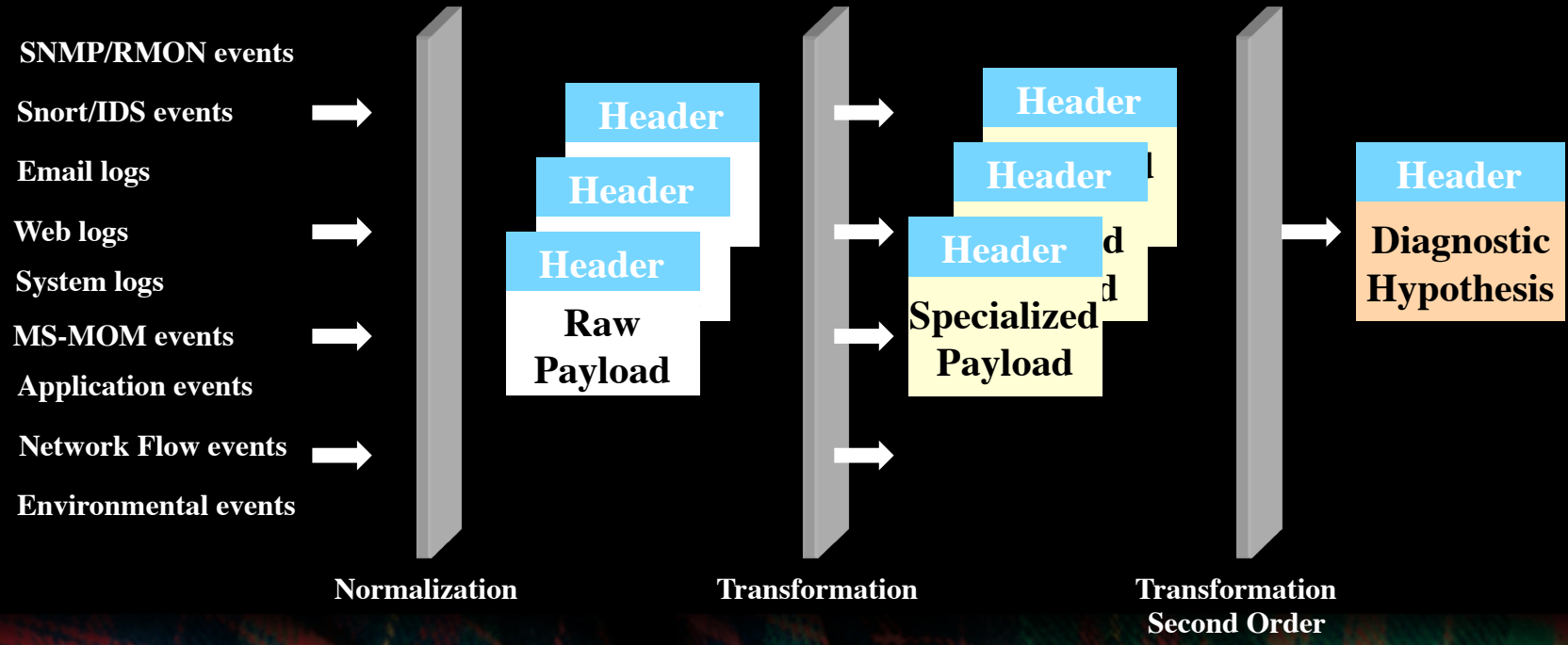
- Normalize, transform, visualize, store, anonymize

