Near Real-Time Multi-Source Flow Data Correlation

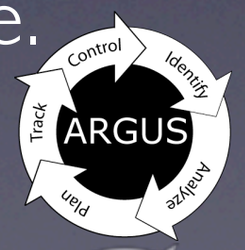
Carter Bullard
QoSient, LLC

carter@qosient.com

FloCon 2013
Albuquerque, New Mexico
Jan 7-10, 2013

Problem Statement

- Cyber incident attribution and forensics, is a complex process.
- To assist in security incident response, recognizable hostile activity needs to be associated with other information system behavior in order to understand the complete cyber security incident life cycle
    - Within a complex internal spoofed stepping stone attack, using a Wiki vulnerability, a machine with an Antartican source address sends a message, that runs a rogue program that sends a command control message to a botnet style agent on an other machine that exfiltrates data back to Antartica.
- For most existing protection strategies, this isn't detectable.

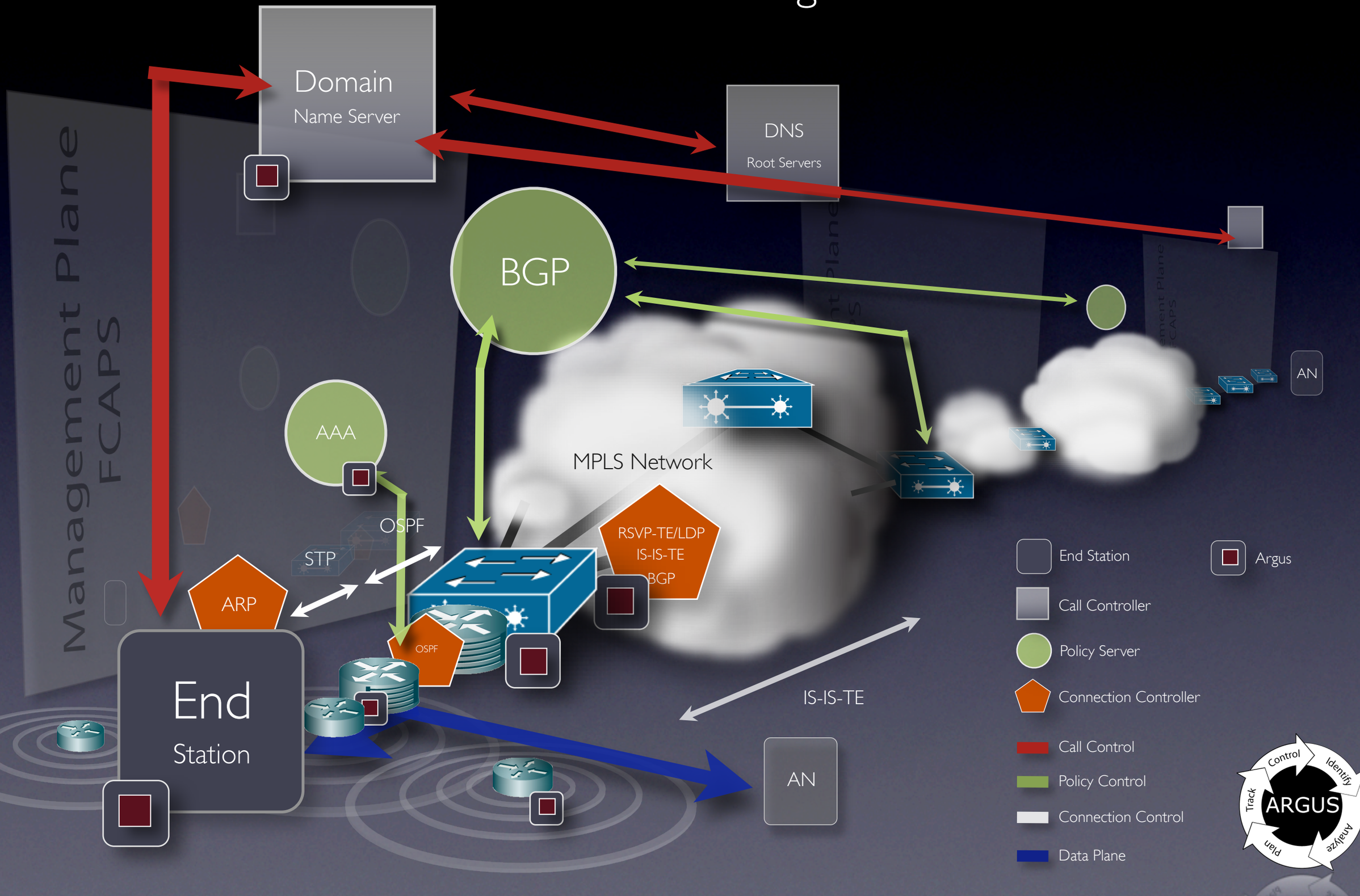


Flow data is an important component

- Exfiltration should be detectable from sensors on the external border, or from a sensor in Antartica - But in this case, nada
- The machine that sent the data should be able to report to something that it sent data to Antarctica - But there aren't any logs that contain that transaction
    - Having some form of audit for the network activity of key hosts, is important.
    - Having a means to associate that transfer with the program that actually sent the data is critical, here.
        - Realizing that that program was run by a program, not by the current user of the system, is important.
- The machine that was accessed by the Antarctican machine, like most internal machines, provide inadequate access control, protection or auditing to track.
    - Associating that program with the stimulating / initiating message from Antarctica is critical
- Realizing that the machine isn't really in Antartica, but its down the hall, is going to be a challenging problem.

Comprehensive Enterprise Awareness

Dealing with the Insider Threat



How to approach this

- Establishing a strategy that can help attribution and forensics analysis for the internal attack
  - Establish formal attribution / non-repudiation systems
- Improve audit so that the basic information is available, reliable, and relevant
  - At least each host should maintain a network activity log
- Improve methods and techniques so that correlation can be used to make the end-to-end attribution possible.
- Currently, for many sites, its really luck, rather than engineering, that makes this stuff work

How to deal with host issues?

- We need to modify system audit strategies to approach this really important problem
- In the absence of direct support, what to do.
  - We can install flow monitors on hosts
    - That will provide the network audit
    - argus is a good candidate
  - We need user and program bindings to flow data to make the back chaining possible to deal with our scenario.
    - Socket audits are possible in some systems
    - Demonstrate using lsof() to provide that info.

Argus Strategy

- In argus we have integrated into the basic argus data generation, collection, processing, storage and analytics, the ability to correlate flow and non-flow data.
- Argus has a facility, Argus Events, that can be used to generate, structure and transport metadata.
- Argus-3.0.6+ supports the collection of many non-flow data sources, including /etc/proc, vm_stat, SNMP data, and lsof() output.
- We've implemented the ability to correlate lsof() data with cached flow data, as a simple example, in all ra* programs

End-to-End Situational Awareness

Network Optimization - Black Core Mesh

SNMP RMON Element Statistics/Traps
Interface Status / Transitional Events
Bulk Link Statistics

System Layer 2-7 Flow Data
System Communication Efficiency
Connectivity / Availability
Offered Load / Loss / Jitter
One-Way Delay (GPS synchronization)
Round Trip Delay

Comprehensive Layer 2-7 Flow Data
Site Communication Efficiency
Enterprise Communication Efficiency
Site Offered Load / Loss / Jitter
Network Transit Times

Comprehensive Layer 2-7 Flow Data
End-to-End Communication Efficiency
Reachability / Connectivity
Received Load / Loss / Jitter
Network Transit Times

Comprehensive Layer 2-4 Flow Data
ISP Communication Efficiency
Ingress Available Capacity / Loss / Jitter
One-Way Delay (GPS Synchronization)
Network Path Status

Comprehensive Layer 2-4 Flow Data
Network Path Assurance / Status
Reachability / Availability Assessment
One-Way Delay (GPS Synchronization)

