Real Time Situational Awareness Using Argus

From Packet(s) to Alarm(s)

Carter Bullard
CEO/President

QoSient, LLC
150 E 57th Street Suite 12D
New York, New York 10022

carter@qosient.com

FloCon 2012
Austin, Texas
Jan 9, 2012

All concepts and technology presented were developed under US DoD Contract N00173-03-C-2008

Carter Bullard carter@qosient.com

- QoSient - Research and Development Company
  - US DoD, IC, DARPA, DISA
    - Very Large Scale Optimization (Operations, Performance, Security)
    - High Performance Network Security Research
    - DARPA CORONET Optical Security Architecture
  - Telecommunications / Performance Optimization
  - FBI / CALEA Data Wire-Tapping Working Group
- QoS / Security Network Management - Nortel / Bay
- QoS / Security Product Manager – FORE Systems
- CMU/SEI CERT
  - Network Intrusion Research and Analysis
  - Principal Network Security Incident Coordinator
- NFSnet Core Administrator (SURAnet)
- Standards Efforts
  - Editor of ATM Forum Security Signaling Standards, IETF Working Group(s), Internet2 Security WG, NANOG

Agenda

- Argus, argus systems and why
- Real-time Argus
  - Sensing
  - Data Collection
    - Simple collection
    - Data Flow Machine
  - Argus Repository
- Argus Clients
  - General client processing
  - Real-time processing
    - Streaming analytics
    - Visualization
    - Notification
- Building Real-time systems

Introduction to Argus

- Discuss the problem space
- Describe Argus design and implementation
- In the context of approaching some real problems

- Cyber Security
  - Insider Threat protection through Non-Repudiation
- Degradation of Service
  - Identification
  - Attribution
  - Mitigation

Argus

http://qosient.com/argus

- Argus is a network activity audit system

  Argus was officially started at the CERT-CC as a tool in incident analysis and intrusion research. It was recognized very early that Internet technology had very poor usage accountability, and Argus was a prototype project to demonstrate feasibility of network transactional auditing.

- The first realtime network flow monitor (1989)
- Top 100 security tools used in the Internet today
  - Generates detailed network resource usage logs
  - Source of historical and near realtime data for the complete incident response life cycle
- Designed to provide useful data for network
  - Operations - Service availability and operational status
  - Performance - End-to-end assessment of user traffic
  - Security - Audit / Non-Repudiation

Argus

- Real-time network flow monitor
- Network flow data collection system
- Network flow data processing
- Audit data repository tools

Argus History

- Georgia Tech (1986)

  Argus was the first data network flow system. Started at Georgia Tech, Argus was used as a real-time network operations and security management tool. Argus monitored the Morris Worm, and was instrumental in monitoring the "Legion of Doom" hacking incident.

- CERT/SEI/Carnegie Mellon University (1991)

  Argus was officially supported by the CERT as a tool in incident analysis and intrusion research. Used to catalog and annotate any packet file that was provided to the CERT in support of Incident Analysis and Coordination, it was a focal point for research in intrusion analysis and Internet security.

- Argus Open Source (1995 - Present)

  Transitioned into public domain in 1995. Supported by CMU and CERT/SEI at many levels including the current argus developers mailing list.

  Used now by a very large number of educational, commercial and governmental sites for network operations, security and performance management.

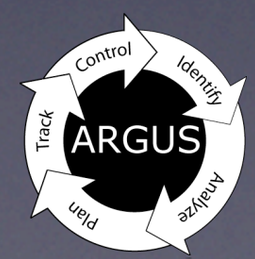
  Top 100 Security Tools worldwide

Who's using Argus?

- U.S. Government
  - DoD Performance/Security Research - Gargoyle
    - https://software.forge.mil/projects/gargoyle
    - JCTD-Large Data, CORONET, NEMO, JRAE, Millennium Challenge
  - Tactical Network Security Monitoring / Performance Analysis
    - Naval Research Laboratory (NRL), DISA, General Dynamics, IC
- Network Service Providers
  - Operational/Performance Optimization
  - Acceptable Use Policy Verification
- Educational (1000's of sites world-wide)
  - Carnegie Mellon University
  - Stanford University
  - University of Chicago
  - New York University

Enterprise wide near realtime network security audit
Distributed security monitoring
Network security research
Acceptable use policy verification

- ISPs, Enterprises, Corporations, Individuals



Network Situational Awareness

- Argus is designed to be THE network SA sensor
  - Ubiquitously deployable DPI traffic sensor
  - Comprehensive (non-statistical) traffic awareness
  - Provides engineering data, not business intelligence
    - Detailed network transactional performance
    - Network fault identification, discrimination and mitigation
      - Reachability, connectivity, availability, latency, path, flow control etc....
    - Customer gets the primitive data, not just reports/alerts
  - Near realtime and historical capabilities
  - Packet capture replacement
- Supporting a large number of SA applications
  - Advanced Network Functional Assurance (Operations)
    - End-to-End transactional performance tracking (data and control plane)
    - Network component functional assurance (NAT, reachability, encryption)
    - Policy enforcement verification/validation (Access control, path, QoS)
  - Advanced Network Optimization (Security and Performance)
    - Supports network entity and service identification, analysis, planning tracking and control, including baselining, anomaly detection, behavioral analysis and exhaustive forensics

Problem Space

US Cyber Security Focus

- US Cybersecurity focus is shifting
  - Shifting from cyberwarfare, back to cyber
- Structured around 4 basic themes
  - Designed-in Security - inherent resistance to attack
  - Tailored Trustworthy Spaces - flexible, adaptive, distributed trust
    - Focus → Wireless Mobile Networks
  - Moving Target - dynamism as a protection mechanism
    - Focus → Deep Understanding of Cyberspace
    - Focus → Nature-Inspired Solutions
  - Cyber Economic Incentives
- Supporting National Priorities
  - Health IT, Smart Grid, Financial Services, National Defense, Transportation, Trusted Identities, Cybersecurity Education

Trustworthy Cyberspace: Strategic Plan for the Federal Cybersecurity Research and Development Program, US NSTC, Dec 2011

DHS Cybersecurity Strategy

- Protecting Critical Information Infrastructure
  - Reduce Exposure to Cyber Risk
  - Ensure Priority Response and Recovery
  - Maintain Shared Situational Awareness
  - Increase Resilience
- Strengthening the Cyber Ecosystem
  - Empower Individuals and Organizations to Operate Securely
  - Make and Use More Trustworthy Infrastructure
  - Build Collaborative Communities
  - Establish Transparent Processes
- Strategy refers to real-time and near real-time mechanisms
  - "... to collect and exchange information in real-time ..." - situational awareness
  - "... capabilities will be communicated in near real-time ..." - resilience
  - "... near real-time machine-to-machine coordination ..." - strengthening
  - "... acting collectively in near real-time to anticipate ..." - collaboration

The Cybersecurity Strategy for the Homeland Security Enterprise, US Department of Homeland Security, Nov 2011

[Near] Real-time Awareness

- Protecting Critical Information Infrastructure
  - Reduce Exposure to Cyber Risk
  - Ensure Priority Response and Recovery
  - Maintain Shared Situational Awareness
  - Increase Resilience
- Strengthening the Cyber Ecosystem
  - Empower Individuals and Organizations to Operate Securely
  - Make and Use More Trustworthy Infrastructure
  - Build Collaborative Communities
  - Establish Transparent Processes
- Approach specifies real-time and near real-time mechanisms
  - "... to collect and exchange information in real-time ..." - situational awareness
  - "... capabilities will be communicated in near real-time ..." - resilience
  - "... near real-time machine-to-machine coordination ..." - strengthening
  - "... acting collectively in near real-time to anticipate ..." - collaboration

The Cybersecurity Strategy for the Homeland Security Enterprise, US Department of Homeland Security, Nov 2011

Theoretical Security Threats and Countermeasures

Countermeasures		Threat				
		Unauthorized			Degradation of Service	Repudiation
		Use	Modification	Disclosure		
Authentication	Cryptographic	x		x		
Integrity	Cryptographic		x			
Confidentiality	Cryptographic			x		
Access Control		x	x	x	x	
Non-Repudiation (audit)		x	x	x	x	x

Derived from ITU-T Recommendation X.805
Security Architecture for Systems Providing End-to-End Communications

	Primary Security Countermeasure
	Secondary Security Countermeasure

Non-Repudiation

- Most misunderstood countermeasure *
  - ITU-T Recommendation X.805 security dimension
    - *Prevent ability to deny that a network activity occurred*
- Principal source of true deterrence
  - Non-repudiation provides comprehensive accountability
  - Creates concept that you can get caught

- Argus approach to network non-repudiation
  - Generate data to account for all network activity
    - Comprehensive Network Transactional Audit
    - Mechanism specified by DoD in NCSC-TG-005
      - The Red Book - Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria (1987)
  - Focus on all X.805 Security Planes
    - User, Control and Management network activity

✳ Crypto-technical redefinition of non-repudiation by Adrian McCullagh in 2000 to apply only to digital signatures has created a great deal of confusion. While you can have repudiation of a signature, it's not the only thing you can repudiate.

Non-Repudiation Concepts

ITU X.813

Information Technology

Open Systems Interconnection

Security Frameworks in Open Systems: Non-repudiation Framework

"The Non-repudiation service involves the generation, verification and recording of evidence. ..... Disputes cannot be resolved unless the evidence has been previously recorded."

The service provides the following facilities which can be used in the event of an attempted repudiation:

- generation of evidence
- recording of evidence
- verification of generated evidence
- retrieval and re-verification of the evidence

Why Non-Repudiation?

- When it exists and structured well, you get
  - Effective information for incident response
    - Fundamental ground truth - if its not there, it didn't happen
    - Classical forensics support
    - Evidence suitable for criminal and civil complaints
  - Enhanced network situational awareness
    - Network Service Behavioral Baselining
      - Who is really using my DNS servers?
      - What is generating Email in my enterprise?
      - How much data did he transmit last night?
    - Network Policy Enforcement Assurance
      - Are my IPS / IDS / Firewall protections working?
    - Network Fault Attribution
      - Is it an attack? Is it real? Is it a bug? Is it Fred?
- Enables enhanced analytics, simulation and 'what if' analysis
  - This host polls this email server every 60.0023 +/- 0.0004231 seconds and has been doing that for 17.6243 months, with only 27 outages lasting .....
  - Will this new access control policy, break anything?

Achieving Non-Repudiation

- Comprehensive Activity Accountability
  - Complete Activity Sensing and Reporting
  - Develop Information System with Formal Properties
    - Fundamental ground truth (if its not there, it didn't happen)
- Accurate and Efficient Activity Representation(s)
  - Stored data must represent actual activity
    - Attribute verifiability
      Must be unambiguous with regard to object identification
      Must have a relational algebraic correctness
    - Time synchronization and precision
      Must convey correct order of events
- Fundamental Data Utility
  - Formal and Mature Data Model
  - Useful Data Availability Properties
  - Effective Storage and Retention Strategies

Real world issues

- Non-Repudiation systems must support addressing real world issues
  - Must capture adequate forensics data for incident response
    - Enterprise focused on contemporary security issues
    - Policy enforcement verification validation
    - Provide high level of semantic capture/preservation
    - Support complex behavioral analysis through packet dynamic awareness
  - Should support real time awareness
    - Data presence information - access control verification
    - Contribute large scale multi-level hierarchical distributed situational awareness
- Provide real deterrence
- In a perfect world, you would have a single source for all your network forensics data
  - Support near real-time and historical requirements
  - FISMA continuous network monitoring role

Real world issues

- Incident Response
  - NASA calls. One of your machines attacked a satellite launch
    - Very important military mission
    - Concerned that you may have done it on purpose.
    - Cost the US Gov't $357M
    - 7.5 months ago
    - FBI is coming over in a few minutes
- In a perfect world, you would
  - Review enterprise network activity audit logs as first step
    - Single location for entire enterprises network logs
      - Query for any activity to NASA network or host
      - Pinpoints local hosts involved
      - Now begins the forensics examination
        - Was the attacking machine broken into?
        - If so, (hope so), where did it come from?
    - With multiple internal non-repudiation systems
      - You should be able to identify external / internal attack progression
        - Attack methodologies
        - Identify stepping-stone hosts

Real world issues

- Xerox machines intellectual property loss
  - News story reveals problems with Xerox machines
    - Photocopy machines don't delete copy images
    - Hospitals have lots and lots of Xerox machines
  - What can you do?
  - With single enterprise border non-repudiation system
    - You would know if anyone from the outside ever discovered your Xerox machines in a scan
    - You would know if anything directly accessed your Xerox machines from the outside
  - With non-repudiation system at the Xerox LAN border
    - You would have logs of all network accesses to machine
    - You would know which accesses extracted data rather than presented data to the printer
    - You would have the content visibility needed to identify what images were extracted.

Real world issues

- Intrusion Detection Behavioral Anomalies
  - Access from user X to supercomputer A account
    - Authenticated, acceptable
    - No apparent system log deviations
    - But came from a host outside the normal COI
    - Human analyst noticed the network inconsistency
    - User was on vacation
    - First indication of significant US Gov't problem with Stakkato

Real world issues

- Unintended/Unexpected data exposure
  - Symptom - Poor application performance
    - Database application exhibiting very poor performance
    - Each transaction taking 0.3-0.4 seconds to complete.
    - All software components running on a single machine
    - Absolutely no clues from debugging information
    - Wasn't this bad last week
    - Very, very, very sensitive medical information
  - Network flow monitoring revealed problem
    - All IPC messaging was being transmitted onto the network
      - Data was being transmitted to the internal software process using network
      - Network turned it back around, after it left the LAN
    - One software component poorly configured
      - Using server's external name (NAT'ed environment)
    - Very, very, very, very bad

Degradation of Service

- A primary design goal of Argus is DoS identification
  - Argus used in DDoS research papers (1996-2010)
  - CERT Advisory CA-1996-01 UDP Port Denial of Service
  - Many commercial DDoS products are flow data based
- Degradation is an attack on Quality of Service
  - QoS sensitive situational awareness is critical
    - QoS anomaly detection
    - QoS fault management
    - QoS intentional assignments
  - DoS protection really needs to be a part of QoS optimization
    - Can't discriminate QoS degradation when there is poor QoS
- Argus data specifically designed to support:
  - QoS Fault identification/discrimination/mitigation/recovery
    - Pre fault QoS Characterization and Optimization
    - Realtime fault detection and QoS anomaly characterization
    - Post fault recovery, forensics and impact assessments
    - Formal QoS optimization processes

Security and Performance

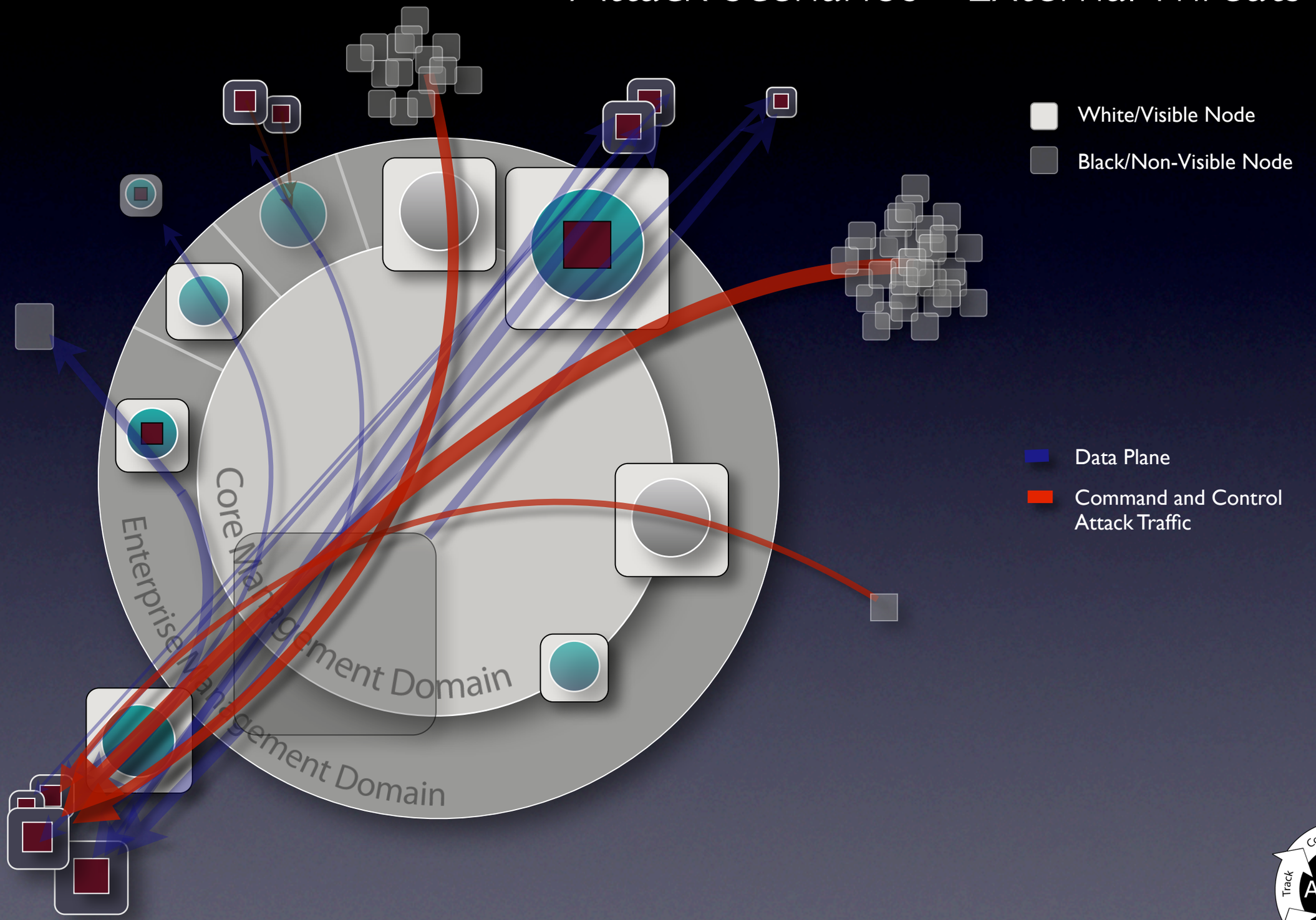
- Security and performance are tightly coupled concepts
  - Network performance is an asset that needs protection
    - DoD GIG Information availability assurance (DoDD 8500.1)
    - Commercial product delivery dependent on network performance
    - Performance is being specifically attacked
  - Security and performance contribute directly to QoS
  - Security and performance are both optimizations
    - Many times at odds with each other
- Performance awareness data is security awareness data
  - Presence with identifying information is much of the forensics story
- Performance as a leading security indicator
  - Exfiltration and spam generation consume resources
  - Classic "man in the middle" and "traffic diversion" detection
    - Scenarios create measurable end-to-end performance impacts
  - [D]DoS detection is a performance anomaly problem