Carnegie Mellon

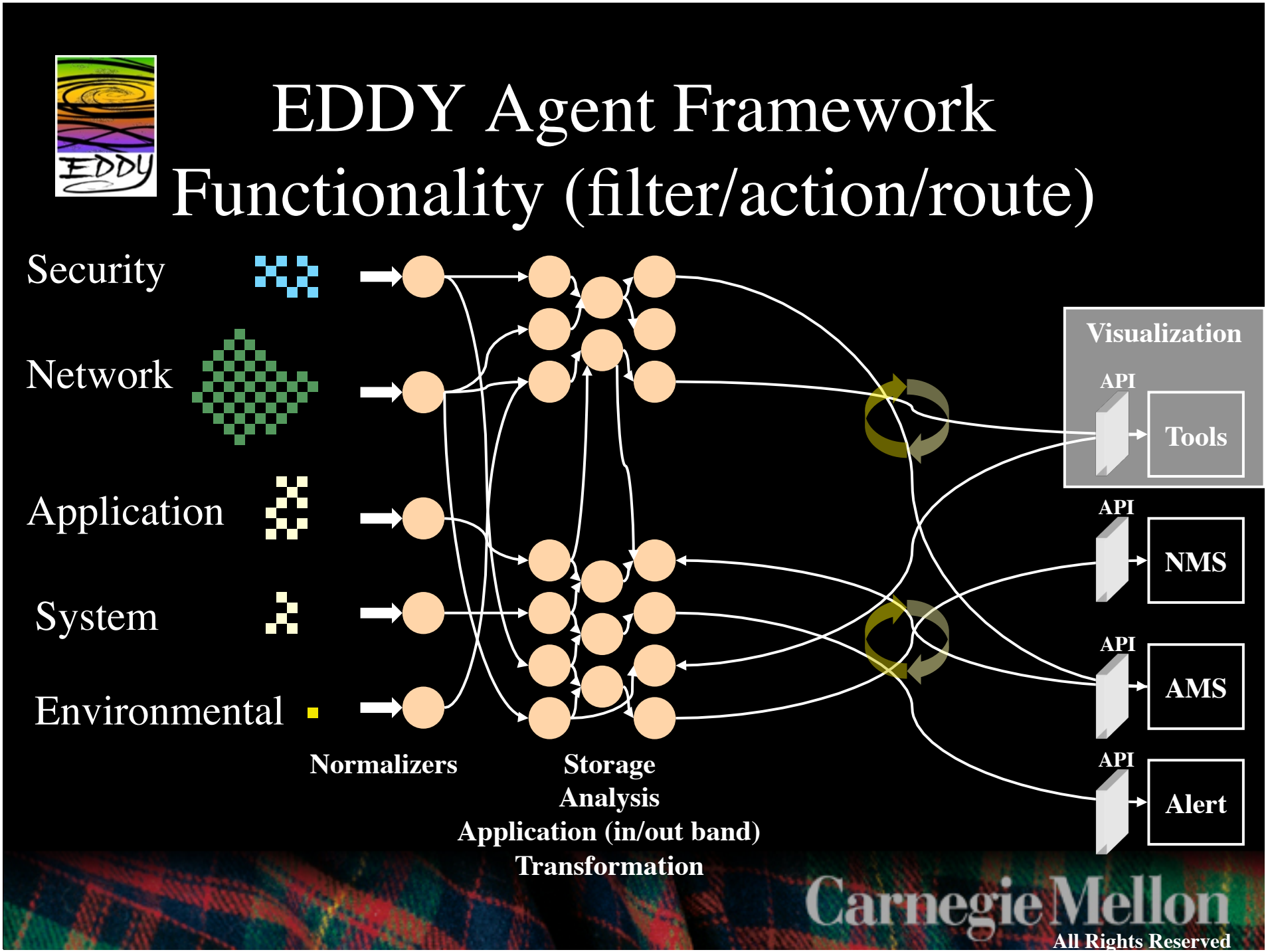


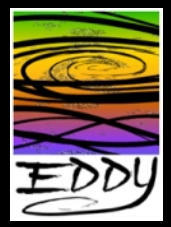
EDDY Extensibility and Scalability

You don't need all the data, pick off only what you need...