Core Service Provider Management Domain

Enterprise Management Domain

Comprehensive Flow Monitor

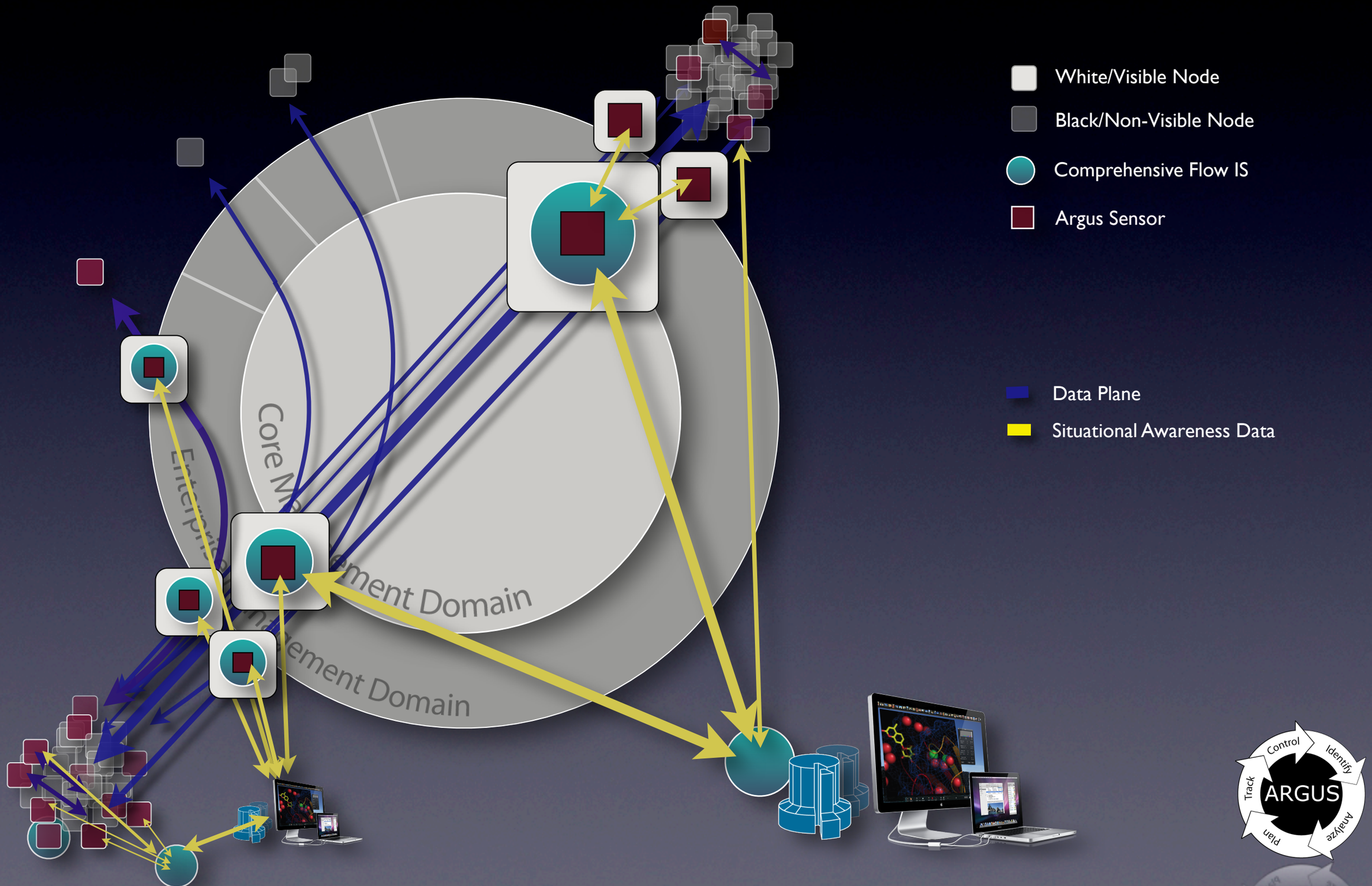
SNMP RMOM Style Monitor

Information System Repository

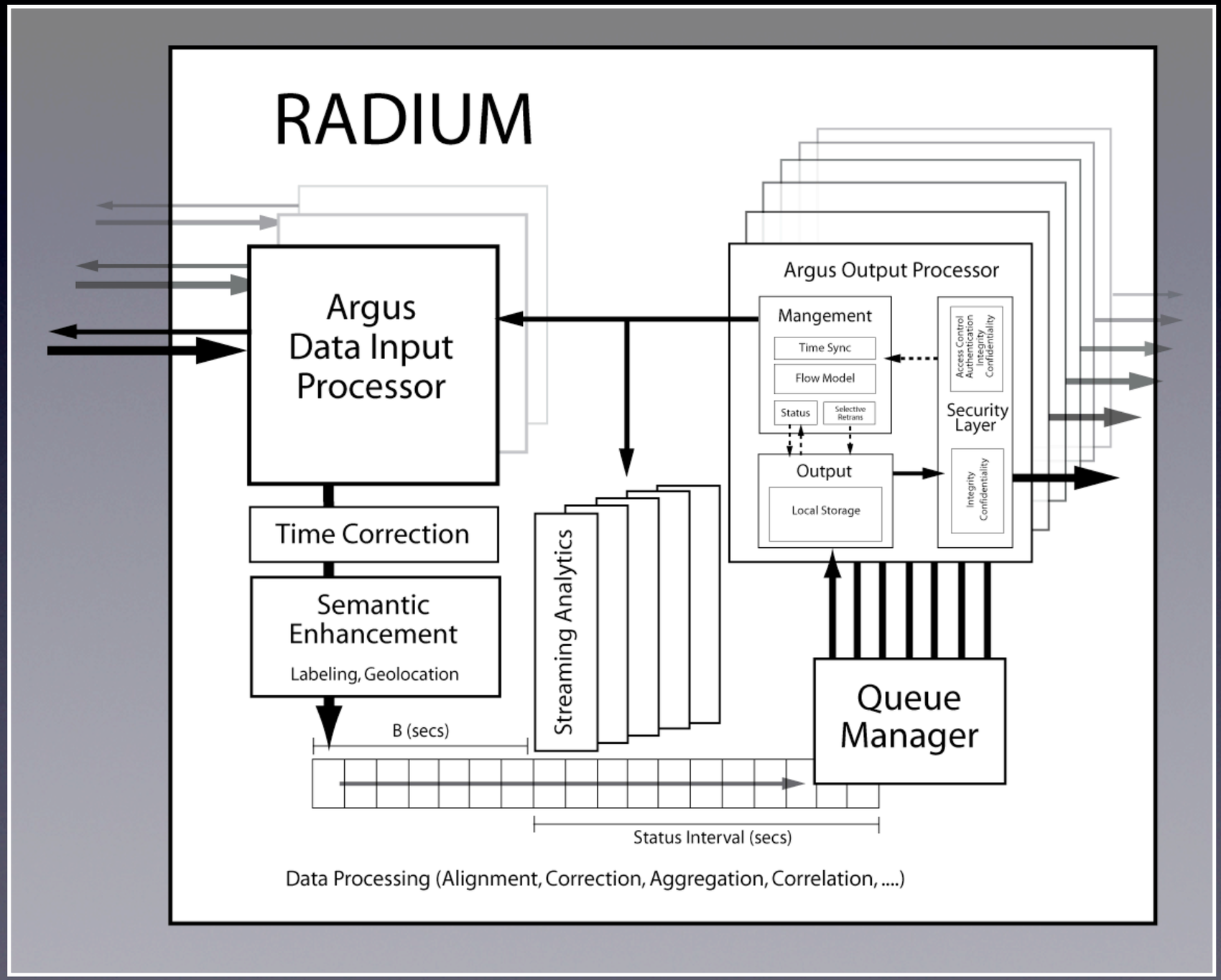
ARGUS
Control
Identify
Analyze
Plan
Track