Degradation of Service (cont)

- QoS Fault Discrimination
  - Traditional QoS fault detection and mitigation
    - End-to-End oriented QoS tracking capability
      - Availability, demands, path, latency and efficiency modifications
      - Host vs Network QoS impact discrimination
      - Distributed sensor strategies provide best "finger pointing" capabilities
    - Historical audit provides baseline analytics for boundary tests
    - Discrimination can involve session dependency analysis
      - Front end network service dependancies
        - ARP, DNS, IP reachability, TCP availability, Service
      - Back end service dependency awareness
  - Discriminating intentional QoS failure
    - Protocol vulnerability exploitations
    - Exclusionary methods for attack designation
      - Flash crowd vs DDoS
      - Indirect attack assessment support

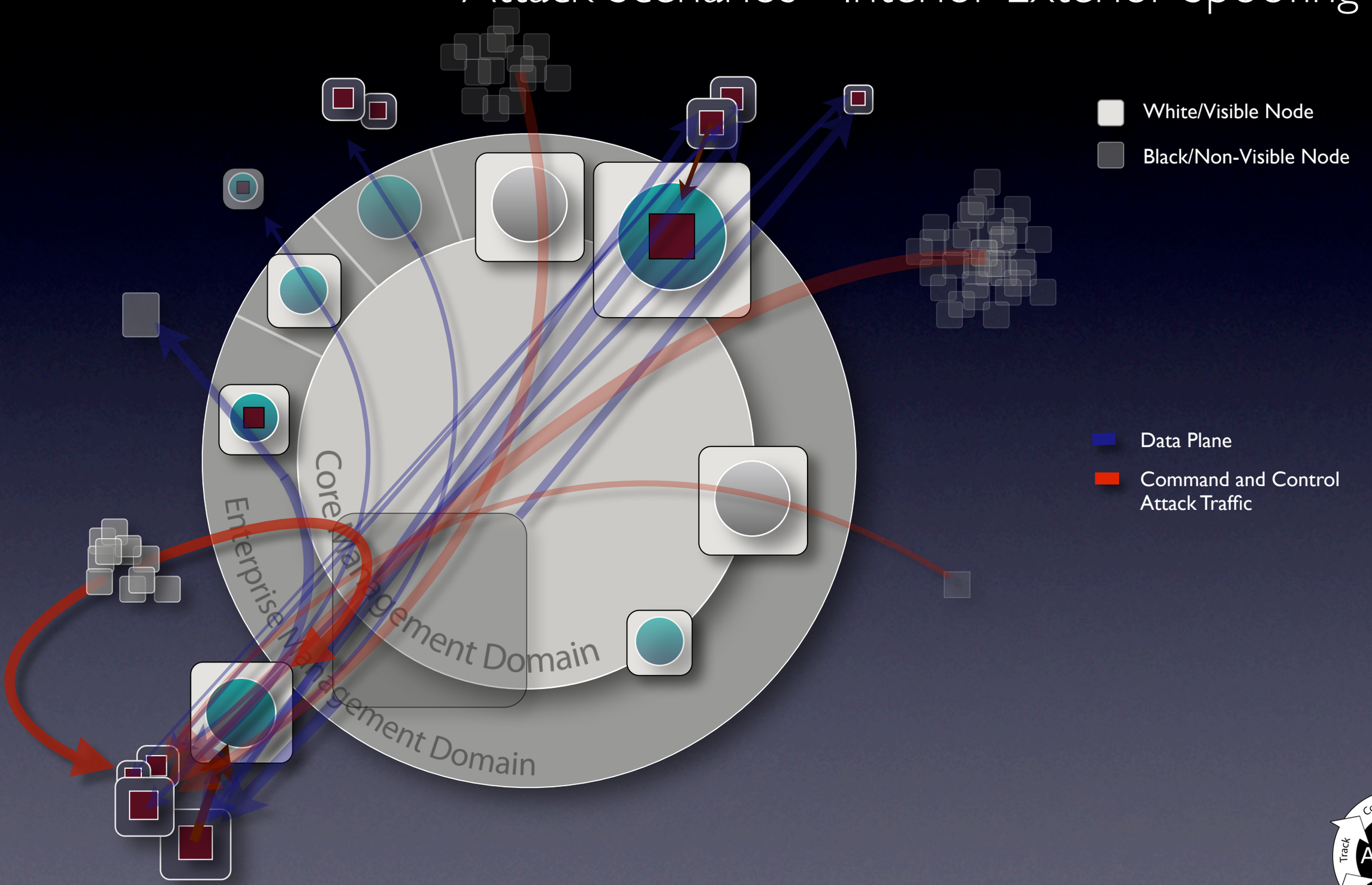
Distributed Situational Awareness

Attack Scenarios - External Threats



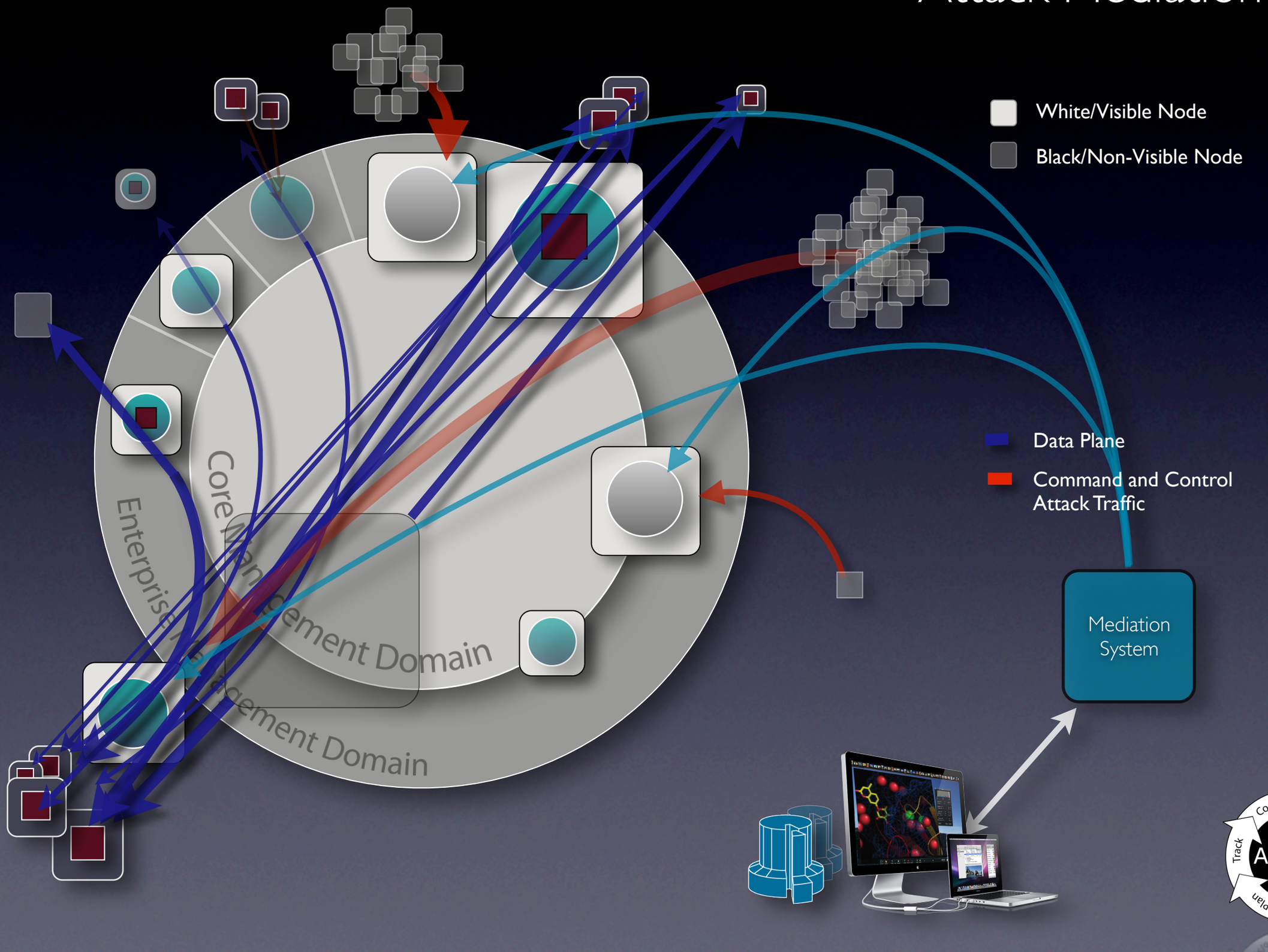
Distributed Situational Awareness

Attack Scenarios - Interior Exterior Spoofing



Distributed Situational Awareness

Attack Mediation



Degradation of Service (cont)

- Methods used to defeat [D]DoS mitigation
  - Mitigation involves denying access from list of exploit IP addresses
  - IP address spoofing
    - Host along attack path emulates [D]DoS traffic
      - Internal host that can "see" the target can forge 100,000's of simultaneous active connections to/from foreign hosts
    - Routing mediated address spoofing
      - BGP modifications allow near local networks to spoof address space
      - Internal modification to locally support foreign address space
        - Static routes can be setup so that "China" is routed to port 23b
        - Control plane attacks (ARP, RIP, OSPF) to advertise "China" is over here
- Result is that you just can't seem to shake the attack
- Distributed sensing detects this scenario
- Net-spatial data and active traceback strategies

Degradation of Service (cont)

- QoS Fault Mediation
  - Argus can provide information for effective mediation
  - Provide realtime forensics for threat analysis
    - Realize that QoS of critical assets are being affected
    - Provide real-time list of active nodes
    - For web attacks provide recurring URL visits
  - Provide CIDR addresses to block
    - Need to be sensitive to ACL limits of network equipment
    - Need to be cleaver when trying to block 50K IP addresses
  - Provide CIDR addresses to allow
    - Historical Community of Interest (COI) for allowable customers
    - The list of networks active at the initial time of attack
- Argus information to assure mediation worked
  - Network now performing within SLA
  - Track conditions to indicate when to revert, if ever

Degradation of Service (cont)

- Methods used to defeat [D]DoS mitigation
  - Mitigation involves denying access from list of exploit IP addresses
  - IP address spoofing
    - Host along attack path emulates [D]DoS traffic
      - Internal host that can "see" the target can forge 100,000's of simultaneous active connections to/from foreign hosts
    - Routing mediated address spoofing
      - BGP modifications allow near local networks to spoof address space
      - Internal modification to locally support foreign address space
        - Static routes can be setup so that "China" is routed to port 23b
        - Control plane attacks (ARP, RIP, OSPF) to advertise "China" is over here
- Result is that you just can't seem to shake the attack
- Distributed sensing detects this scenario
- Net-spatial data and active traceback strategies

Formal Non-Repudiation Systems

J-STD-025A

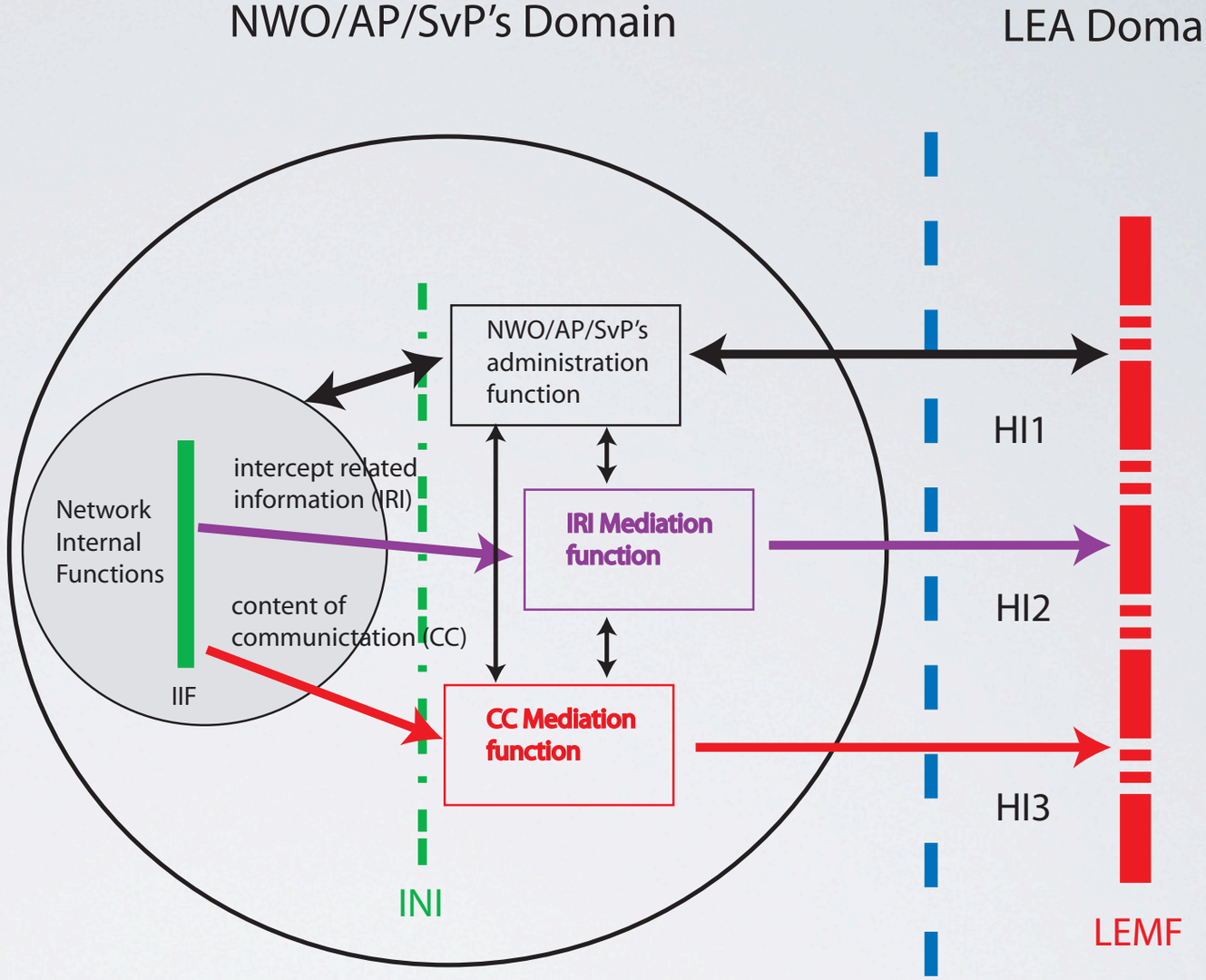
WAI/GT/FuncSpecs v1.0.1 (2000-06)

- Telephone Billing Records (retrospective)
- J-STD-025A / ETSI TS 101 671 (prospective)
  - Dialed Number Recorder (DNR/Pen Register)
  - Full Audio Interception (Title III/FISA)
- When concepts applied to data networks:
  - Content capture unencrypted (keys)
  - Information Protection Requirements
  - Geo-Location Information
  - Time Constraints
  - Unchanged State of Service

ETSI ES 201 671

Telecommunications Security

Lawful Interception(LI); Handover interface for the lawful interception of telecommunications traffic



LI handover interface HI

IIF: internal interception function
INI: internal network interface

HI1: administrative infomration
HI2: intercept related information
HI3: content of communication

NOTE 1: Figure 1 shows only a reference configuration, with a logical reprsentation of the entities involved in lawful interception and does not mandate separate physical entities.

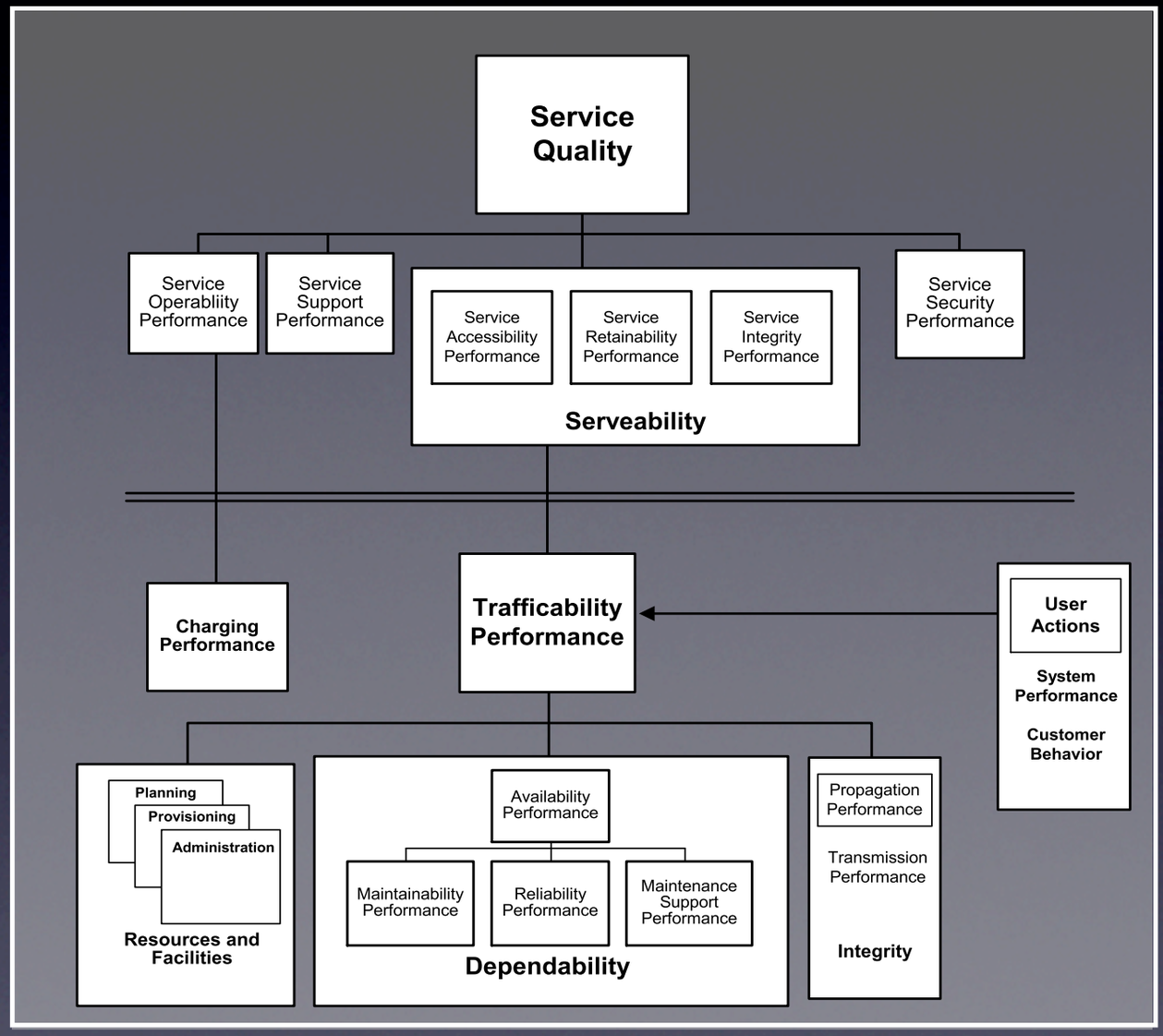
NOTE 2: The mediation functions may be transparent.