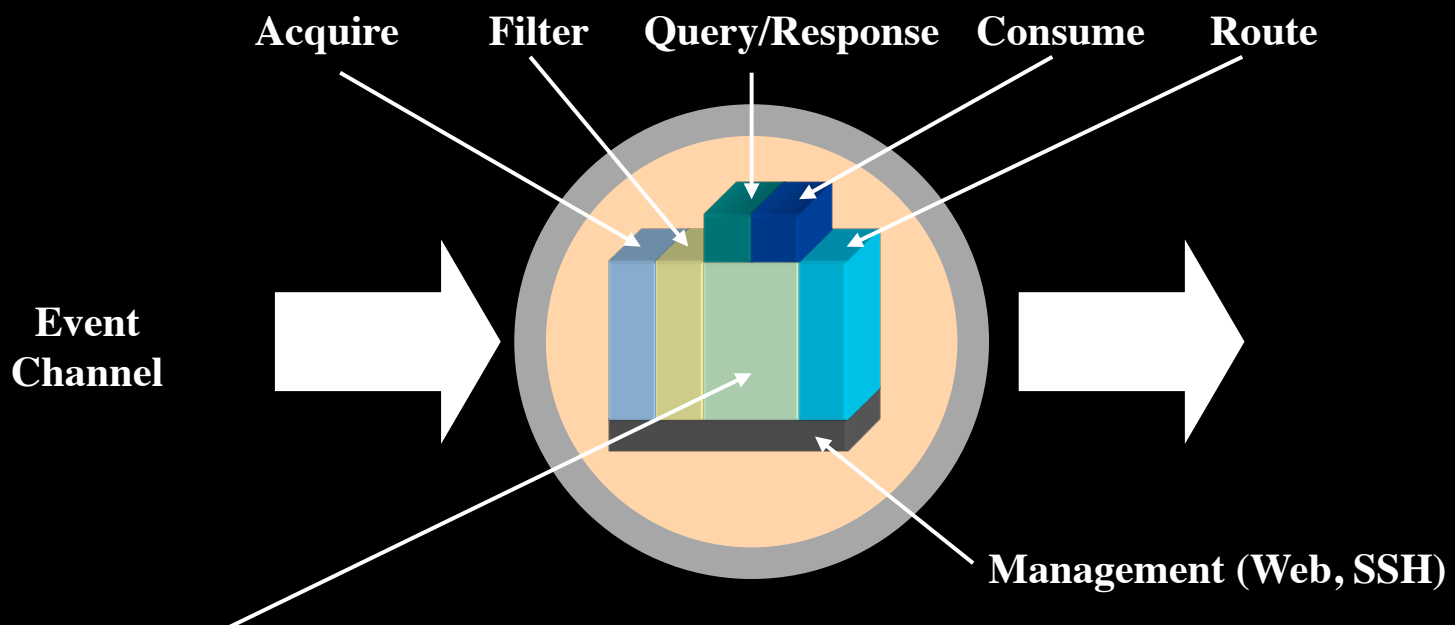


Carnegie Mellon


EDDY Agent Framework
Functionality (filter/action/route)
Security
Network
Application
System
Environmental
Normalizers
Storage
Analysis
Application (in/out band)
Transformation
Visualization
API
Tools
API
NMS
API
AMS
API
Alert
Carnegie Mellon
All Rights Reserved