Complex Comprehensive Awareness

Local and Remote Strategies



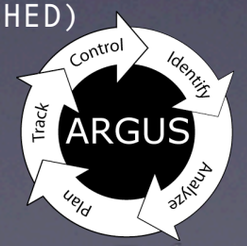
Radium
Data Flow Design



Argus Events

- Argus event type specific format for a particular collection, using a generic XML free form strategy.

```
event[49241]=  2013/01/04.12:47:16.733468:srcid=192.168.0.68:prog:/usr/local/bin/argus-lsof
<ArgusEvent>
  <ArgusEventData Type = "Program: /usr/sbin/lsof -i -n -P">
    COMMAND     PID           USER   FD   TYPE     DEVICE SIZE/OFF   NODE NAME
    mDNSRespo    53 _mdnsresponder   56u  IPv4 0xbb72da10      0t0    UDP *:50451
    awacsd       69             root  241u  IPv4 0xbb72da10      0t0    TCP 192.168.0.68:57367->17.172.208.94:443 (CLOSED)
    apsd         71             root   10u  IPv4 0xbb72da10      0t0    TCP 192.168.0.68:53556->17.149.32.65:443 (ESTABLISHED)
    blued        72             root    4u  IPv4 0xbb72da10      0t0    UDP *:*
    ntpd         75             root   20u  IPv4 0xbb72da10      0t0    UDP *:123
    radium      110             root   10u  IPv4 0xbb72da10      0t0    TCP 192.168.0.68:49166->192.168.0.68:561 (ESTABLISHED)
    radium      110             root   11u  IPv6 0xbb72da10      0t0    TCP [::1]:562->[::1]:49171 (ESTABLISHED)
    [snip]
    Keynote   68546           carter    8u  IPv4 0xbb72da10      0t0    TCP *:49901 (LISTEN)
    raevent   69821           carter    5u  IPv6 0xbb72da10      0t0    TCP [::1]:51255->[::1]:562 (ESTABLISHED)
    perl5.12  69824             root    4u  IPv6 0xbb72da10      0t0    TCP *:561 (LISTEN)
    perl5.12  69824             root    6u  IPv4 0xbb72da10      0t0    UDP *:*
    perl5.12  69824             root    8u  IPv6 0xbb72da10      0t0    TCP 192.168.0.68:561->192.168.0.68:49166 (ESTABLISHED)
    perl5.12  69824             root    9u  IPv6 0xbb72da10      0t0    TCP [::1]:561->[::1]:58040 (ESTABLISHED)
  </ArgusEventData>
</ArgusEvent>
```

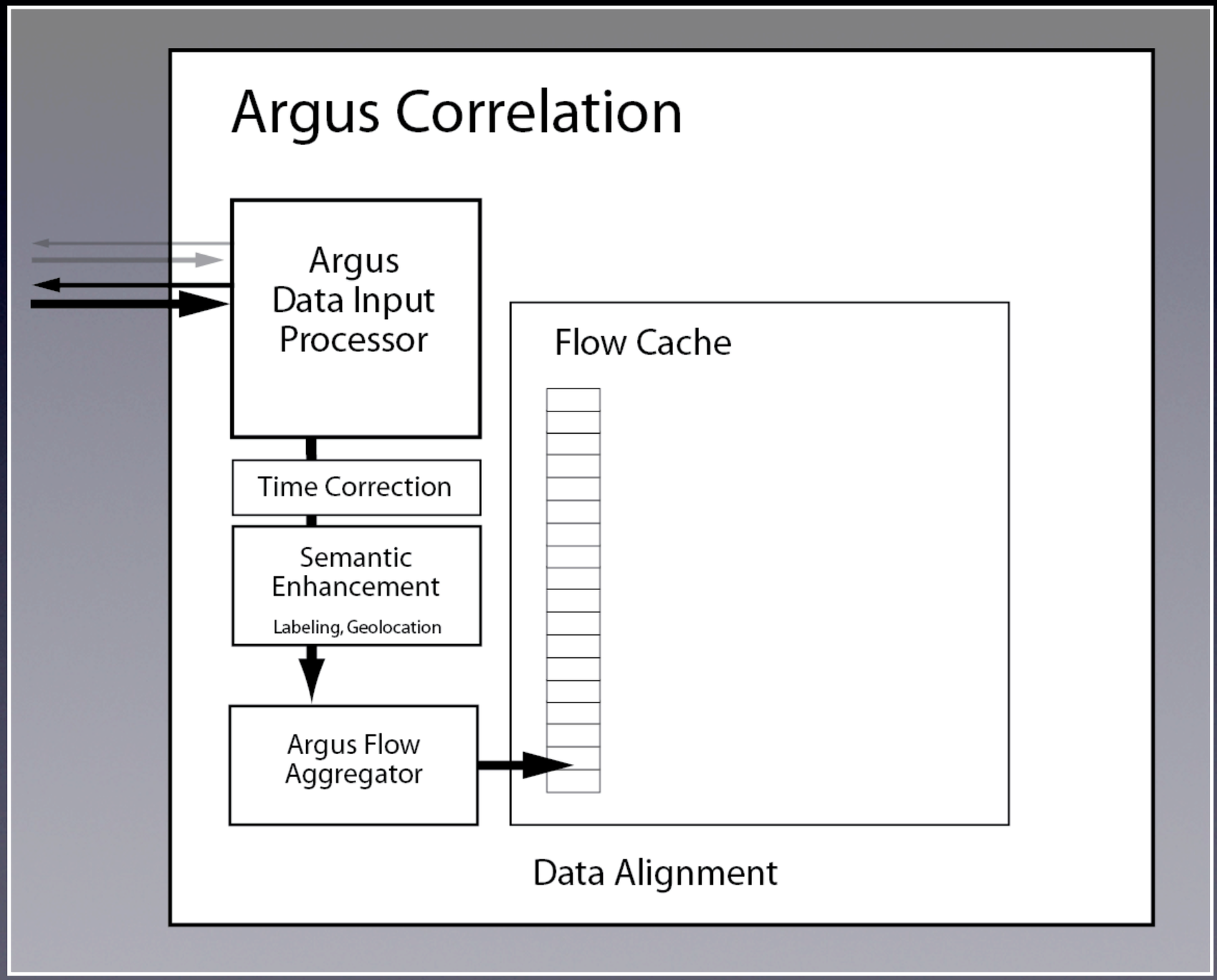


Argus Events Configuration

```
# Argus.conf Argus Event management configuration syntax is:
#       Syntax is: "method:path|prog:interval[:postproc]"
#           Where:  method = [ "file" | "prog" ]
#                   pathname | program = "%s"
#                   interval = %d[smhd] [ zero means run once ]
#                   postproc = [ "compress" | "compress2" ]
#
#ARGUS_EVENT_DATA="prog:/usr/local/bin/ravms:20s:compress"
#ARGUS_EVENT_DATA="prog:/usr/local/bin/rasnmp:1m:compress"
#ARGUS_EVENT_DATA="file:/proc/vmstat:30s:compress"
ARGUS_EVENT_DATA="prog:/usr/local/bin/argus-lsof:30s:compress"
```