Functional Block Diagram Showing Handover Interface HI

ETSI ES 201 671 V2.1.1 (2001-09) Telecommunications security; Lawful Interception(LI); Handover interface for the lawful interception o ftelecommunications traffic

What Are CDRs Used For?

- Billing
- Traffic Engineering
- Network Management
- Maintenance
- Marketing
- Product Development
- Security
  - Fraud Detection
  - Forensics Analysis
  - Incident Response
  - Non-Repudiation / Audit



From ITU-T Recommendation E.800 Quality of Service, Network Management and Traffic Engineering

Network Auditing

- Specified by DoD in NCSC-TG-005
  - The Red Book - Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria (1987)
- Goal to provide Non-Repudiation
  - Comprehensive audits accounting for all network use
  - Creates **real deterrence** in formal systems
    - Fear of getting caught is extremely powerful
  - Utility comes from the quality of collected information
- Internet network transaction auditing is emerging
  - Started at the CMU CERT-CC in early 1990's - Argus
  - Directly modeled after the PSTN CDR
  - Aspects of IP network auditing are being standardized

Achieving Non-Repudiation

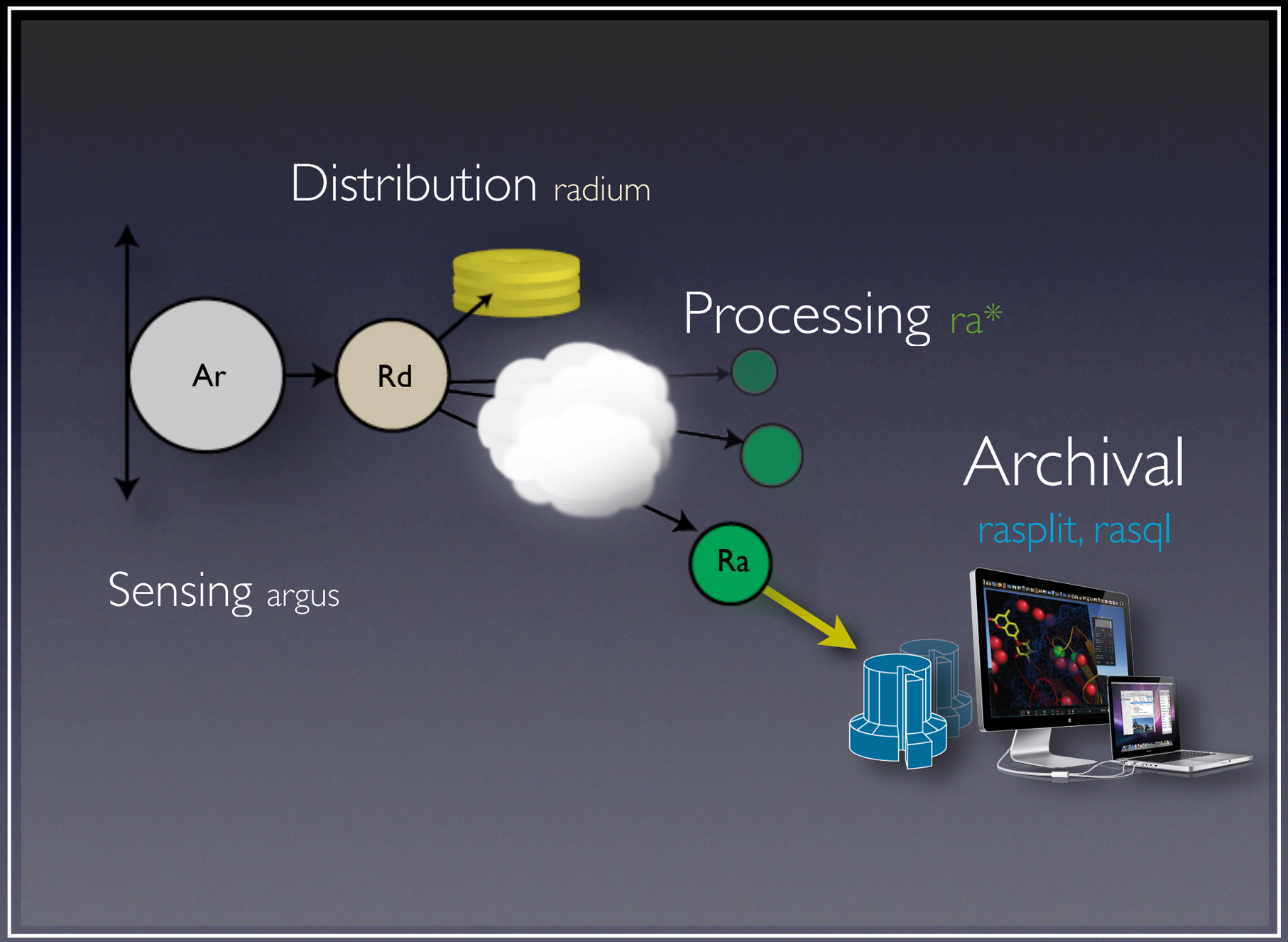
- Comprehensive Activity Accountability
  - Complete Activity Sensing and Reporting
  - Develop Information System with Formal Properties
    - Fundamental ground truth (if its not there, it didn't happen)
- Accurate and Efficient Activity Representation(s)
  - Stored data must represent actual activity
    - Attribute verifiability
      - Must be unambiguous with regard to object identification
      - Must have a relational algebraic correctness
    - Time synchronization and precision
      - Must convey correct order of events
- Fundamental Data Utility
  - Formal and Mature Data Model
  - Useful Data Availability Properties
  - Effective Storage and Retention Strategies

Comprehensive Accountability

- Account for all network activity
  - Because any network activity can be associated with a cyber-security activity
    - Generally, if you aren't looking 'there', 'there' is where they will be
    - Hidden variables enable the adversary
  - Observation scope must be relevant
    - Utility of collected information should be very high
    - Using PSTN as guide, ISP can collect anything, but share nothing.
- Argus approach to network non-repudiation
  - Generate data to account for all network activity
    - Comprehensive Network Transactional Audit
    - Mechanism specified by DoD in NCSC-TG-005
      - The Red Book - Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria (1987)
  - Focus on all X.805 Security Planes
    - User, Control and Management network activity

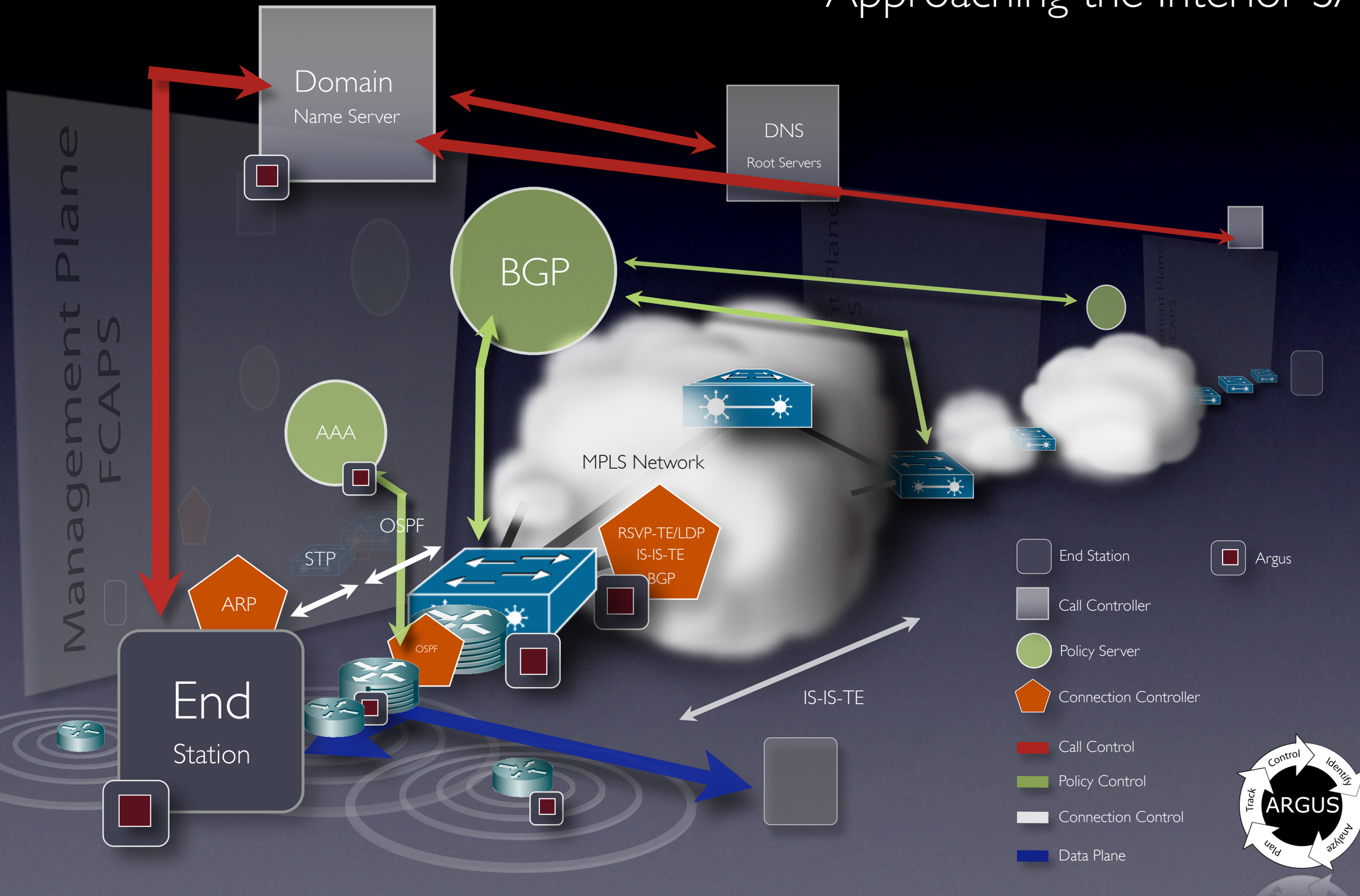
Real-Time Argus

Argus System Design



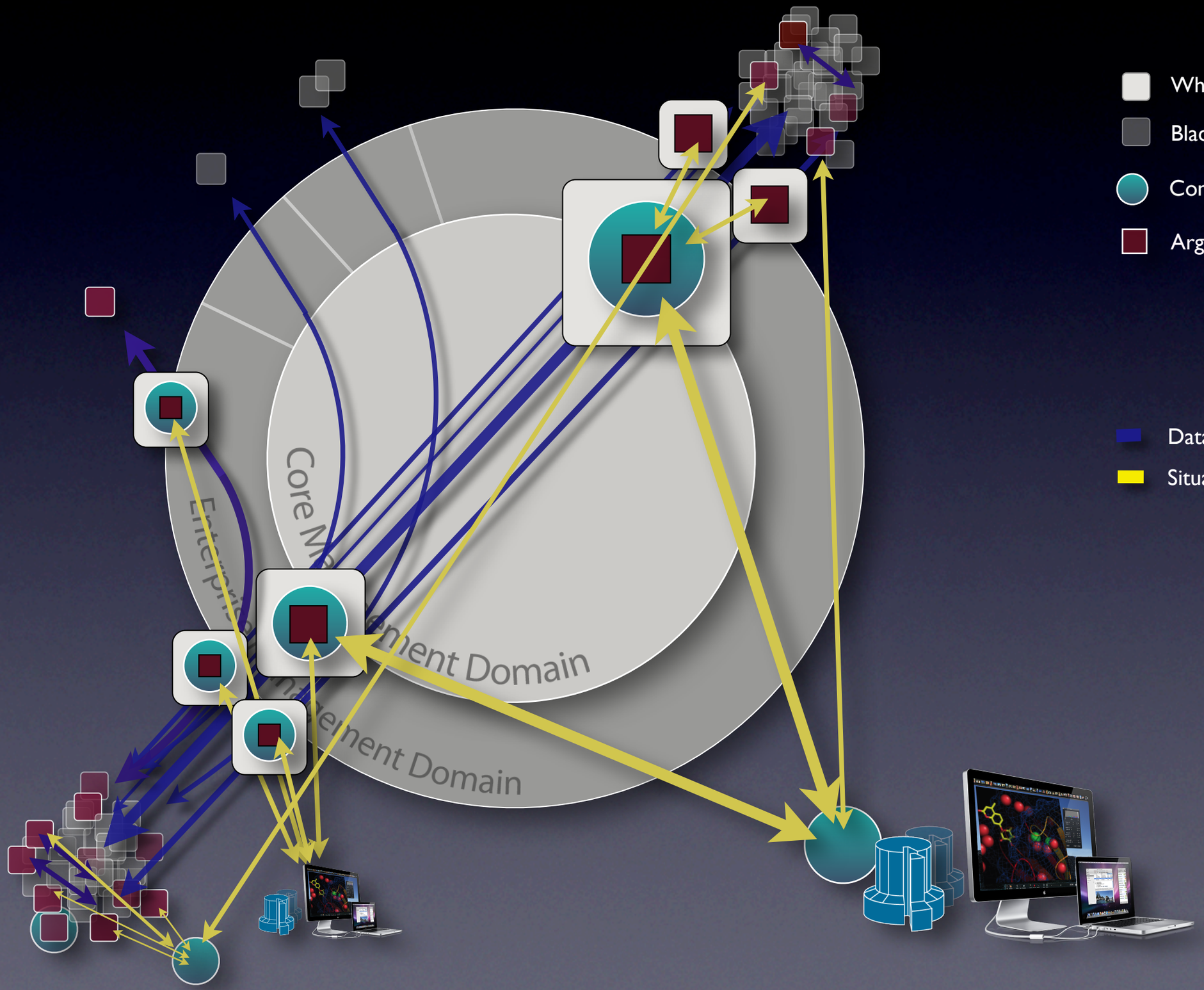
Comprehensive Enterprise Awareness

Approaching the Interior SA



Complex Comprehensive Awareness

Local and Remote Strategies



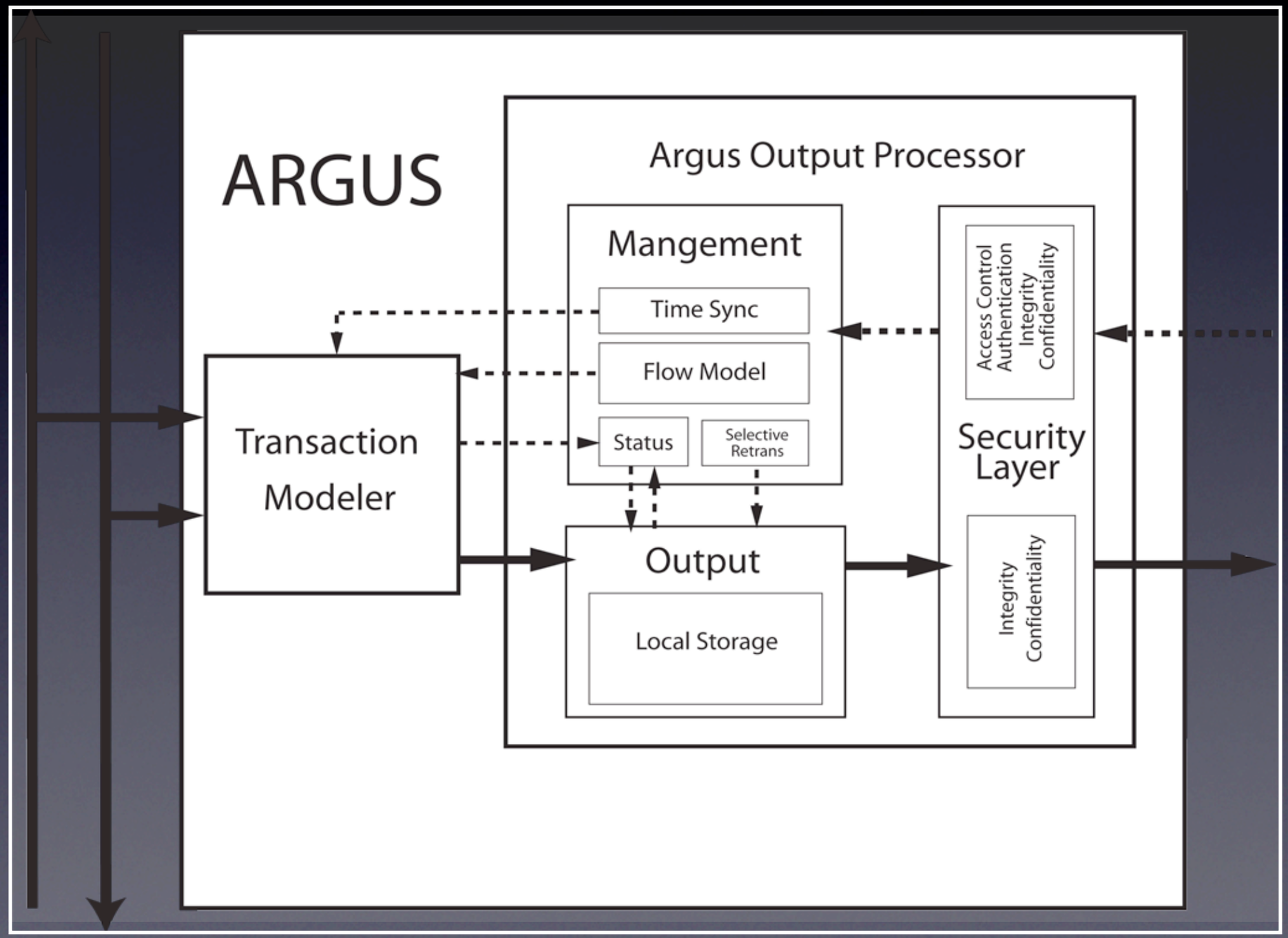
Sensing
Argus Data Generation

Argus Data Generation

- Packets to Flows
- Getting Started with Argus
- Argus Deployment
- Configuration
- Running Argus