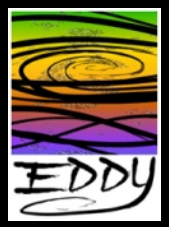


EDDY Agent Appliance Anatomy

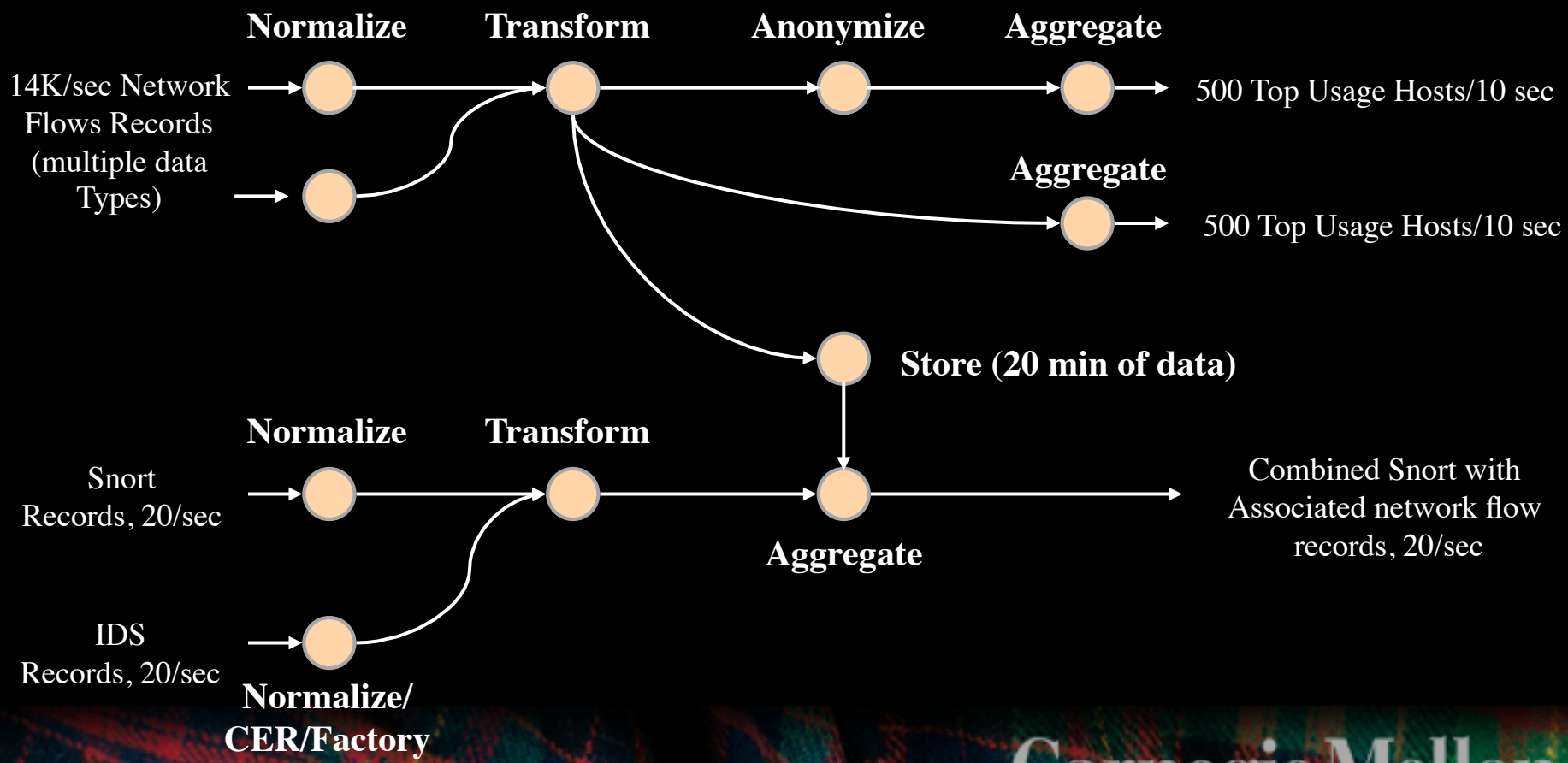


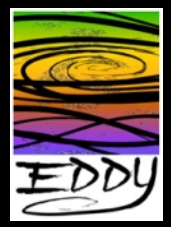
Transform: to name few....
Anonymize , store, archive, morph (many flavors), join, transfer (external communication), aggregation, normalize, etc.