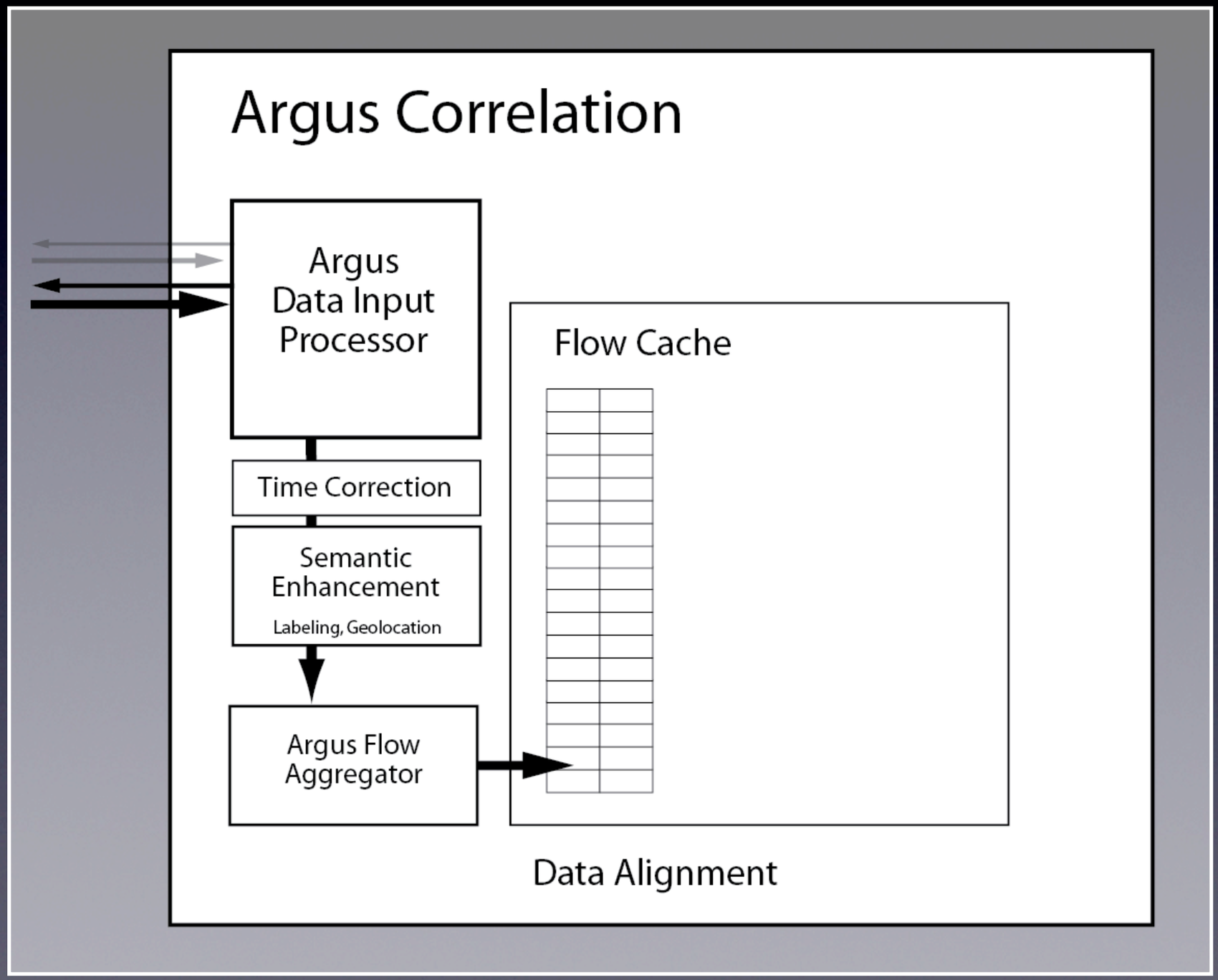
Argus Correlation Design

Radium Process



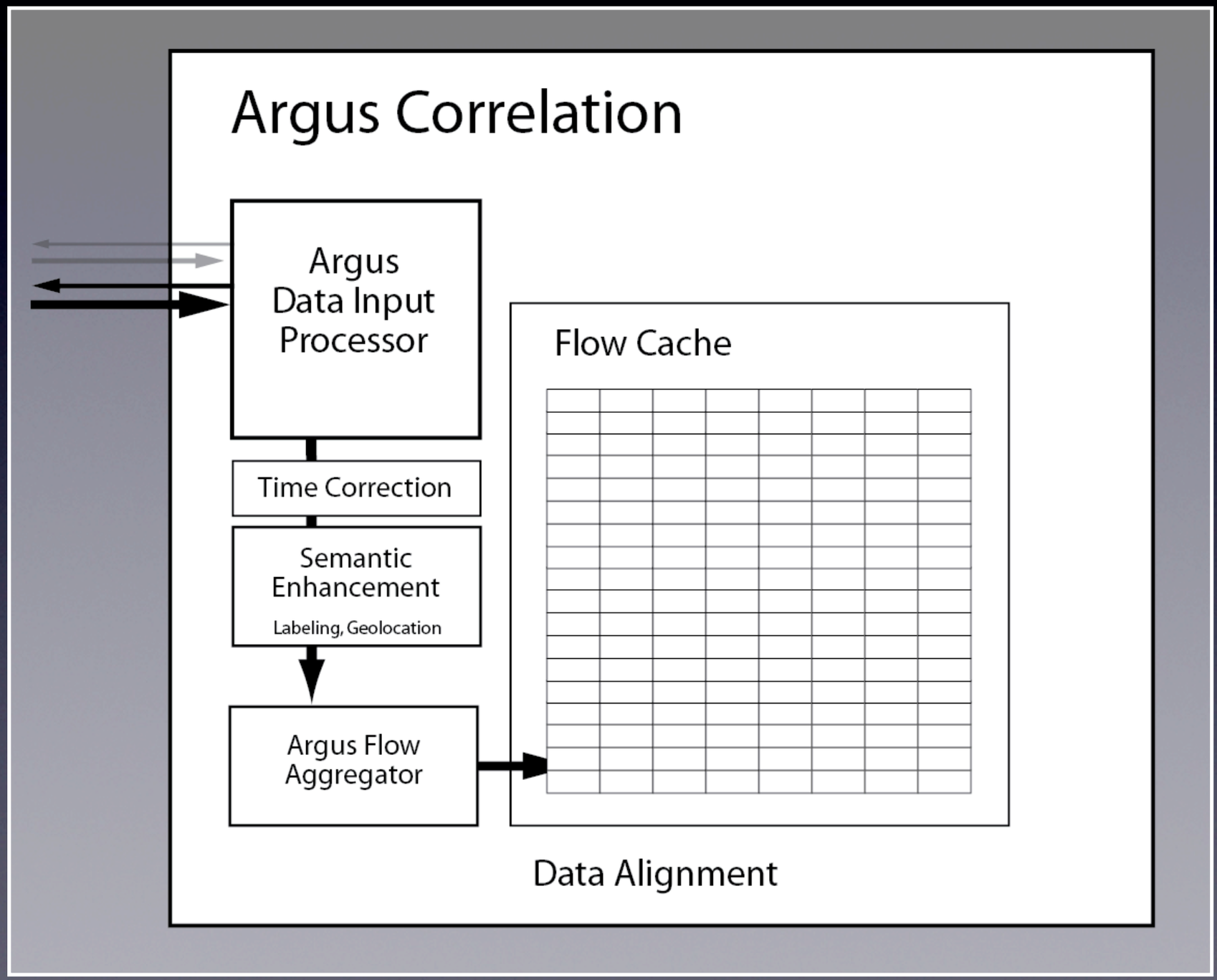
Argus Correlation Design

Radium Process



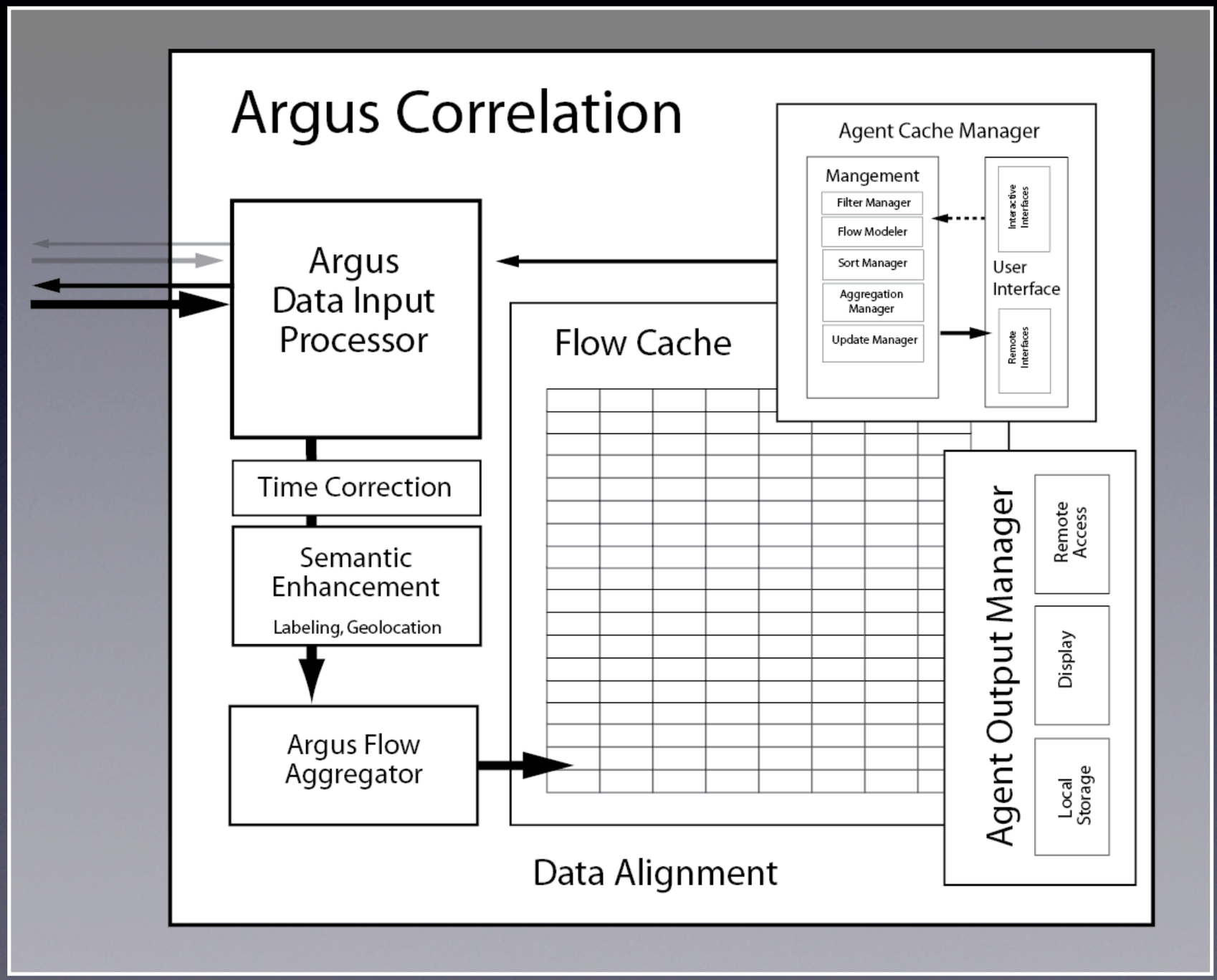
Argus Correlation Design

Radium Process



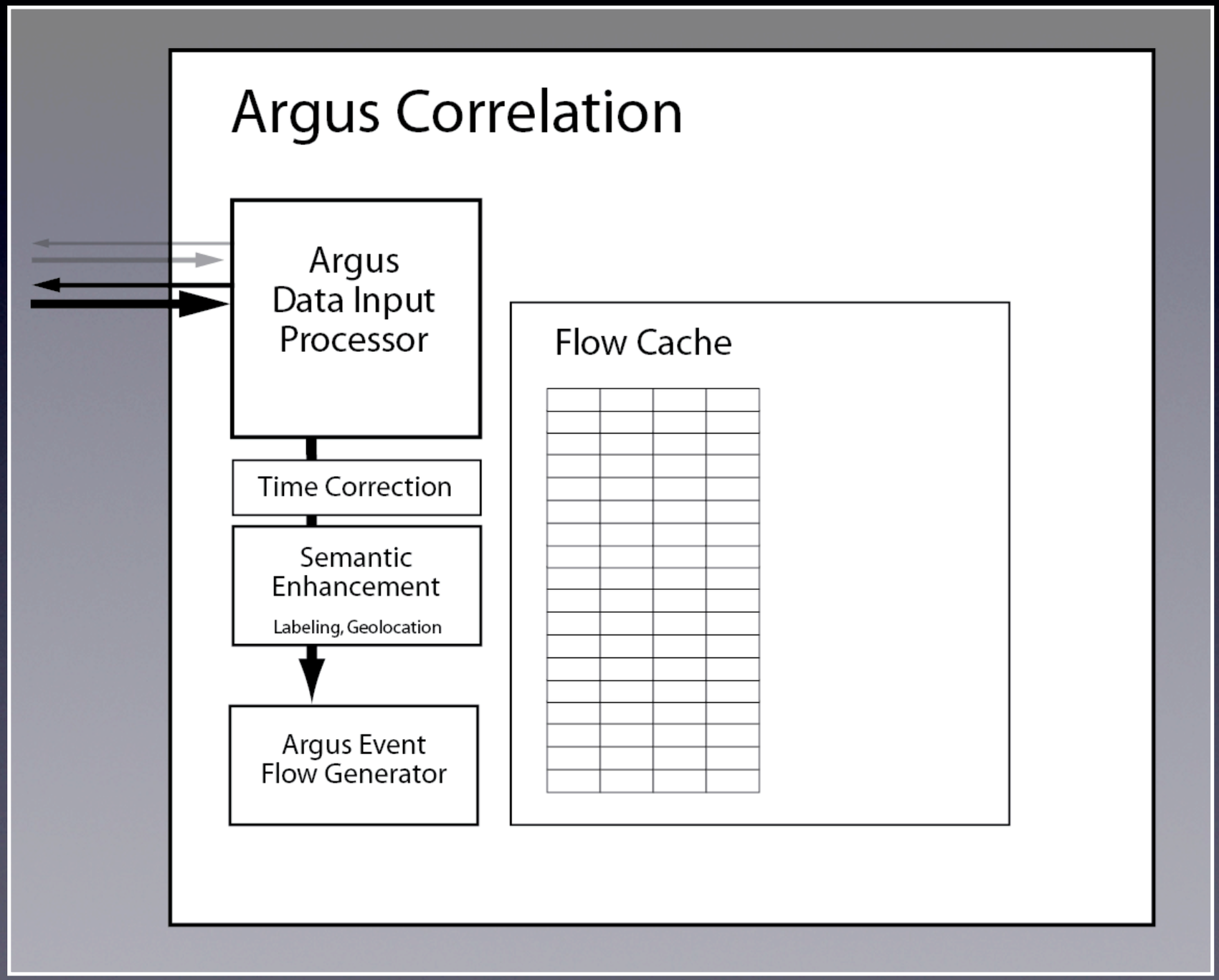
Argus Correlation Design

Radium Process



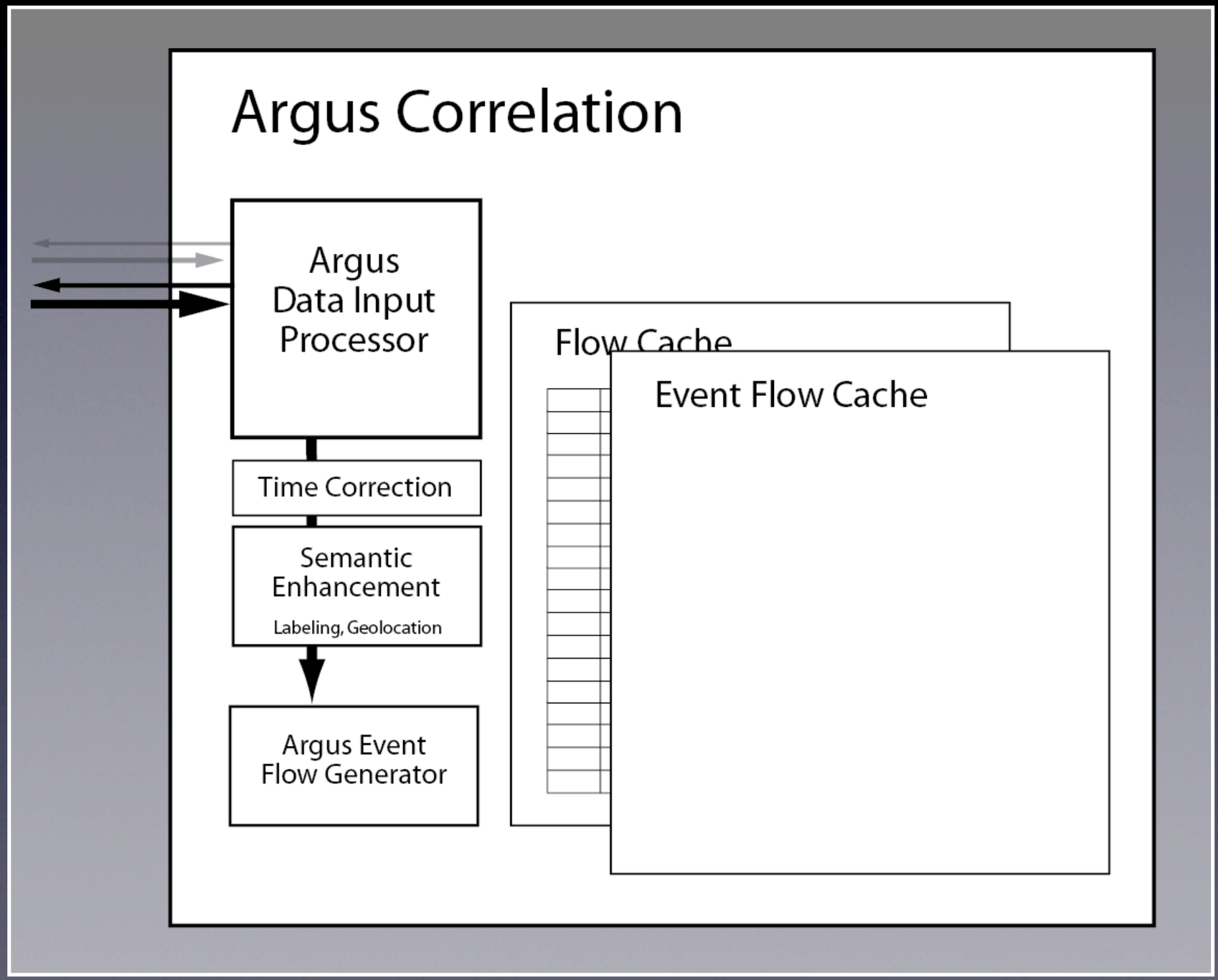
Argus Correlation Design