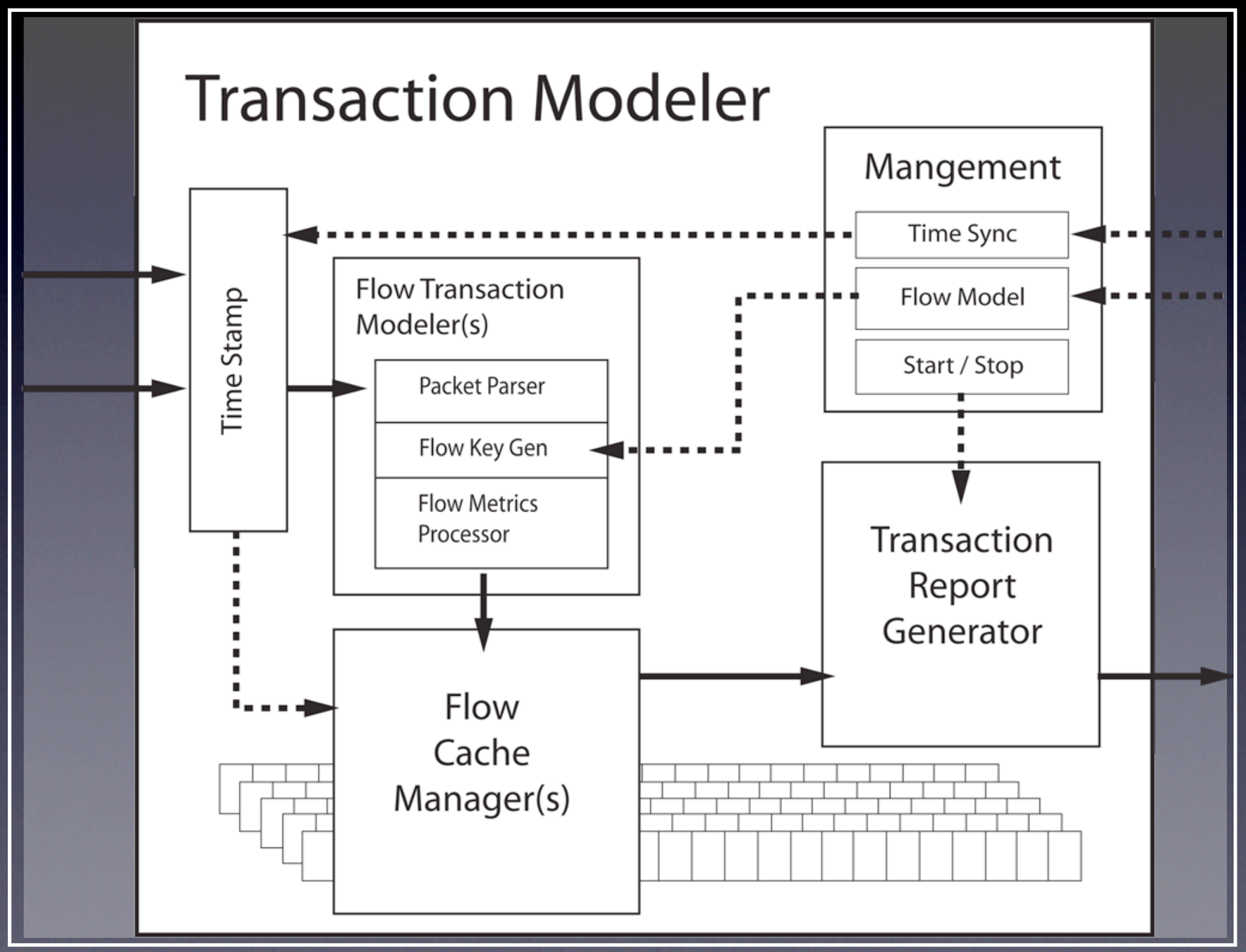
Argus Sensor Design

Packets to Flows



Argus Sensor Design

Transactional Processor



Network Flow Information

- All types contain IP addresses, network service identifiers, starting time, duration and some usage metrics, such as number of bytes transmitted.
- More advanced types are transactional, convey network status and treatment information, service identification, performance data, geo-spatial and net-spatial information, control plane information, and extended service content.

- Available IP Flow Information
  - Argus
    - Control and Data Plane network forensics auditing
    - Archive, file, stream formats. (Binary, SQL, CSV, XML)
  - YAF/SiLK - CERT-CC (IP data only)
    - Designed for Cyber security forensics analysis
    - IETF IPFIX stream formats. Binary file format.
  - IPDR - Billing and Usage Accountability (IP data only)
    - ATIS, ANSI, CableLabs, SCTE, 3GPP, Java CP, ITU/NGN
    - File and stream formats (XML).
  - Netflow, JFlow, Sflow (IP data only)
    - Integrated network vendor flow information - statistical/sampled
    - Used primarily for router operations, network management

Packets to Flows

- Packet Timestamping
  - Methodology, Time Synchronization and resolution
- Packet Header Parser
  - Multiple flow tracking strategies determines parser
  - Supports OSI, IEEE, IP and Infiniband packet formats
  - Innermost Layer 3 target header (service layer)
    - Complex encapsulation stacking
      - L2 -> L3 -> L2 -> L3 -> L4 -> L2 -> L3
      - Support protocol discovery
  - Limited by packet snap size
    - Argus supports complex packet capture support
    - Privacy issues
    - Control plane vs data plane parsing

Packets to Flows

- Flow Key Generation
  - All packets are classified into a flow of some kind
  - Argus supports 14 fundamental flow types
    - Not protocols, flow types (P, P1-P2, Multicast/Unicast, etc....)
    - Bi-directional support for all flow types (when they exist)
    - Bi-direction flow keys for all supported encapsulations
- Flow Key is "key" to all flow tracking
- One packet one flow rule
  - Simplify flow machine call structure
  - Control plane is the bending of the rule
    - ICMP packet accounted for in ICMP flow
    - ICMP state mapped to flow identified in contents

Packets to Flows

- Flow Metrics Processor
  - Metric and attribute generation
    - Some metrics can be derived from packet itself
      - Packet size, application demand, reachability
    - Others require state
      - connectivity, availability, RTT, rate, loss, jitter, size distribution
  - Flow attribute (re)assignments
    - Flow state machine tracking
    - Dynamic attribute tracking
- Flow Cache Manager
  - Controls reporting of flow status
  - Controls dynamic flow redefinitions/reassignments

Getting Started

- http://qosient.com/argus
- ‘Using Argus’ and ‘Getting Argus’ Links
- Argus documentation
  - Man pages provided in distribution
  - HOW-TO and FAQ on the web site.
  - Argus developers mailing list
    - argus-info@lists.andrew.cmu.edu.
    - Most questions are answered here
    - Email carter@qosient.com

Getting Argus

- http://qosient.com/argus/downloads.htm
- Current stable version is argus-3.0.4
- Provided as tarball source package
- Ported to 27 platforms
  - Linux, xBSDs, Mac OS X, Windows, HPUX, Solaris, VxWorks, AIX, OpenWRT, Tilera
- Depends on:
  - libpcap - http://tcpdump.org/release
  - flex - http://flex.sourceforge.net
  - bison - http://www.gnu.org/software/bison

Making Argus

- Simple installation
  - ./configure; make
- Complex environments
  - Read ./README and ./INSTALL
  - Cygwin/OpenWRT
- Support standard autoconf options
  - ./configure --help
  - Common variations
    - prefix=/your/destination/directory
    - SASL Support
    - Native compiler options

Installing Argus

- Simple installation
  - make install
- ./INSTALL describes some complex examples
- /etc/argus.conf
- System startup configuration
  - Linux chkconfig.1 support
  - MacOS X /Library/LaunchDaemons support
- RPM support - ./lib/argus.spec

Argus Configuration

Configuration

- argus.conf
  - Running Environment
  - Monitor Characteristics
  - Flow Data Metrics
  - Security Mechanisms

Configuration
Running Environment

- Argus Daemon
- Argus Access Port
- Argus Bind IP Address
- Argus Interface
- Argus Go Promiscuous
- Argus Collector
- Argus Chroot Directory
- Argus Set User ID / Group ID
- Argus Output File
- Argus Set PID & PID Path
- Argus Packet Capture File
- Argus Environment Variables

Configuration
Monitor Characteristics

- Argus Monitor ID
- Argus Flow Status Interval
- Argus MAR Status Interval
- Argus Debug Level
- Argus Filter Optimizer
- Argus Self Synchronize
- Argus Event Data

Configuration
Flow Data Metrics

- Argus Flow Type
- Argus Flow Key
- Argus Generate Response Time
- Argus Generate Packet Size
- Argus Generate Jitter Data
- Argus Generate MAC Data
- Argus Generate Application Byte Metrics
- Argus Generate TCP Performance Metrics
- Argus Generate Bi-Directional Time Stamps
- Argus Capture Data Length
- Argus Tunnel Discovery
- Argus Keystroke

Configuration
Security Mechanisms

- Argus support the use of SASL to provide strong authentication and confidentiality protection.
- The policy that argus uses is controlled through the use of a minimum and maximum allowable protection strength. Very SASL specific.
  - RA_MIN_SSF
    - This is the minimum security strength factor for the connection. An SSF of 0 allows for no protection. An SSF of 1 will supply integrity protection without privacy.
  - RA_MAX_SSF
    - The MAX_SSF is normally used to specify the strength of encryption. 56, as an example, specifies 56-bit DES. This value should not be less than the MIN_SSF.

Configuration
Very Common Problems

- Can't start argus
  - Permissions (interface/filesystem)
    - run as root
- Can't connect to running Argus
  - Tcp_wrappers getting in the way
    - check syslog()
- Argus closes connection after a while
  - Client doesn't read data fast enough
    - improve resources between argus and client

Configuration

- Support for Real Time Operation
  - ARGUS_FLOW_STATUS_INTERVAL
    - This single variable specifies time before sensor reports on flow activity
    - Should be set to 5-30 seconds for near real-time operation
    - Value set to <= 1 second for real-time operation
  - ARGUS_MAR_STATUS_INTERVAL
    - Used as sensor health status indicator
    - Should be adjusted to detect probe failure in operating time domain
- Obsoleted
  - ARGUS_GENERATE_START_RECORD
    - Generated out-of-order output stream
    - Cost of increased data load exceeded awareness benefits

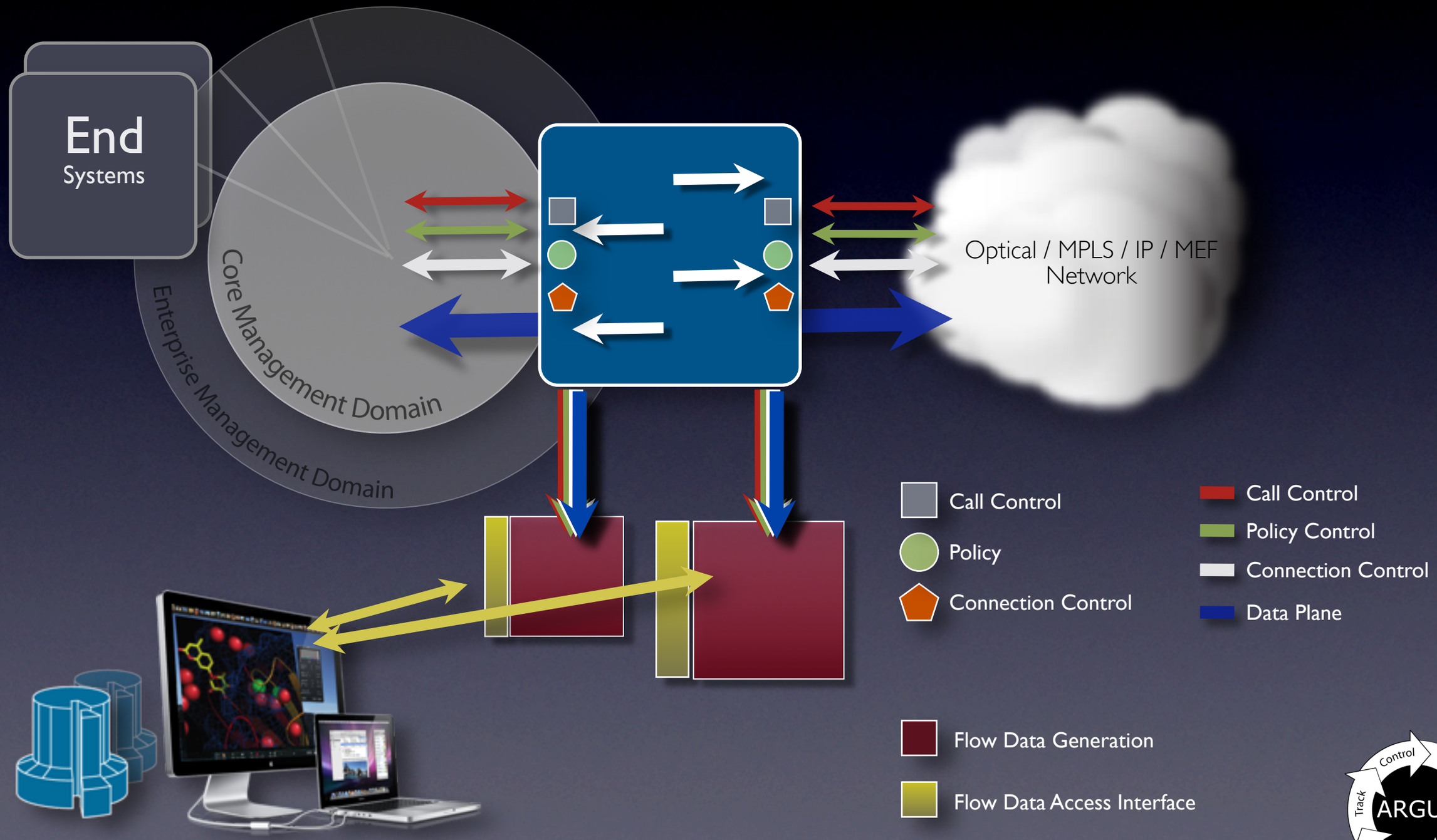
Configuration (cont)

- Argus Data Transport Strategies for Real-Time Operation
  - Reliable Pull Transport
    - Pull strategy, where the collector initiates the transport service, provides the best down stream awareness and control of sensor data availability
  - Connectionless Push Transport
    - Realtime operation demands high transport performance with minimal to no transport establishment and recovery time
      - ARGUS_OUTPUT_STREAM
        - Specifies push transport strategies for data output
        - UDP based push data transport: argus-udp://host:port
        - Multicast push transport: argus-udp://multicastGroupAddr:port
          - Multicast can enable huge parallelism in flow data processing
      - ARGUS_MAR_STATUS_INTERVAL
        - Needs to be adjusted to around 1-5 seconds.

Deployment

- Monitoring Strategies
  - Enterprise Border Monitoring
  - Subnet Monitoring
  - End System Monitoring
  - Complex/Comprehensive Monitoring

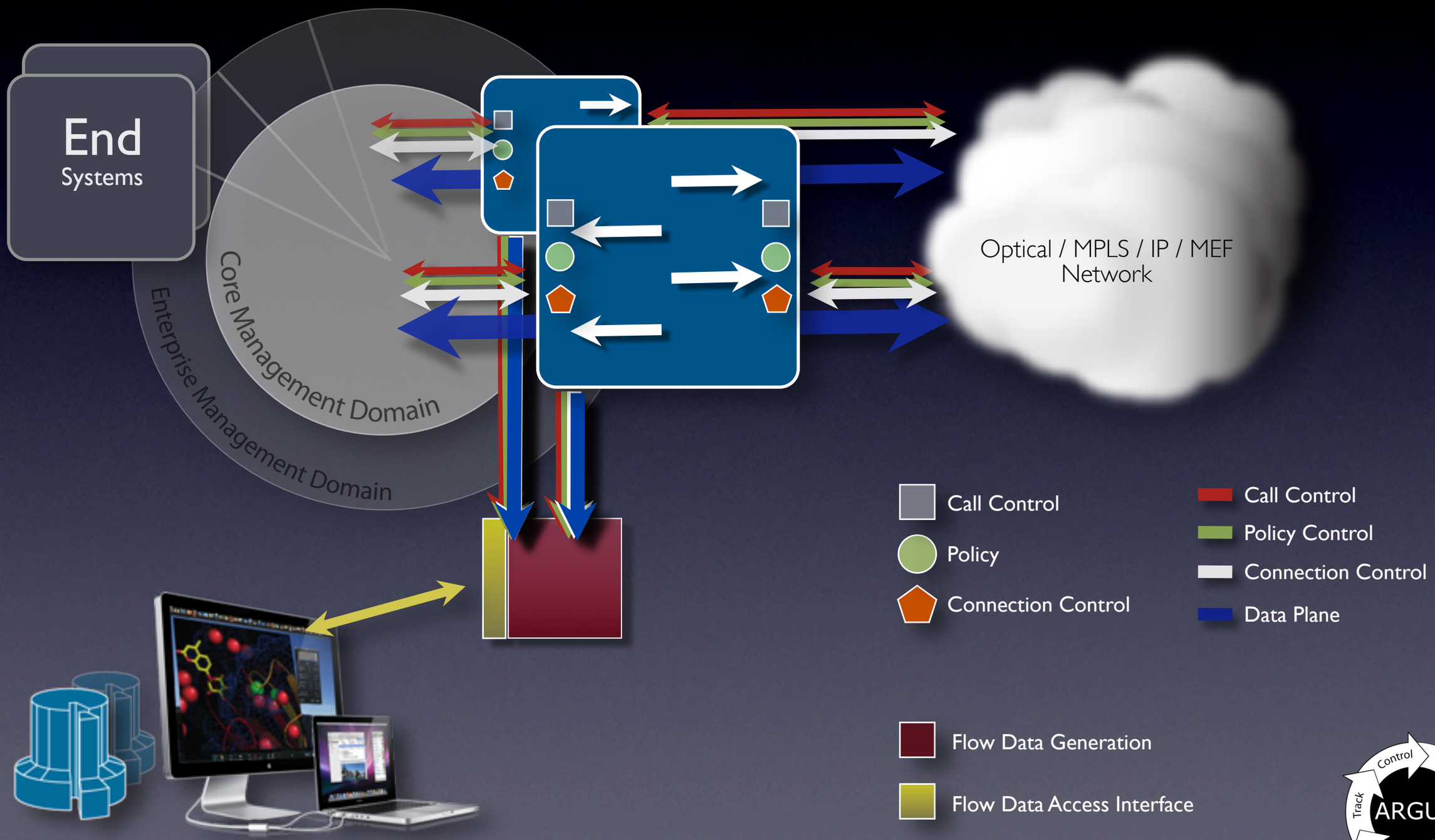
Enterprise Border Awareness
Internal/External Strategies