Carnegie Mellon

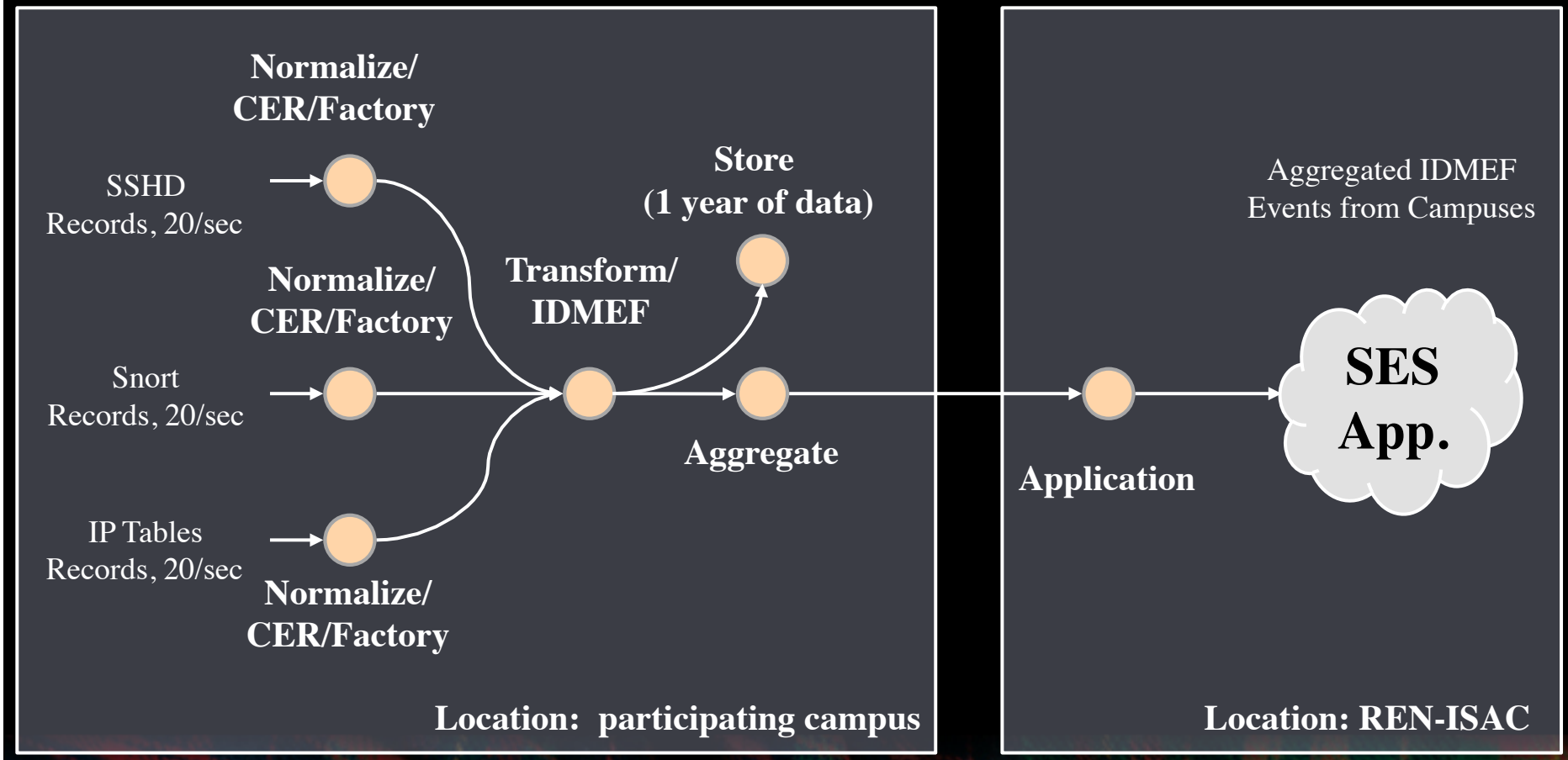


Appling domain agnostic methods to domain specific solutions

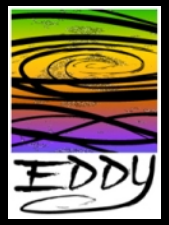




Domain specific solutions cont. SES/REN-ISAC Security App.

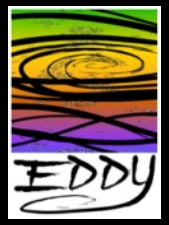


Carnegie Mellon



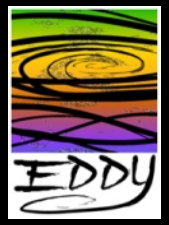
EDDYgrep

- Command line Unix grep like to that extracts CER's via filtering of attributes
- Futures
  - Adding Unix Awk like functionality to add/edit date to CER
  - Inject results directly into external DB



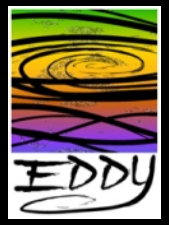
EDDY Futures

- Experiment long haul event transport (Qatar)
- Add to the QCloud research environment (Qatar)
- Mature the appliance
- Experiment with data mining cloud events



What EDDY is

- Architecture for cross domain diagnostics
- An enabling technology that provides
  - Event ledger
  - Dissemination and correlation infrastructure,
    - Afford research access to event data (anonymized)
  - A development platform for diagnostic research
    - Domain specific
    - Domain agnostic