Radium Process



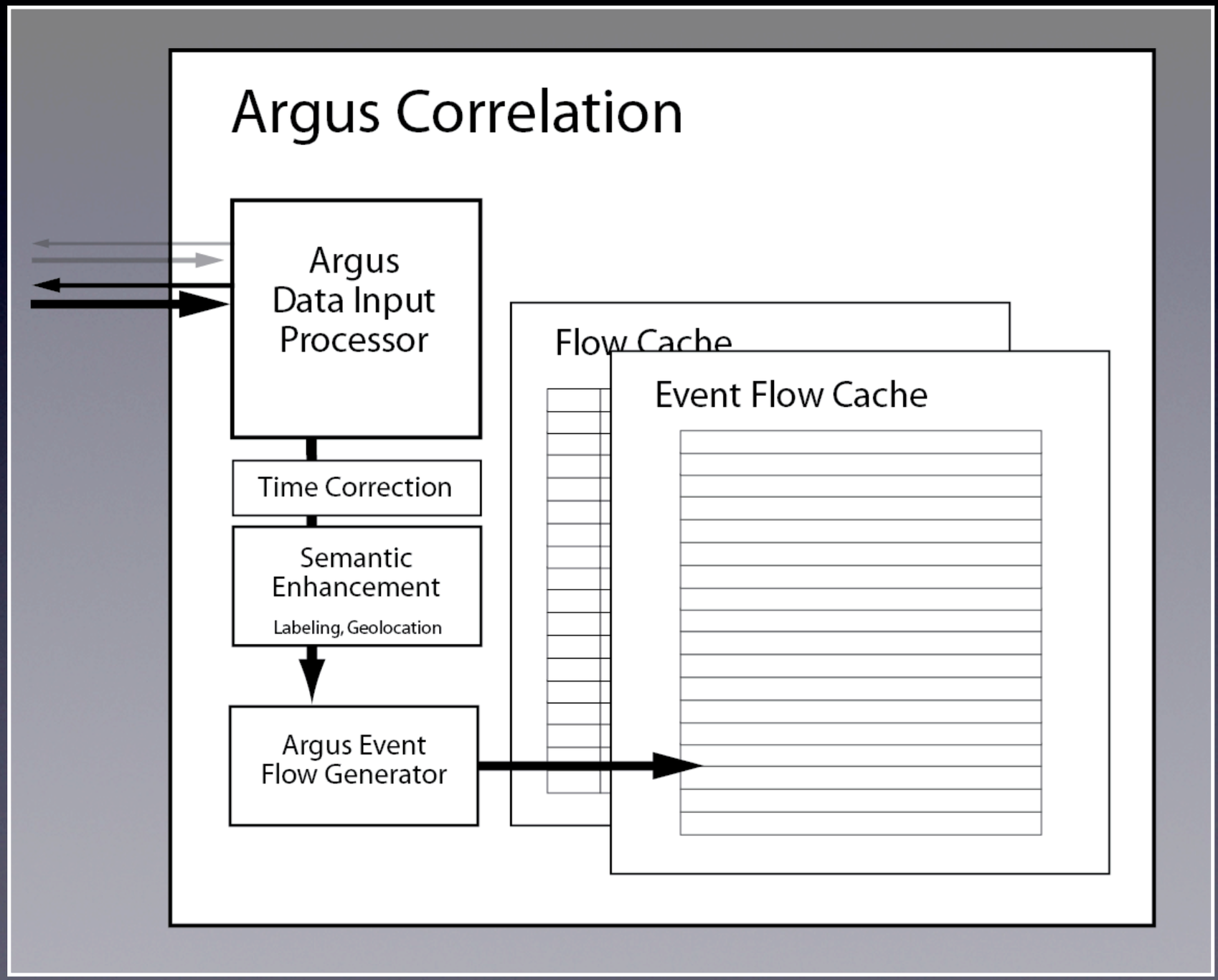
Argus Correlation Design

Radium Process



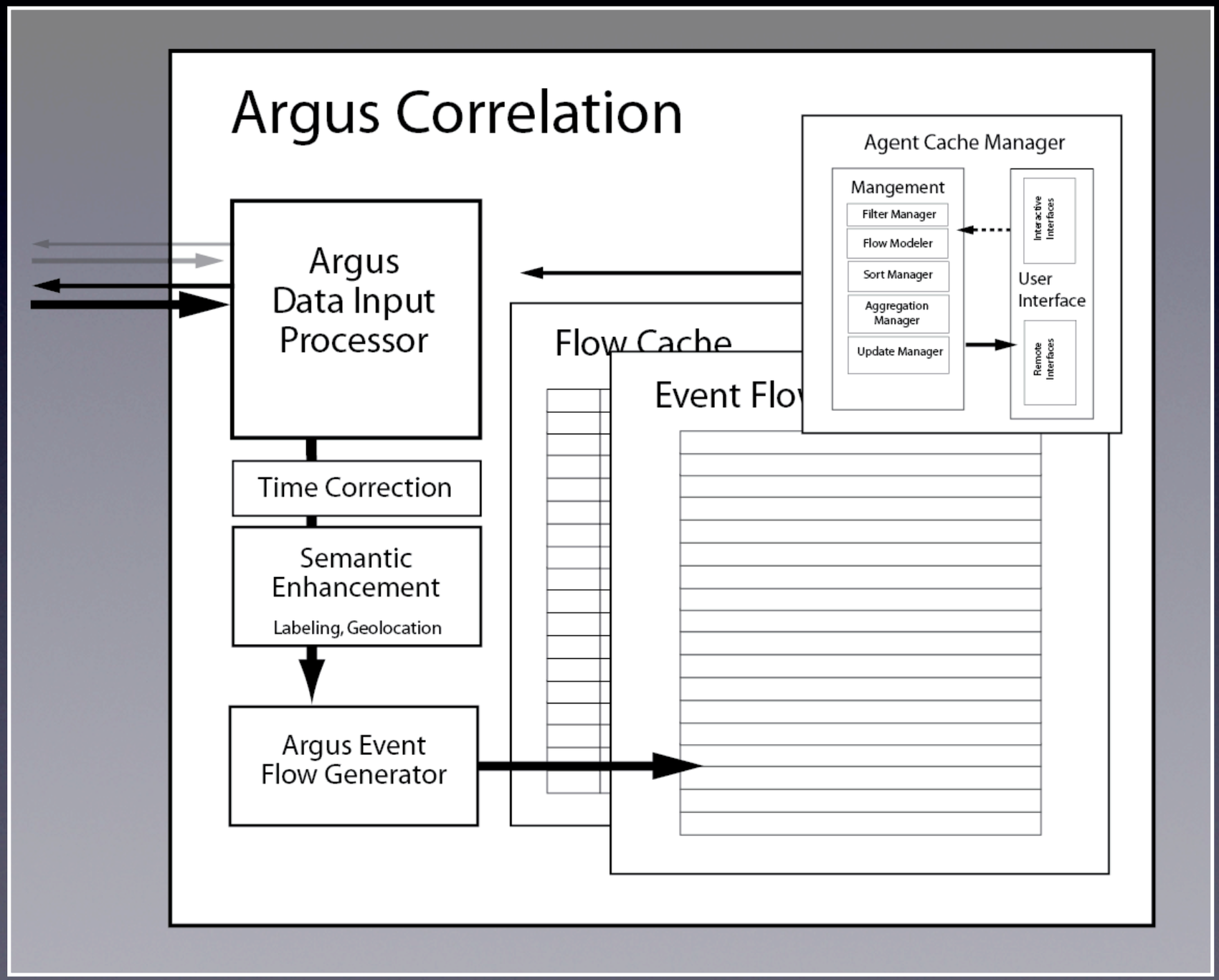
Argus Correlation Design

Radium Process



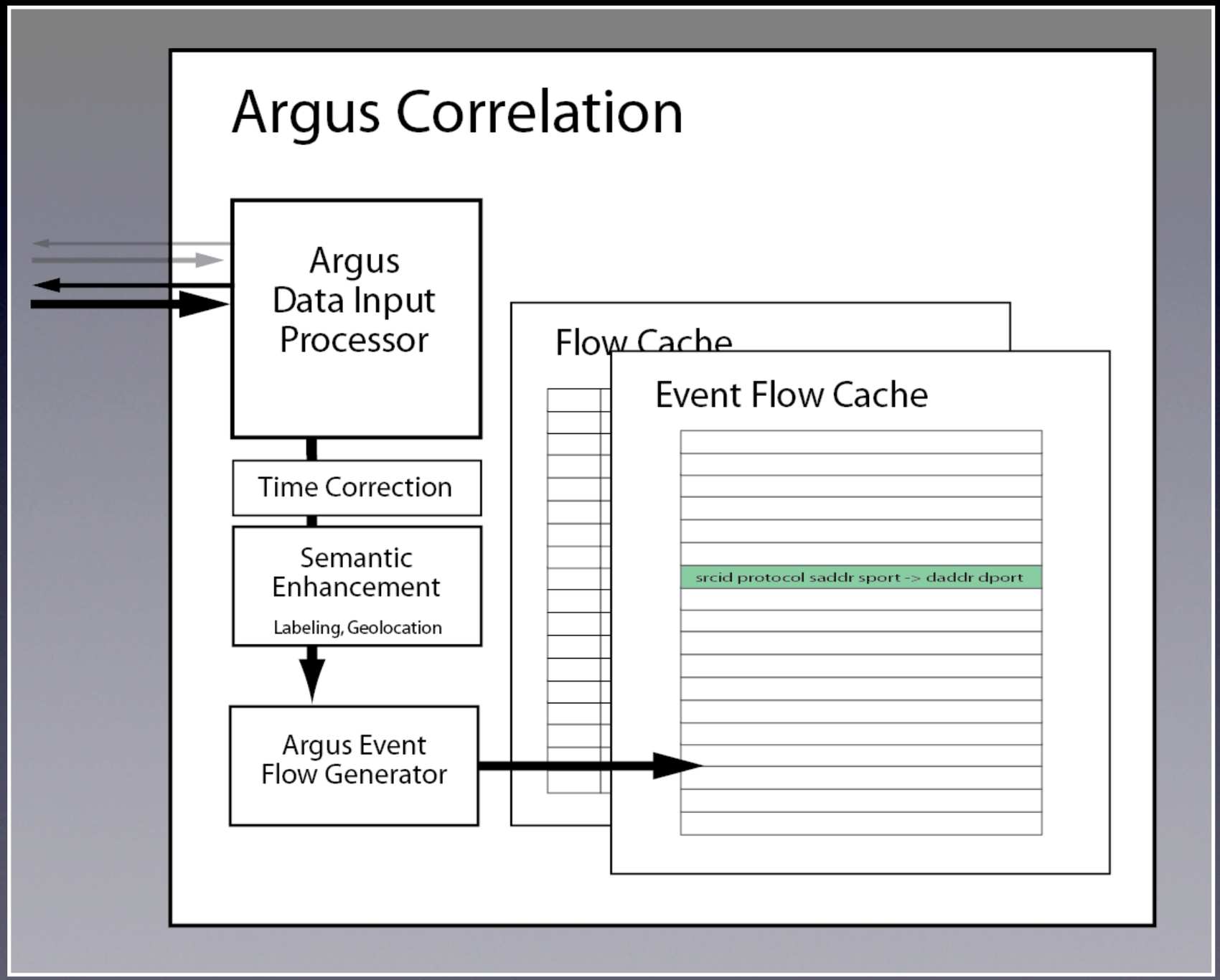
Argus Correlation Design

Radium Process



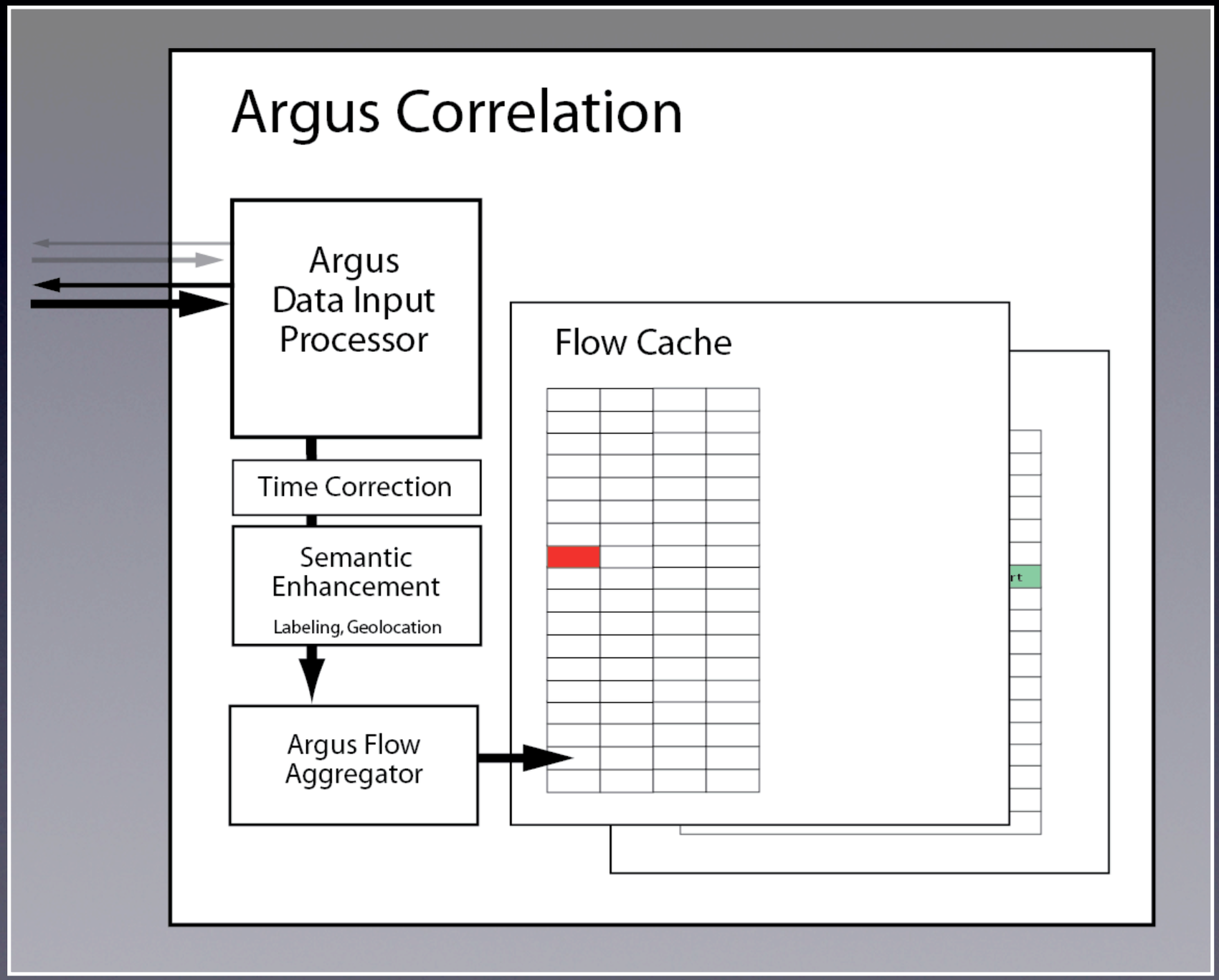
Argus Correlation Design

Radium Process



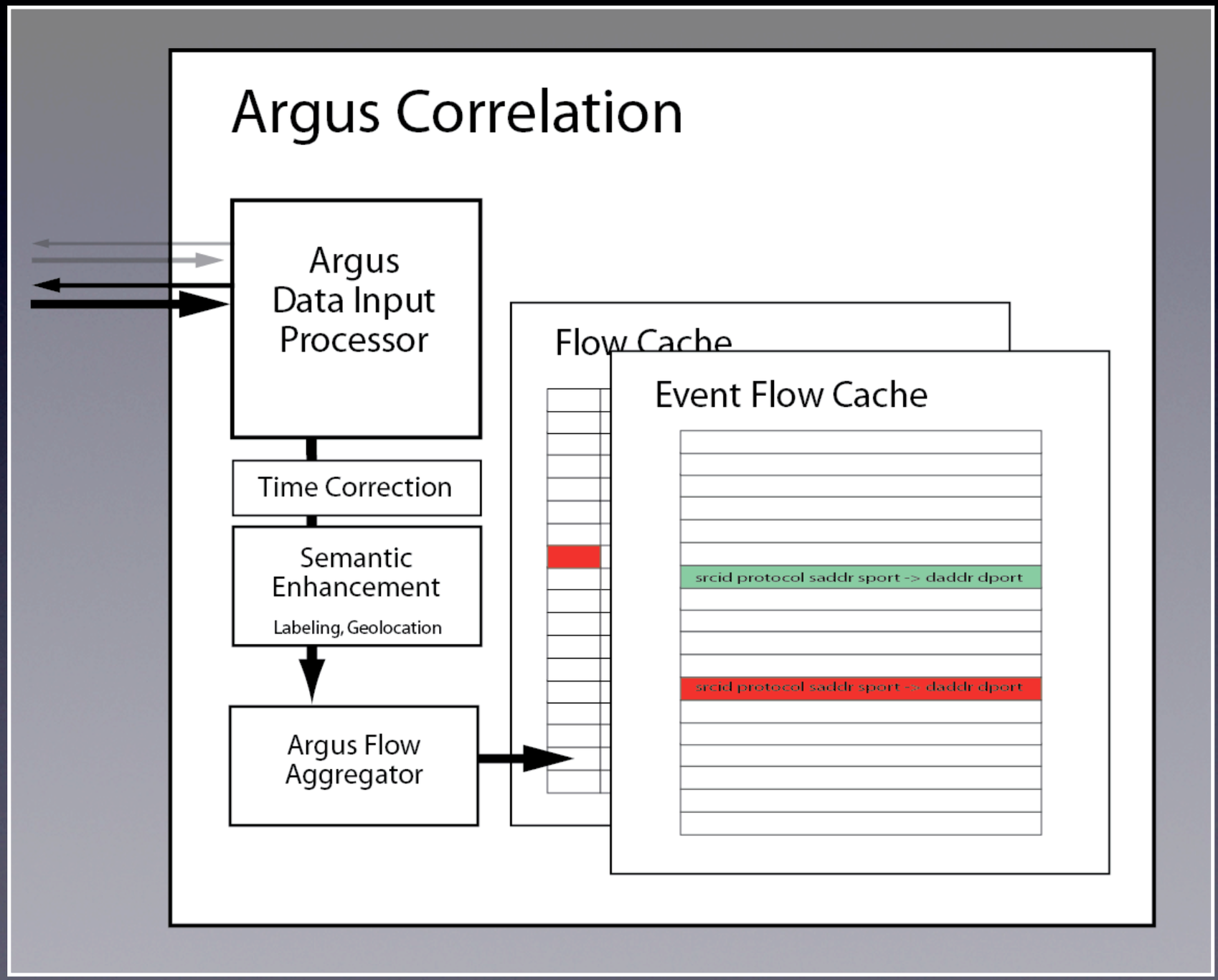