Enterprise Border Awareness

Asymmetric Routing Strategies

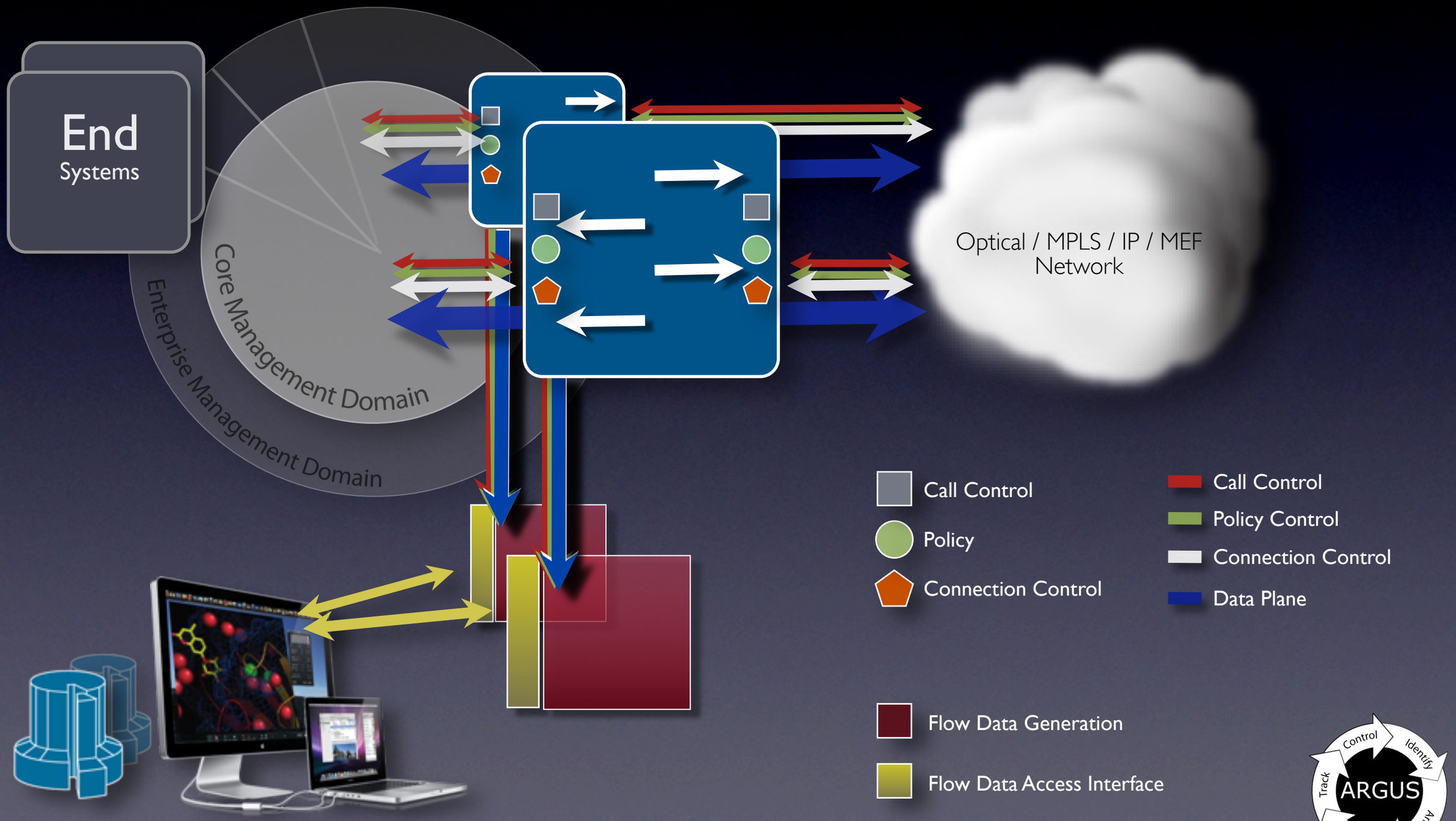
Single Probe



Enterprise Border Awareness

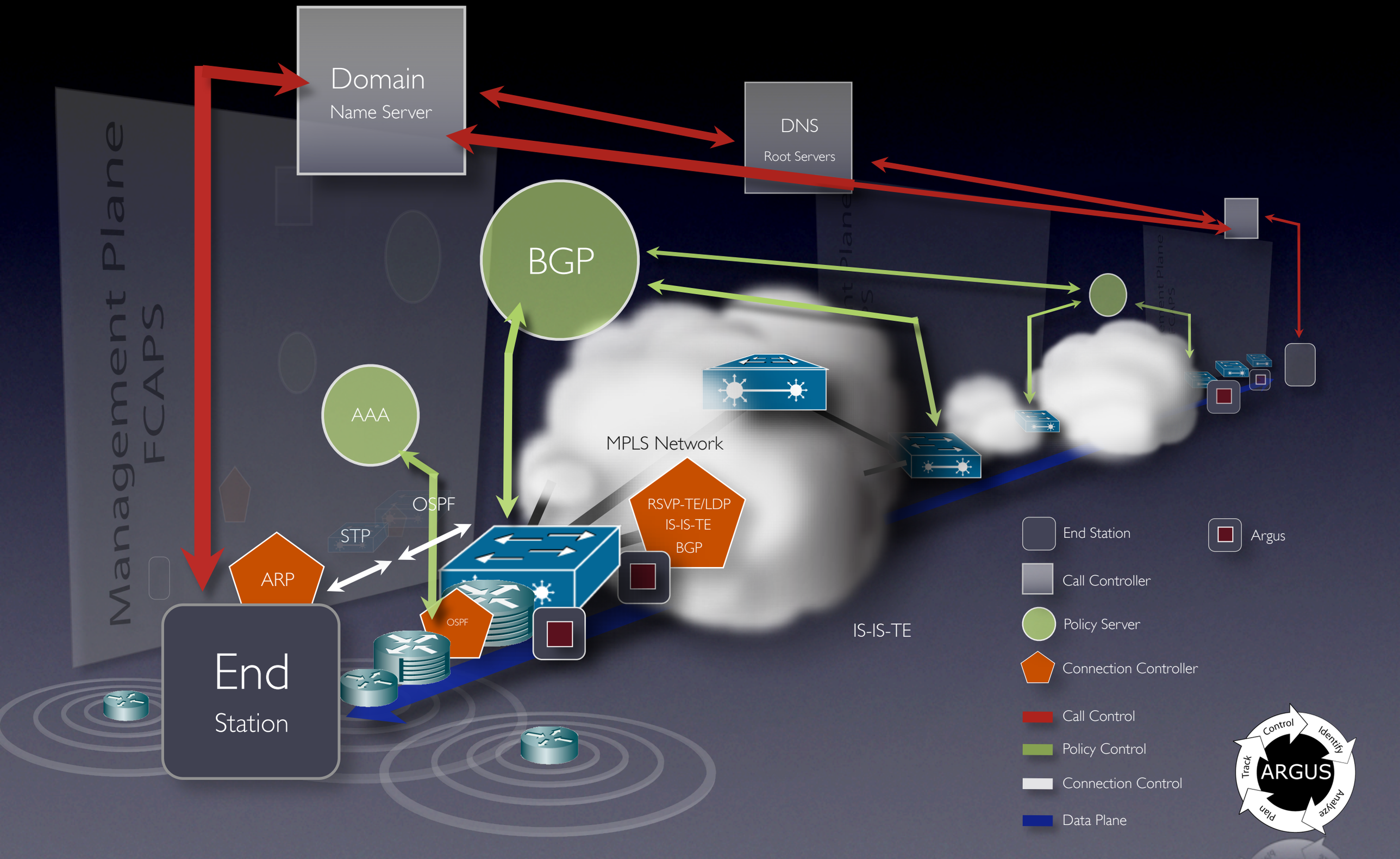
Asymmetric Routing Strategies

Multiple Probes



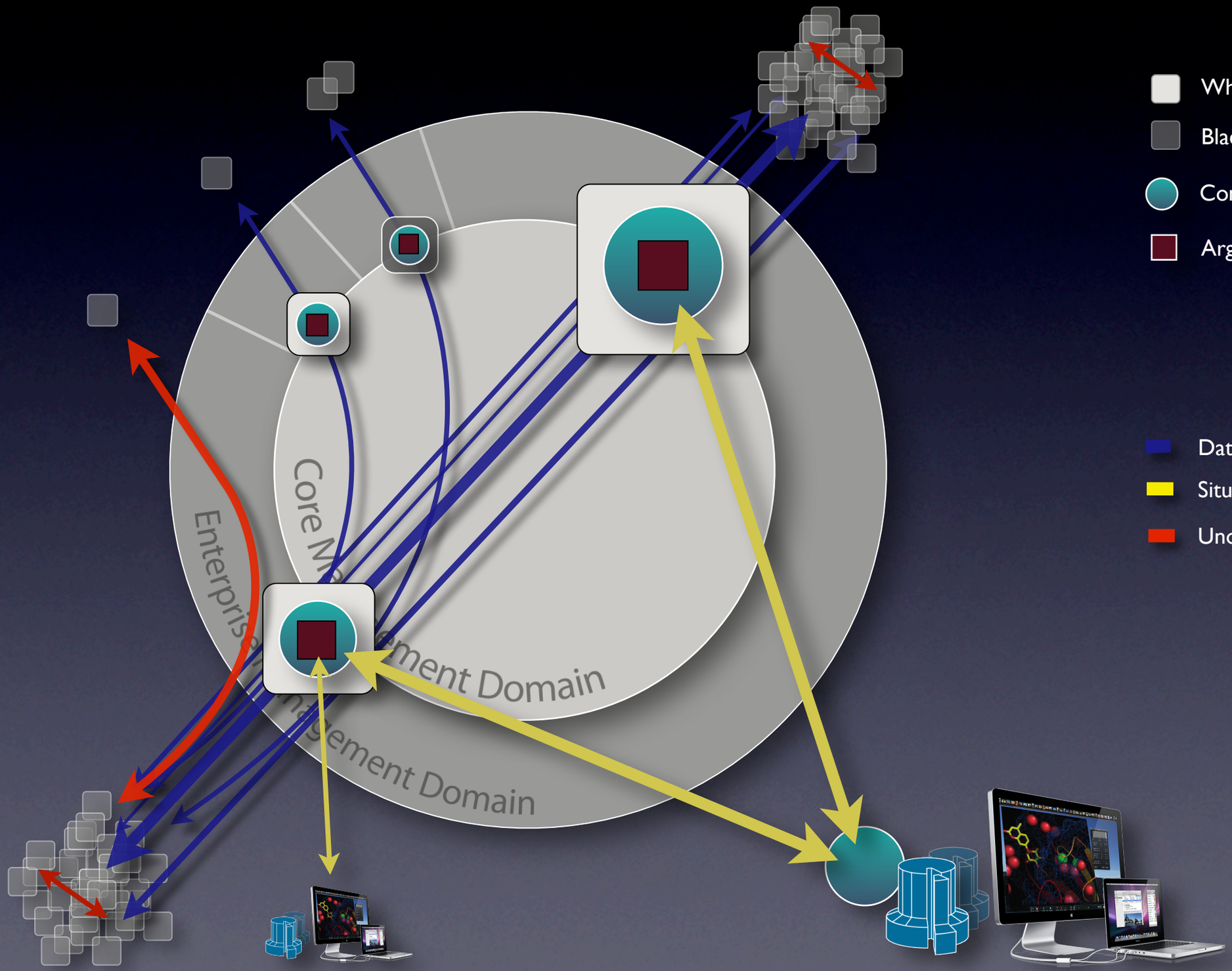
Enterprise Border Awareness

Outside Inside / Them vs Us



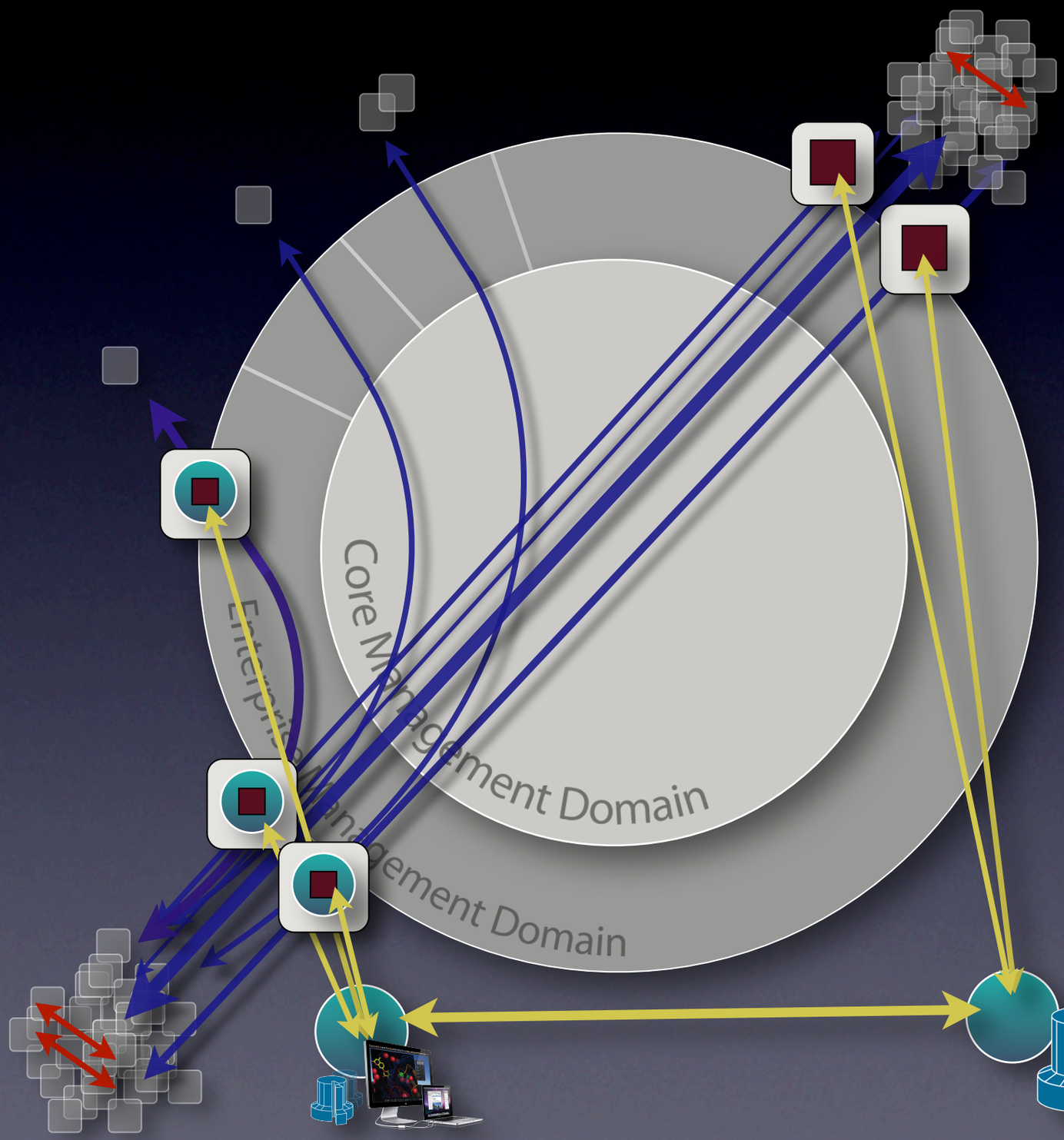
Enterprise Border Awareness

Outside Inside / Them vs Us



Subnet Border Awareness

Local and Remote Strategies



White/Visible Node

Black/Non-Visible Node

Comprehensive Flow IS

Argus Sensor

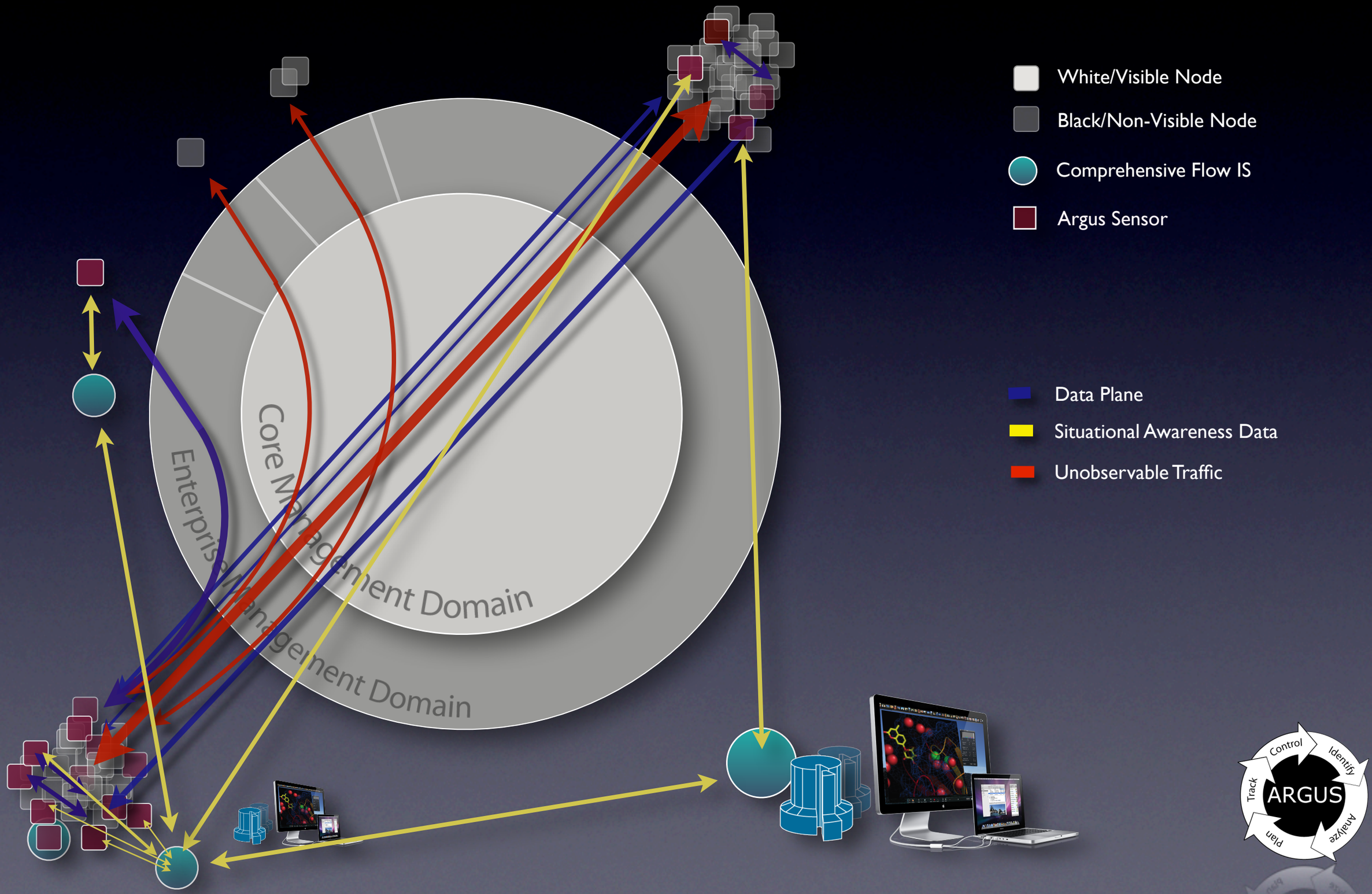
Data Plane

Situational Awareness Data

Unobservable Traffic

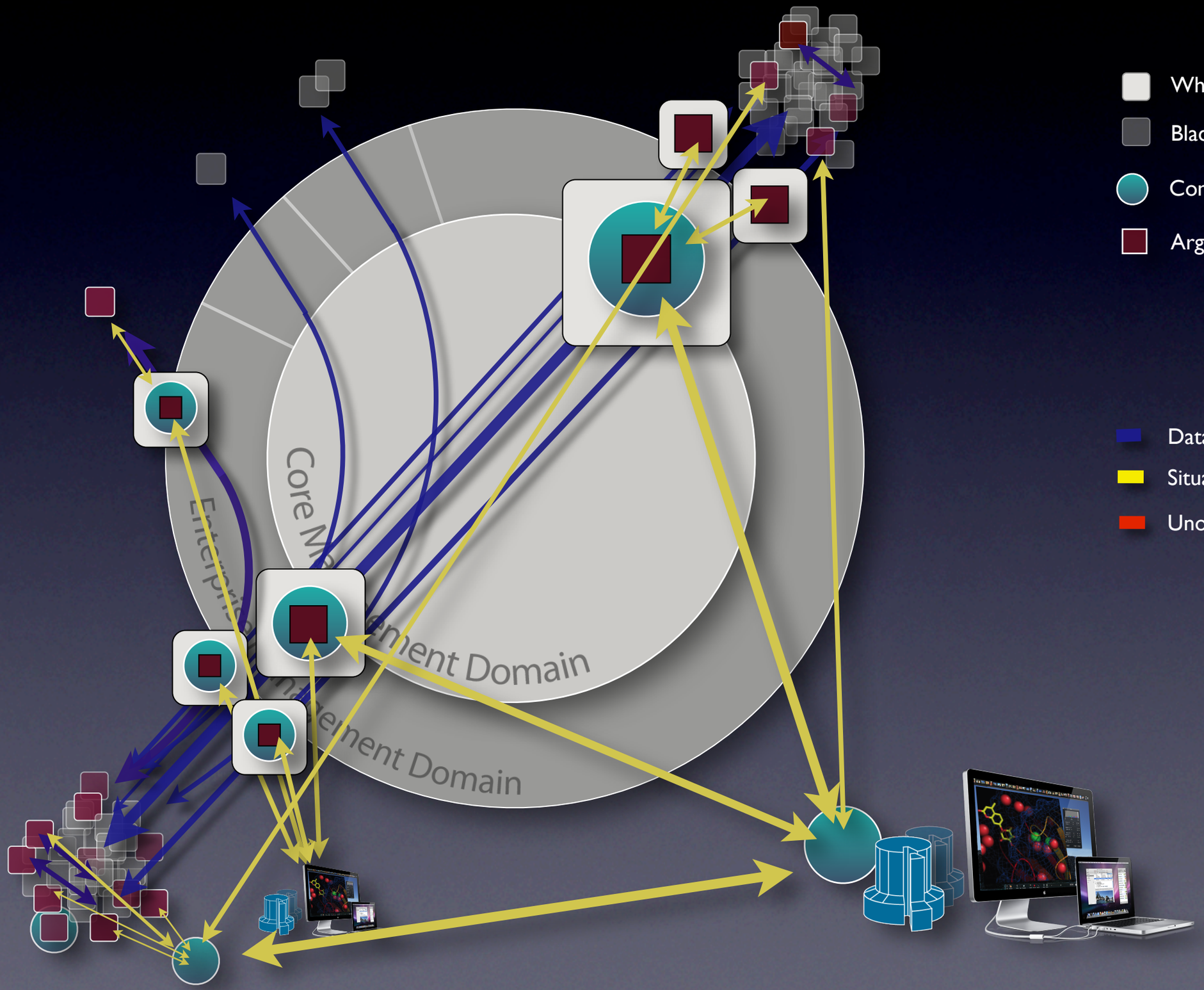