What EDDY is not

- A system/network/application/security management platform
- The analysis engine, it enables the analysis to happen with domain expertise

Carnegie Mellon

Carnegie Mellon CyDAT

Cyber-center for Diagnostics Analytics and Telemetry
The Cloud Diagnosing Itself

- Architecture and Standards
  - Design and define specifics for the IT Diagnostic Plane
  - Standards for data format and transport
- Open Source Prototype
  - A reference implementation for experimentation with the Diagnostic Plane
- Observatory
  - Leverage a large-scale event facility at Carnegie Mellon for engineering and research collaboration on real data
  - Computing Services provides data, needs engineering analyses
  - Facilitate data export to other researchers
  - Research on structure and behavior of the Diagnostic facility
  - Engage corporate collaboration

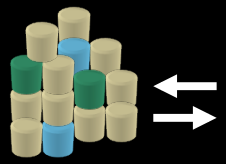
Carnegie Mellon

Observatory Service Infrastructure

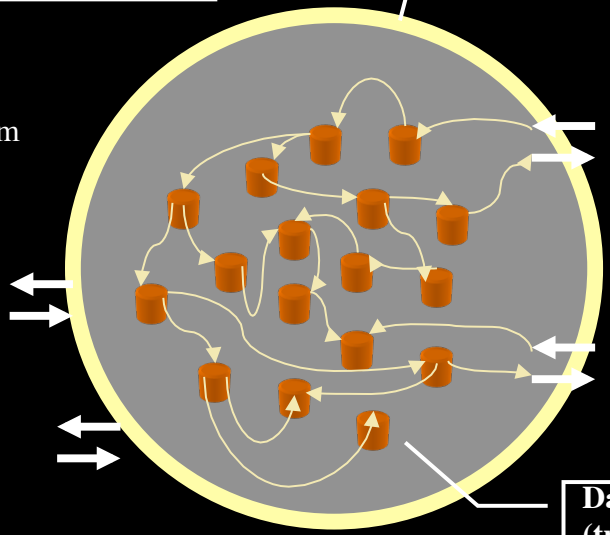
- Data flow:
  - publish/subscribe (EDDY/XMPP) <1K/s
  - high performance push API (EDDY/HP) 14K/s+
- Query: under development
- Common data format: EDDY Version II

Real data from experimental sensor networks

Real data from production enterprise systems, networks, and environmental information. Custom probes, syslog aggregation, network flows



Researcher Analysis Applications



Firefly Networks

Robotics, MOTES, etc.

Data orchestration services (translation, storage, etc.)

Data orchestration/storage agents

Data orchestration/storage agents

Application/System Events

Network Events

Security Events

Carnegie Mellon

Observatory Services

- Multi-Campus Infrastructure
  - Import services
    - Sources: researcher sensors, computing services data from servers and networks, from facilities management
  - Access to data (stored and streams) – leverage Andrew authentication/authorization
  - Data Translation
    - Anonymization, aggregation, domain agnostic and domain specific from researcher requests
    - Leverage CyDAT Observatory compute cluster
  - Enforcing policies “in concert” with ISO and IRB
- Futures
  - Collaborative data access to global campuses

Observatory Services Cont.

Data Sources to Date (Pittsburgh, exploring Qatar)

Network flow records

- Argus2-3, netflow5 – campus egress and core
  - Payload available

Campus event

- DNS
- Mon
- Syslog (various events, ex. Smtp logs)
- Sshd
- IPtables
- SES/Aggregation
- Snort

Want to Learn More?

- Web sites
  - www.cmu.edu/eddy
  - www.cylab.cmu.edu/research/cydat.html
- Principal Investigators
  - Chas DiFatta (chas@cmu.edu)
  - Mark Poepping (poepping@cmu.edu)

Carnegie Mellon

Questions/Comments

Carnegie Mellon

The Case for Comprehensive Diagnostics

Musings on Cloud Diagnostics

Chas DiFatta (chas@cmu.edu)
Jim Gargani (jgargani@cmu.edu)
Mark Poepping (poepping@cmu.edu)

Carnegie Mellon