Argus Correlation Design

Radium Process



Argus Correlation Design

Radium Process



Argus Strategy

- Argus events processing generates flow descriptions and annotation labels that contain the user and the program
- We append these labels to the record.
- And then process like any other flow record

- Lot of rules on how argus labels work.

- Argus Metadata Tutorial has a lot of stuff on this topic.

Live Demonstration

from Presentation Laptop

ra and ratop screens showing live traffic as observed from the laptop

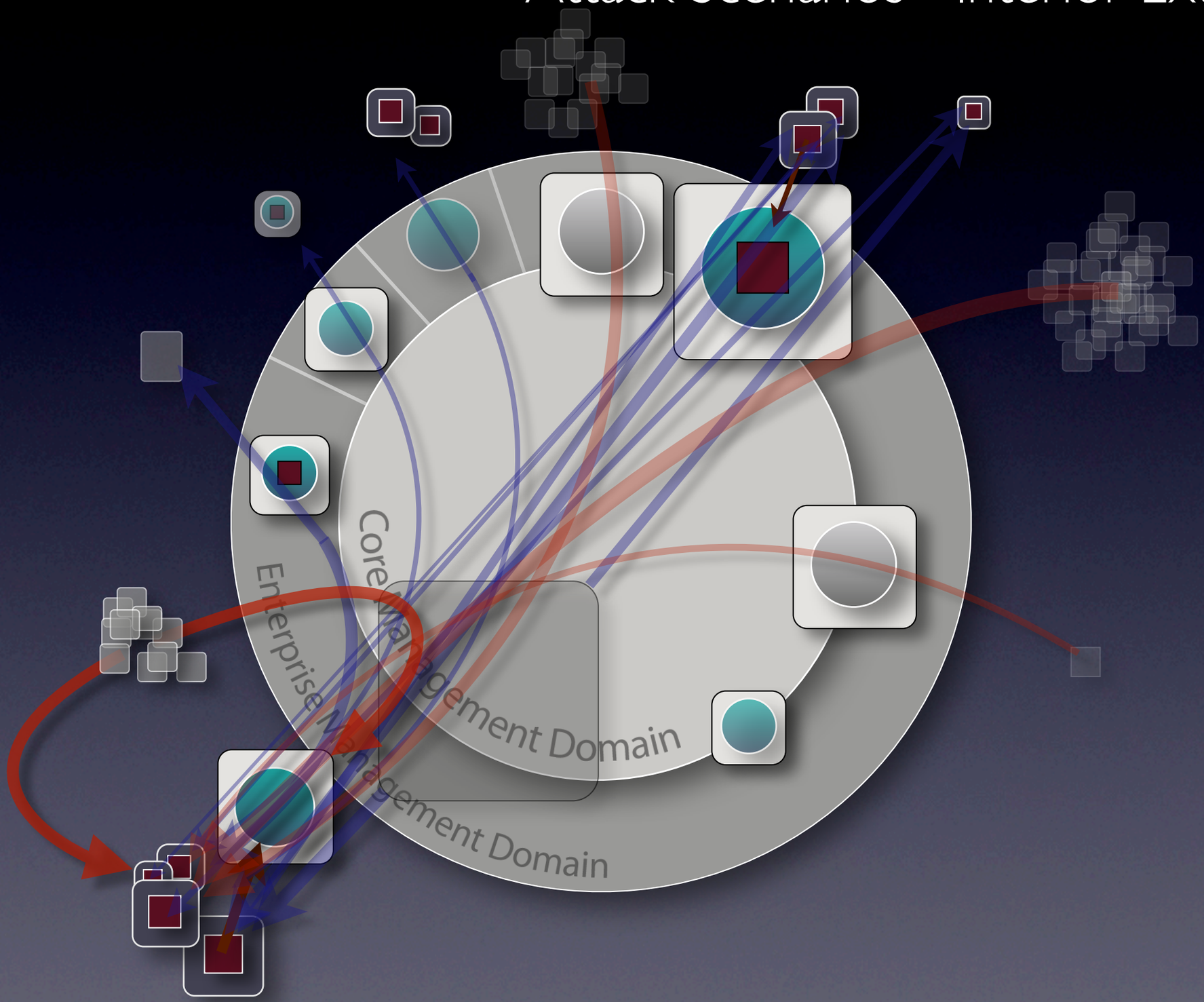
and realtime labeling of user, pid, program name

inserted into the flow record itself.

Supporting Slides

Distributed Situational Awareness

Attack Scenarios - Interior Exterior Spoofing



White/Visible Node

Black/Non-Visible Node

Data Plane

Command and Control
Attack Traffic

Spoof Correlation

- Simple multi-domain flow correlation
- However, with NAT, encryption, tunneling, traditional flow correlation is not possible.
  - No applicable flow identifiers for matching
  - Flow granularity mismatch
- Need flow metadata to make assessment
  - Content
  - Time
  - Packet dynamics (PD).
- Absence of correlation is the key
  - Statistical systems are unusable