End System Awareness

Local and Remote Strategies



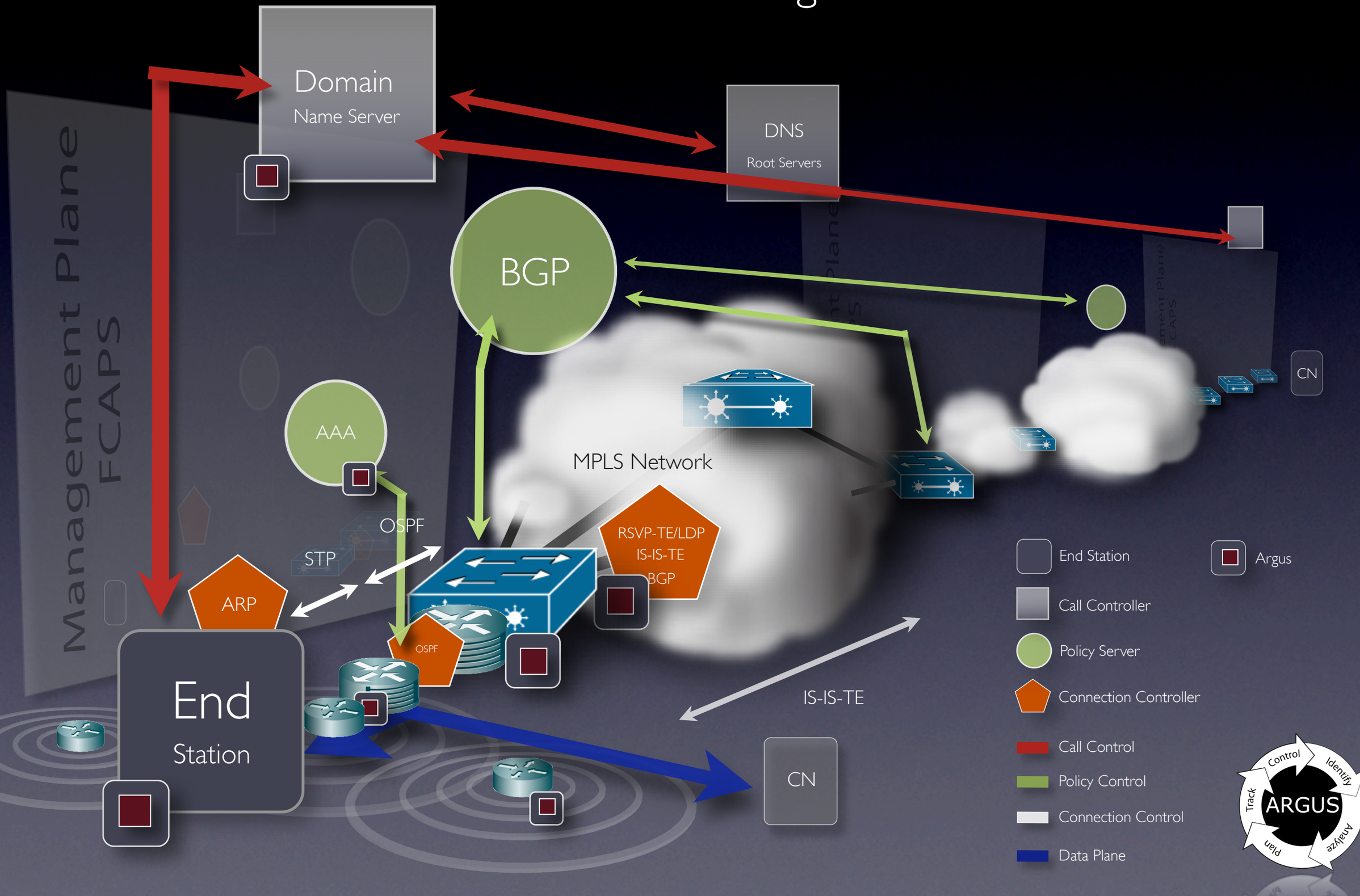
Complex Comprehensive Awareness

Local and Remote Strategies



Comprehensive Enterprise Awareness

Dealing with the Insider Threat

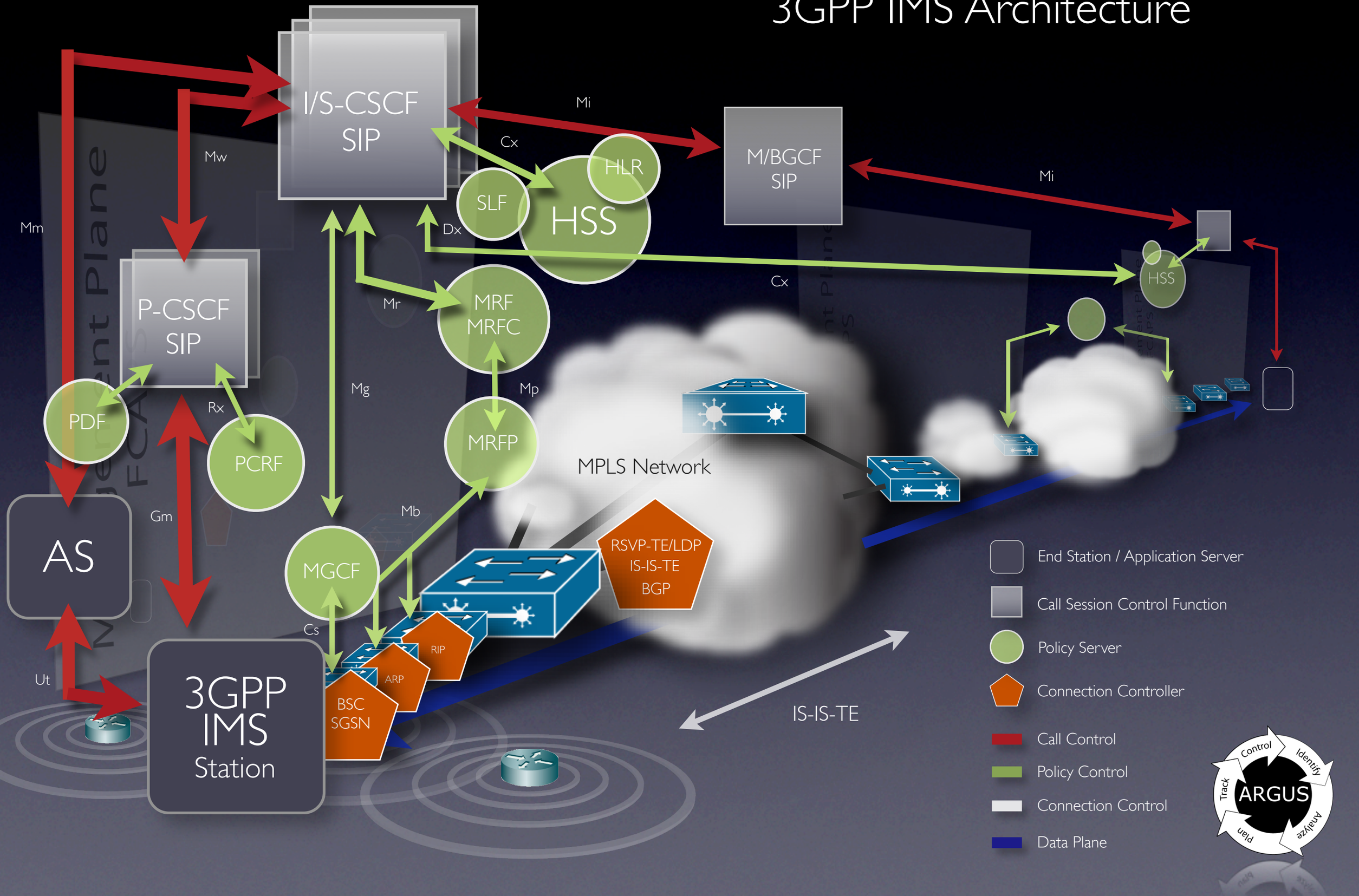


Complex Monitoring

- Critical elements
  - Time synchronization
  - Comparable flow key models
    - If collection system provides complex streaming analytics and aggregation
  - Observation Domain ID Allocations
    - Unique identifiers throughout the complete system
- Real-Time Operation
  - All sensors use same ARGUS_FLOW_STATUS_INTERVAL
  - All intermediate processing operates in the same time domain

Mobile User / Data Networks

3GPP IMS Architecture



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

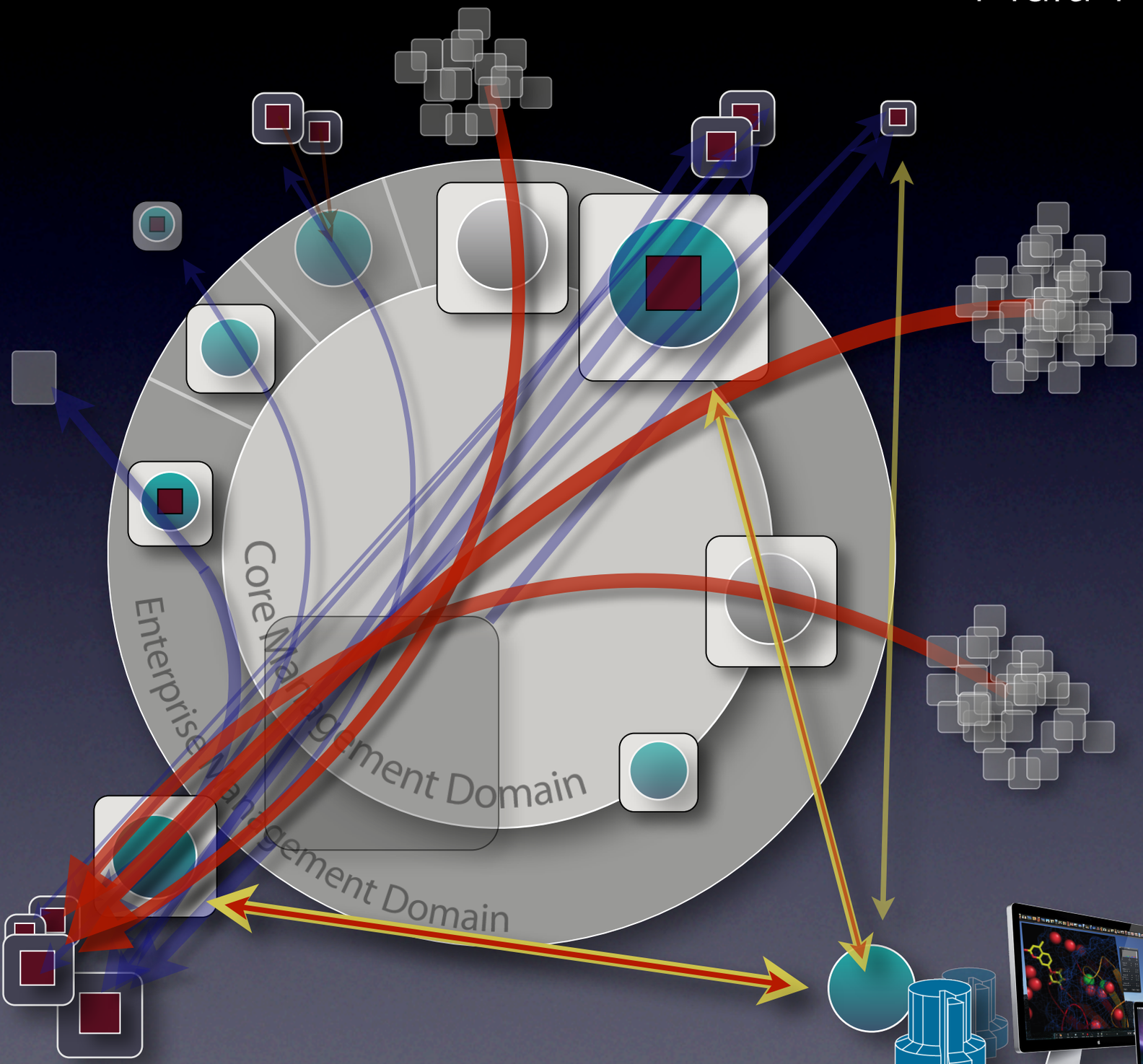
SNMP RMOM Style Monitor

Information System Repository

ARGUS
Control
Identify
Analyze
Plan
Track

Distributed Situational Awareness

Multi-Probe Multi-Site



White/Visible Node

Black/Non-Visible Node

Comprehensive Flow IS

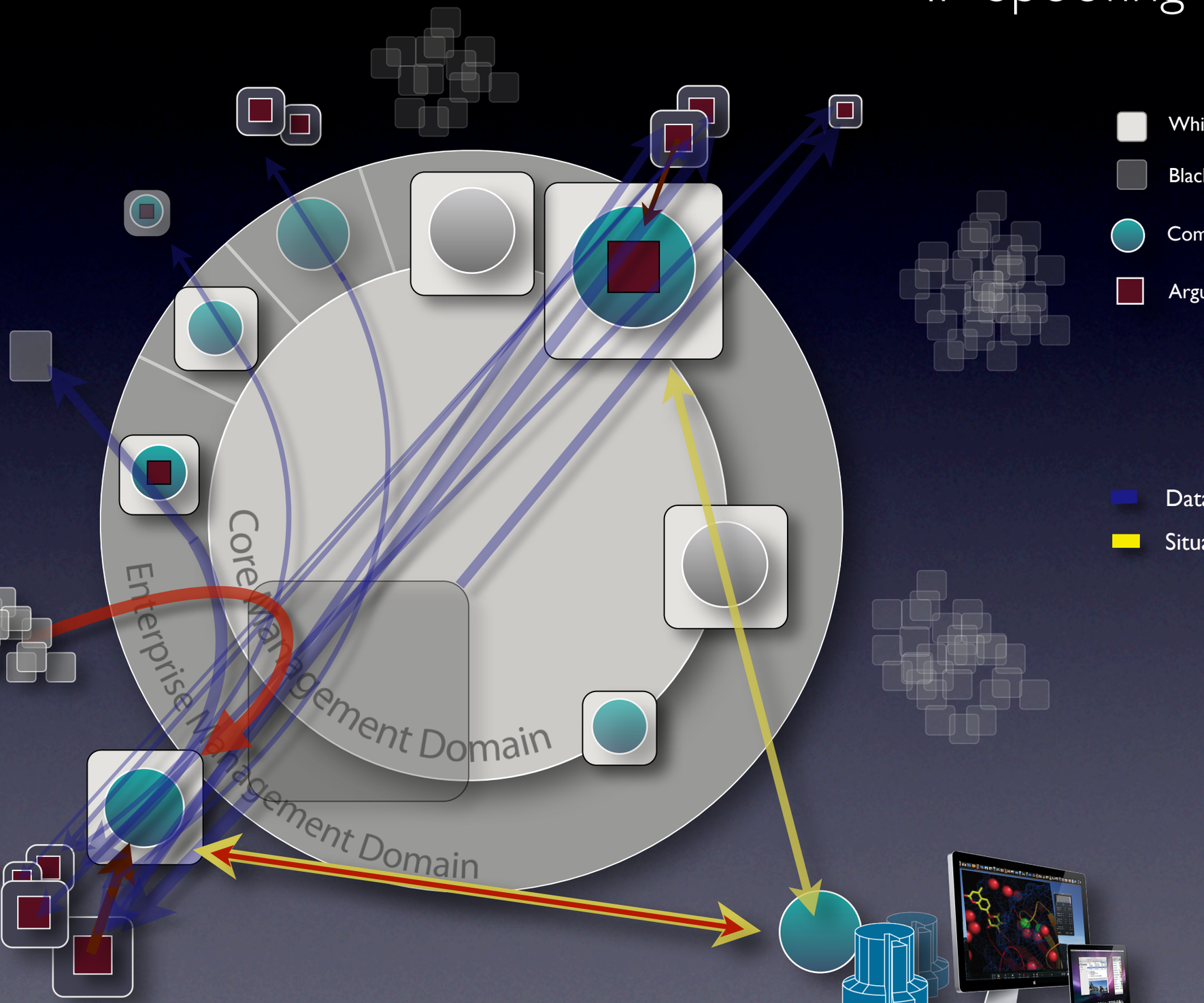
Argus Sensor

Data Plane

Situational Awareness Data

Distributed Situational Awareness

IP Spoofing Scenarios



White/Visible Node

Black/Non-Visible Node

Comprehensive Flow IS

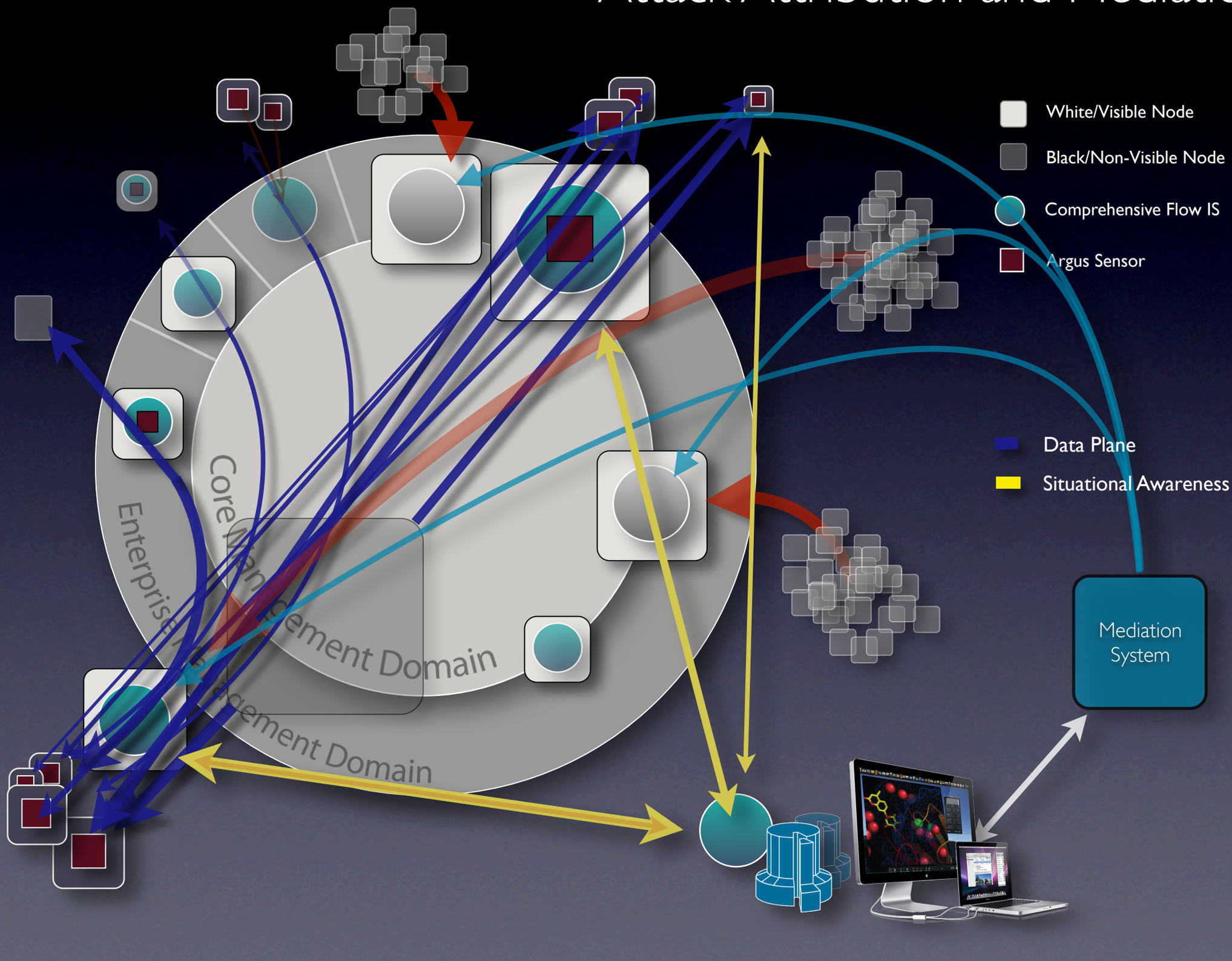
Argus Sensor

Data Plane

Situational Awareness Data

Distributed Situational Awareness

Attack Attribution and Mediation



Data Collection

Data Collection

All ra* programs can read data from any Argus data source, files, stream, encrypted, and/or compressed, and can write current file structure.

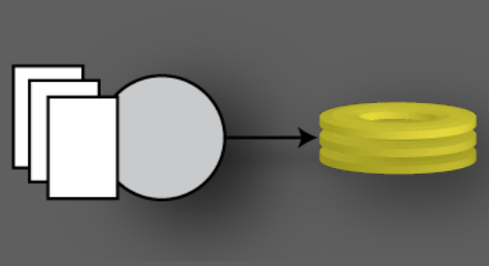
Making a real-time argus based system needs just a little bit more.

- File Distribution
- Radium Distribution
- Argus Repository Establishment
  - cron
  - rasplit/rastream
  - rasqlinsert/rasql

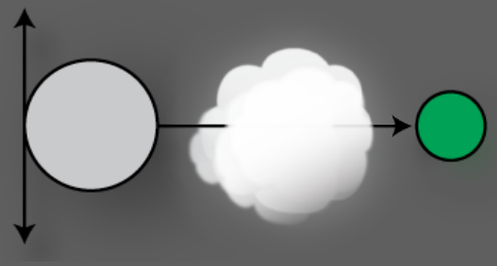
Data Collection

- Argus Data Distribution
  - Real Time Streaming Distribution
    - Data Flow Machine Architecture
      - Stream Processing Pipelines
    - Transport Protocols
      - Push and Pull Reliable and Unreliable Unicast
      - Push Multicast
  - File Based
- Argus Data Collection
  - Simple Collection Strategies
  - Complex Hierarchical Collection and Distribution

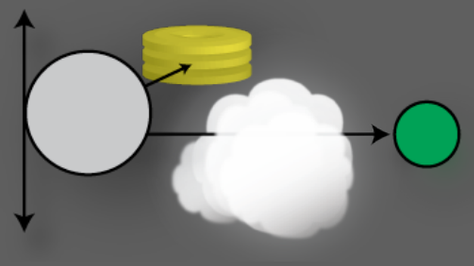
Data Collection



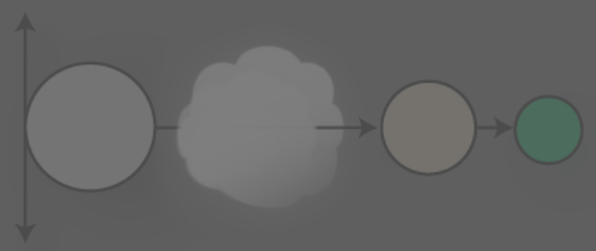
Argus reading from packet files or network and writing directly to disk



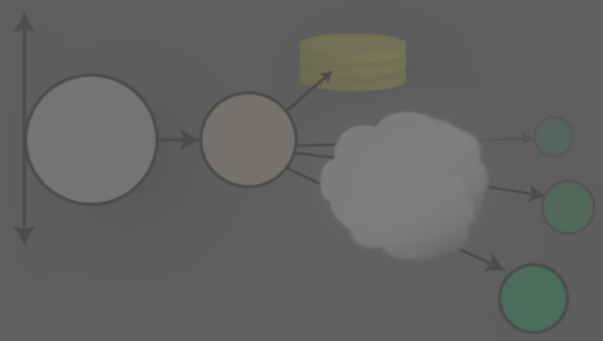
Argus reading from the network and writing directly to network based client



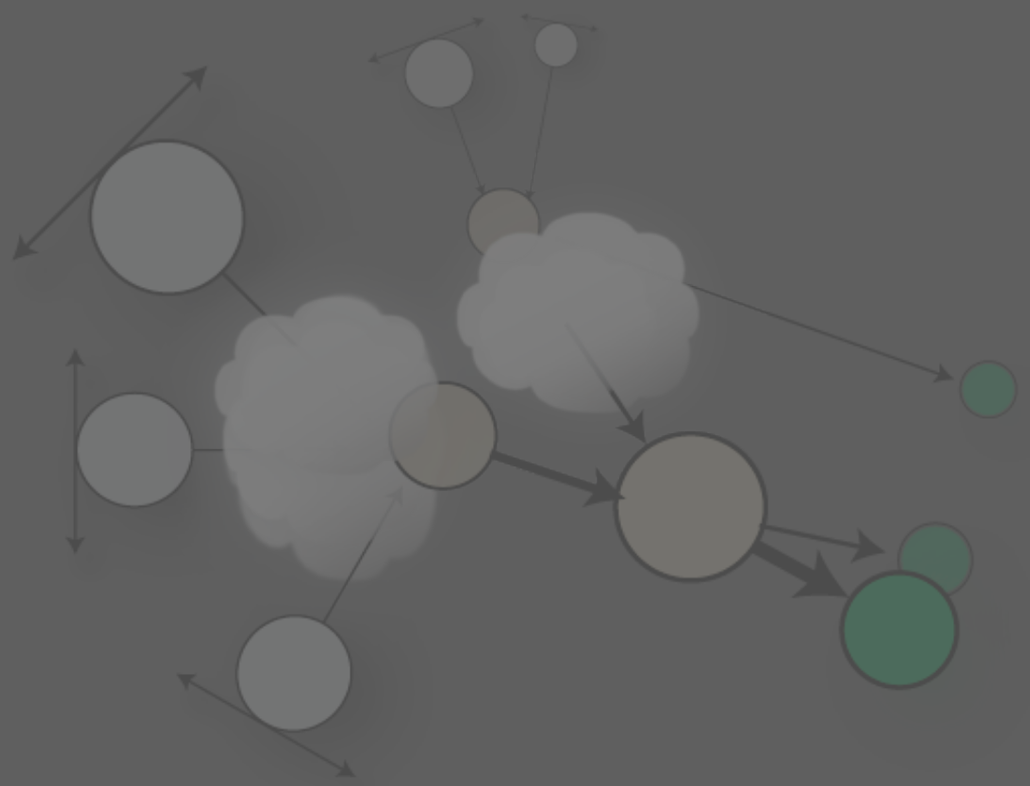
Argus reading from the network and writing directly to disk and network based client



Argus reading from the network and writing directly to a network Radium, writing to a client



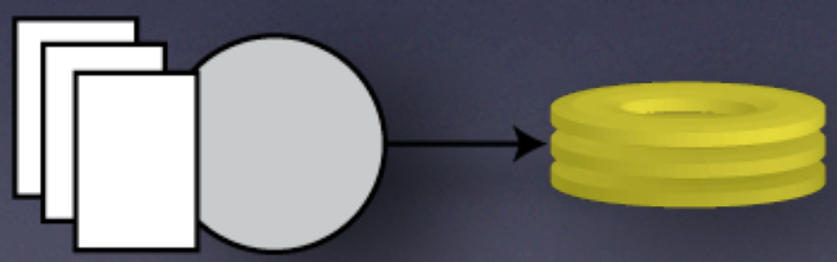
Argus writing to local Radium which is writing directly to disk and to network based clients



Many Argi writing directly to a Radium based distribution network, which is providing data to a set of clients.

Data Collection

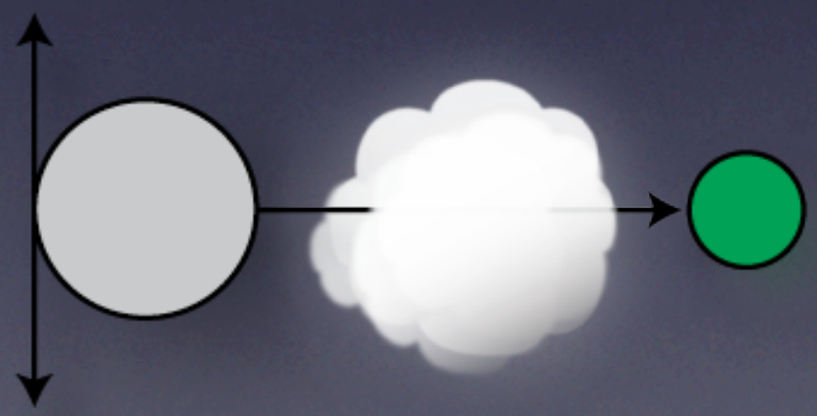
- Local Generation and Storage
  - Basis for argus-2.0 argusarchive.sh
  - Direct argus support for renaming files
  - Normally cron mediated
  - Issues with time and record spans
  - System designer has most control !!!



Argus reading from packet files or network and writing directly to disk

Data Collection

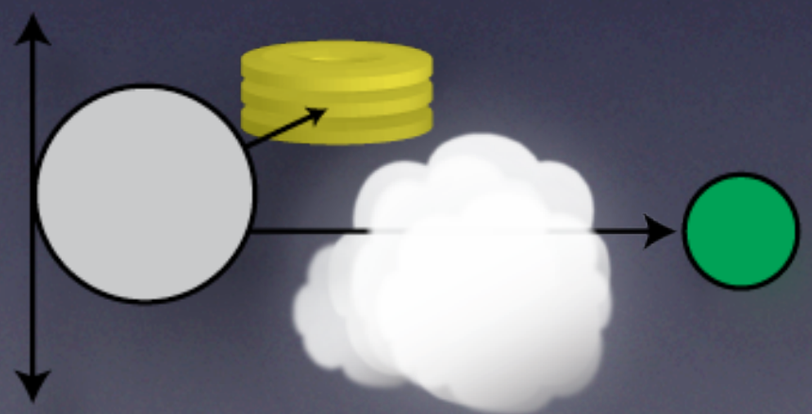
- Local Generation Remote Collection
  - Most high performance systems use this strategy
    - Provides explicit scalability and performance capabilities
    - Relieves argus from physical device blocking
    - Network interfaces generally faster than local storage devices
  - Introduces network transport issues
    - Reliability, connection vs. connection-less, unicast vs multicast, congestion avoidance, access control and confidentiality



Argus reading from the network and writing directly to network based client

Data Collection

- Local Storage and Remote Collection
  - Used when data reliability is most critical
    - Local storage provides explicit data recovery
    - File collection provides additional distribution flexibility
    - Scheduled transport
- Reduces ultimate sensor performance
    - Argus itself is doing a lot of work
    - Packet processing is really the ultimate limit



Argus reading from the network and writing directly to disk and network based client

Data Collection

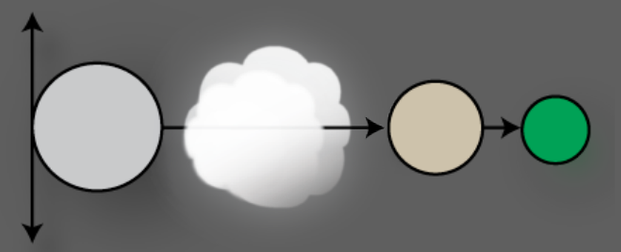
Complex Collection Hierarchies

Data Collection

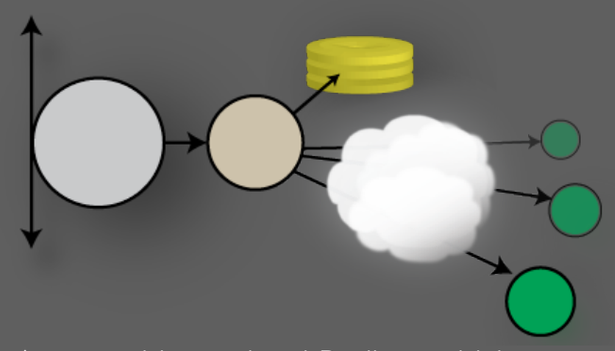
Argus reading from packet files or network and writing directly to disk

Argus reading from the network and writing directly to network based client

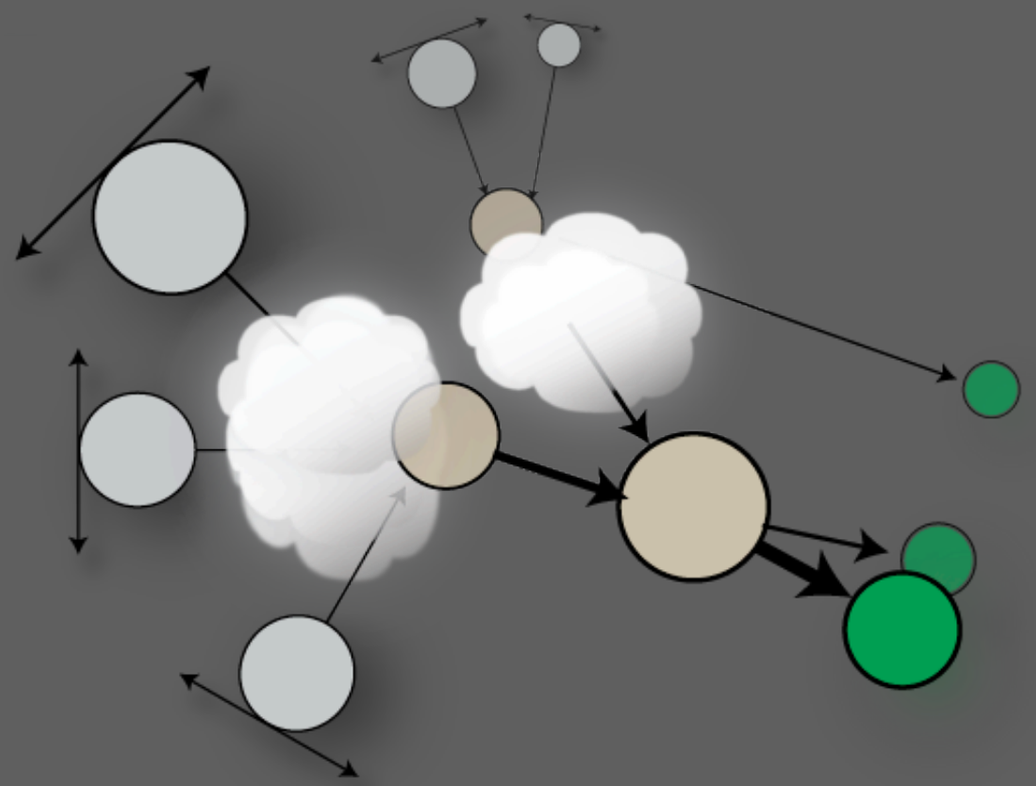
Argus reading from the network and writing directly to disk and network based client



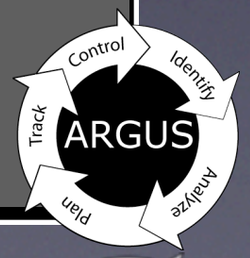
Argus reading from the network and writing directly to a network Radium, writing to a client



Argus writing to local Radium which is writing directly to disk and to network based clients

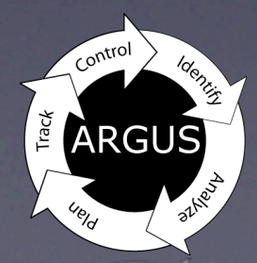


Many Argi writing directly to a Radium based distribution network, which is providing data to a set of clients.



Data Collection

- Radium
  - Primary argus data distribution technology
  - Radium is a ra* program with an argus output processor.
    - Read from many sources
    - Write to many clients
    - Serve up argus data files
    - Process/transform data
    - Configuration is combo of argus() and ra()
- Supports very complex data flow machine architectures.

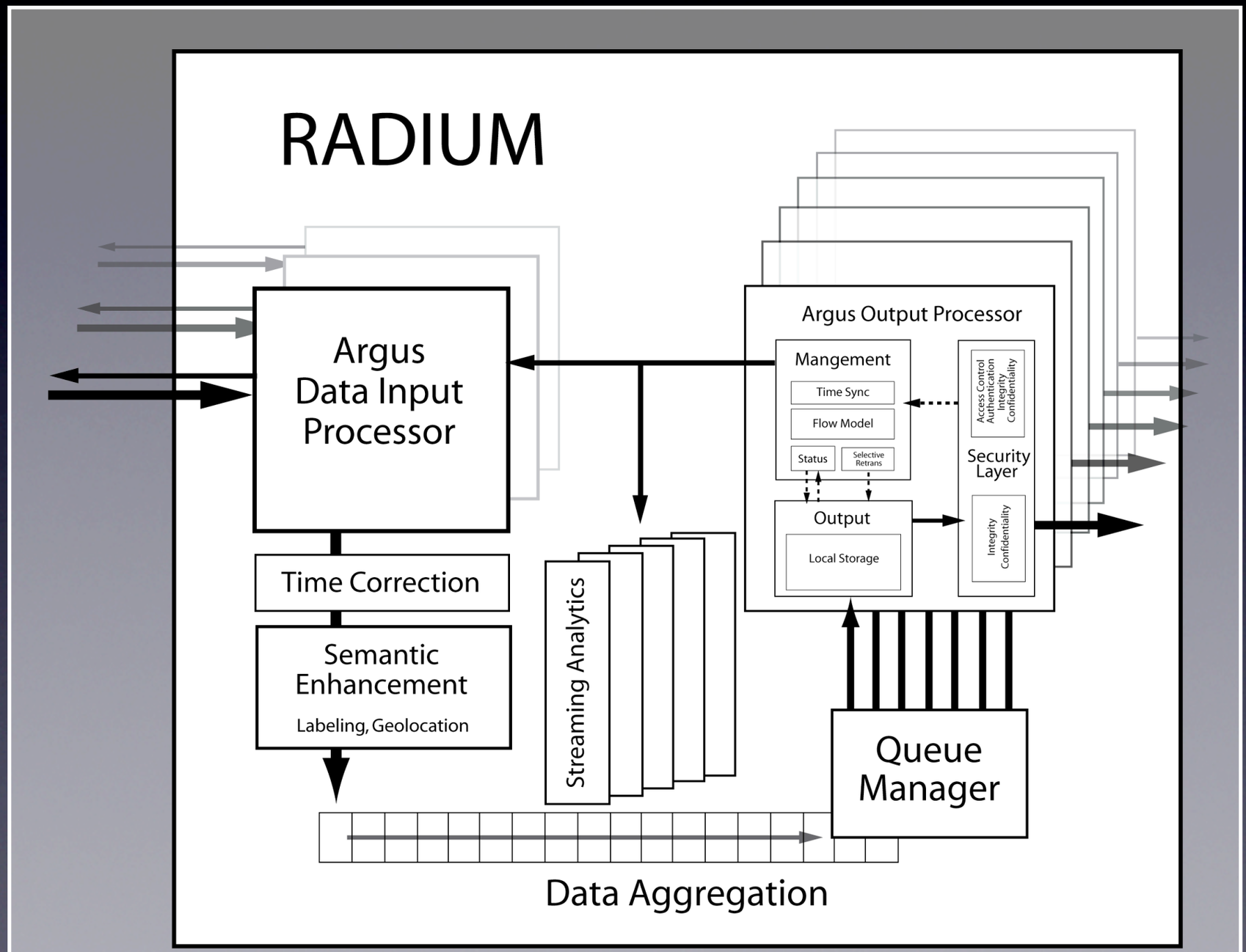


Radium

- Hybrid Argus and Argus client
  - Argus Client
    - Read argus data from all supported files and streams
    - Can read Netflow, Sflow, Jflow and FlowTools data
    - Reads up to 256 argus data sources, generates 1 output
    - Supports most ra* functions by design:
      - Filtering
      - Labeling - full ralabel.1 functionality
      - Flow Correction - time sync correction, direction
      - Aggregation - rabins() behavior
      - Stream Analytics - future work
- Argus
  - Supports 256 argus data output processors
    - One radium, one output stream x256
    - Independent processors, independent outputs
      - Different transports, filters, sockets, files, etc.....

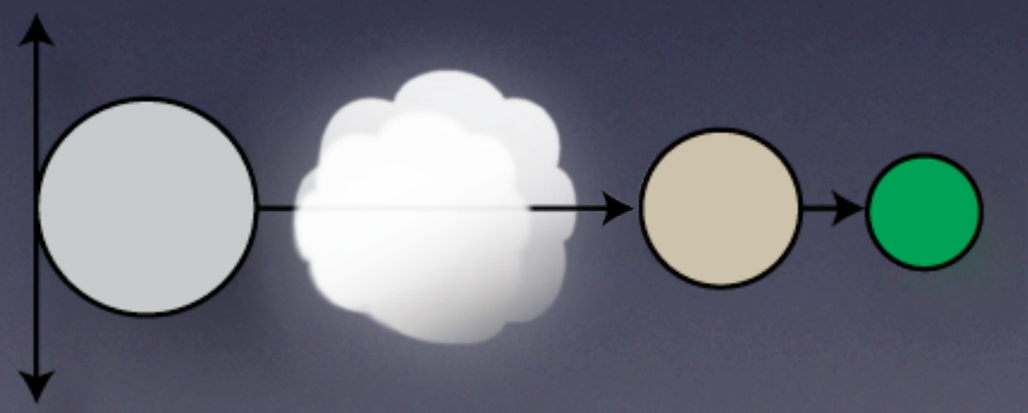
Argus Collection Design

Radium Process



Data Collection

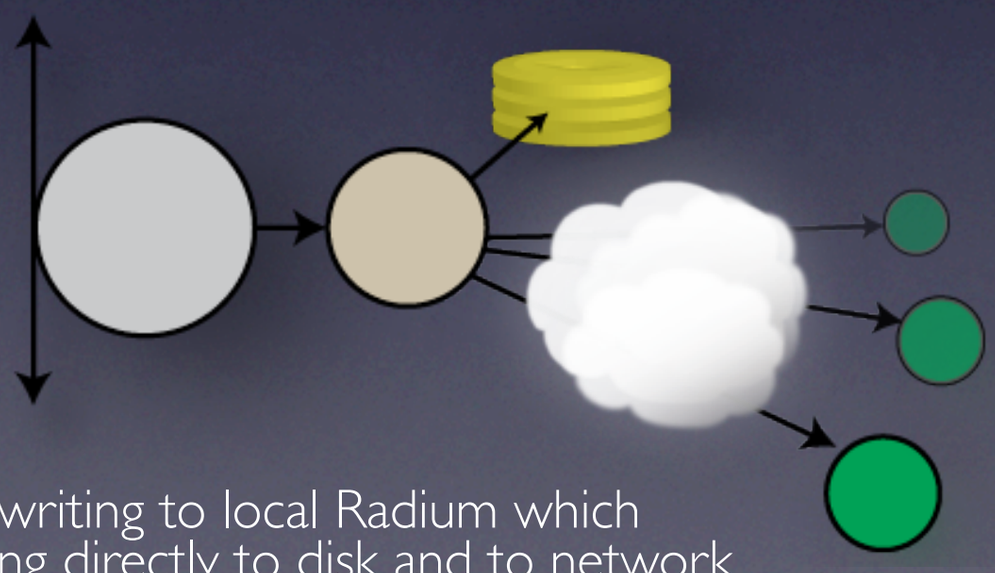
- Local Generation Remote Distribution
  - Most prevalent strategy used in argus-3.0
    - Provides explicit scalability and performance capabilities
    - Provides most stable collection architecture from client perspective
    - Single point of attachment for complete enterprise
  - Least reliable of ‘advanced’ strategies
    - Radium failure interrupts continuous stream collection, with no opportunity for recovery



Argus reading from the network and writing directly to a network Radium, writing to a client

Data Collection

- Local Distribution and Storage
  - Best methodology
    - Provides explicit scalability and performance capabilities
    - Provides most reliable collection architecture
    - Multiple points of attachment, multiple points of control
  - Most expensive strategy at data generation
    - Radium deals with device and remote client requests for data which does come with a processor and memory cost



Argus writing to local Radium which is writing directly to disk and to network based clients

Data Collection

- Complex data flow machine architectures
  - Architecture of choice for scalability
    - Provides explicit scalability and performance capabilities
    - Provides most parallelism
    - Multiple points of attachment, multiple points of control
  - Can get a little complex
    - Merging of multiple flows, multiple times, introduces complex data duplication issues, and allows for complex, incompatible data schemas to co-exist

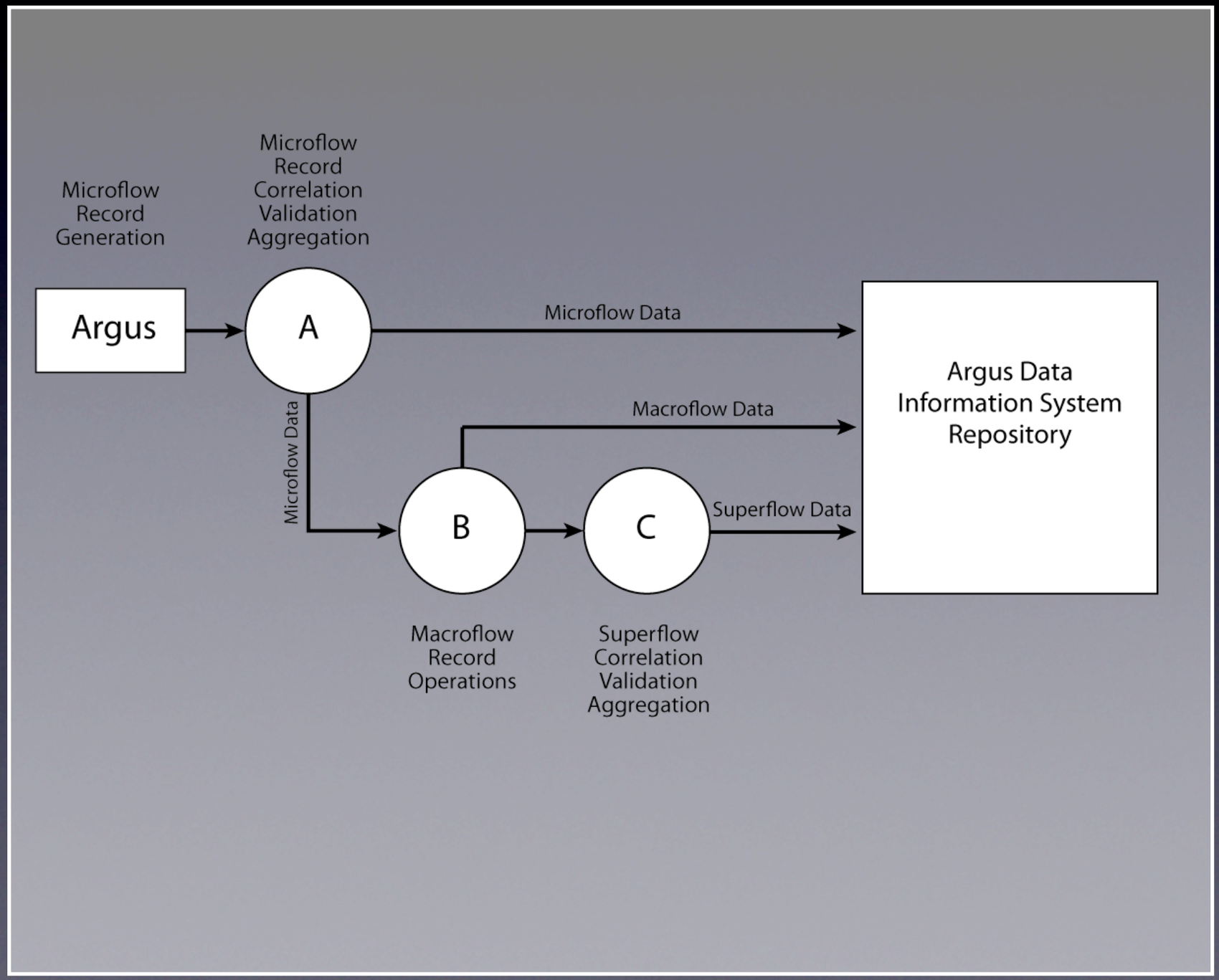
Many Argi writing directly to a Radium based distribution network, which is providing data to a set of clients.

Radium

- Real-time operation
  - Radium, is designed as a non-blocking data distribution node
  - Implemented as multi-threaded input and output processor(s)
    - Input processed and placed in single process queue
      - Read up to 256 argus data sources
      - Generates 1 output data stream
      - Queue manager continuously distributes records to the collection of output processors
      - The more cores, the less queuing, locking and scheduling
    - Aggregation and analytics introduce delay
      - rabins() function demands buffer holding times
        - Aggregation over a fixed period of time.
      - Stream Analytics - process within locked time "bin"
      - Queue manager must wait for analytics to complete

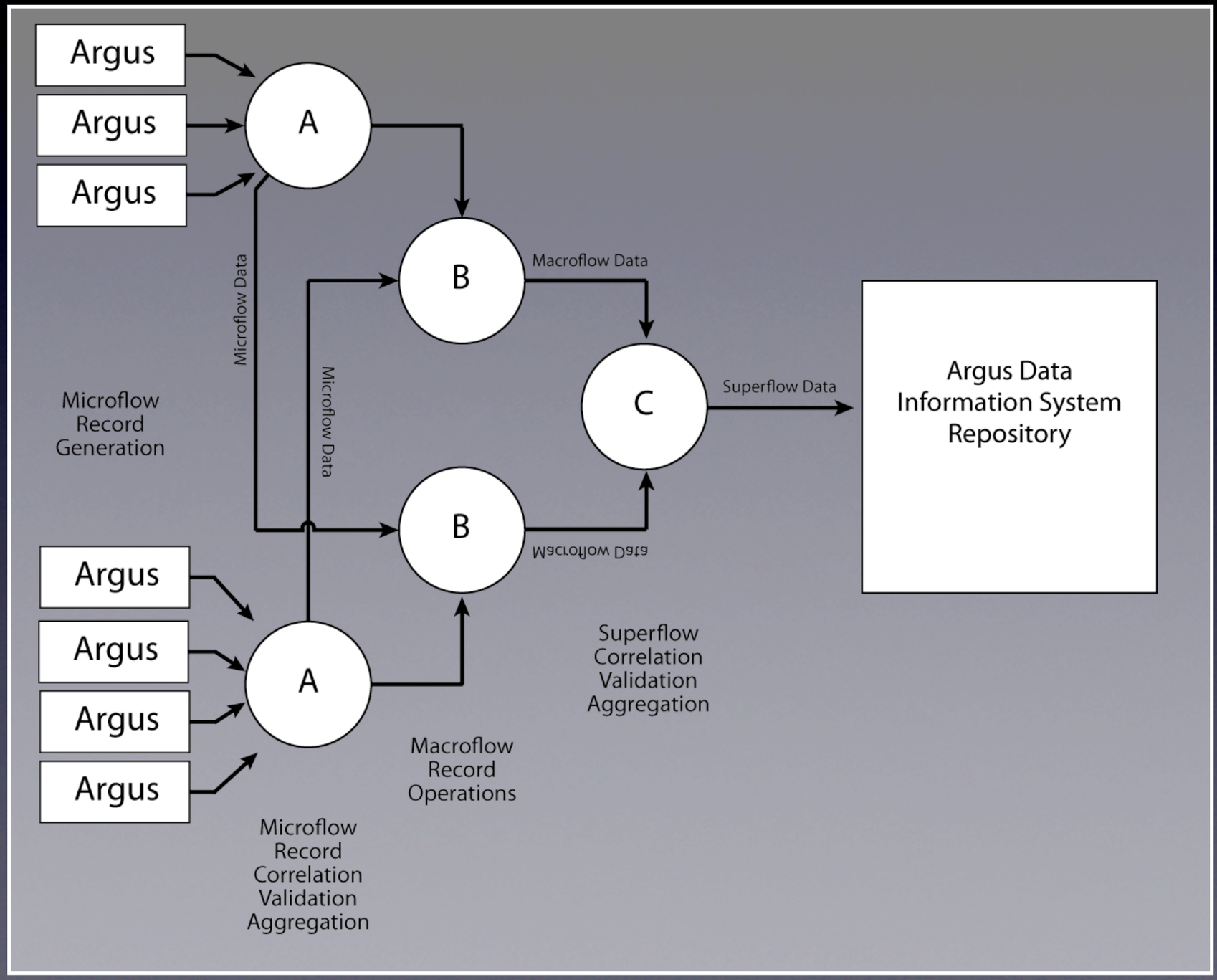
Radium

Data Flow Machine Architectures



Radium

Data Flow Machine Architectures



Argus Repositories

Argus Repositories

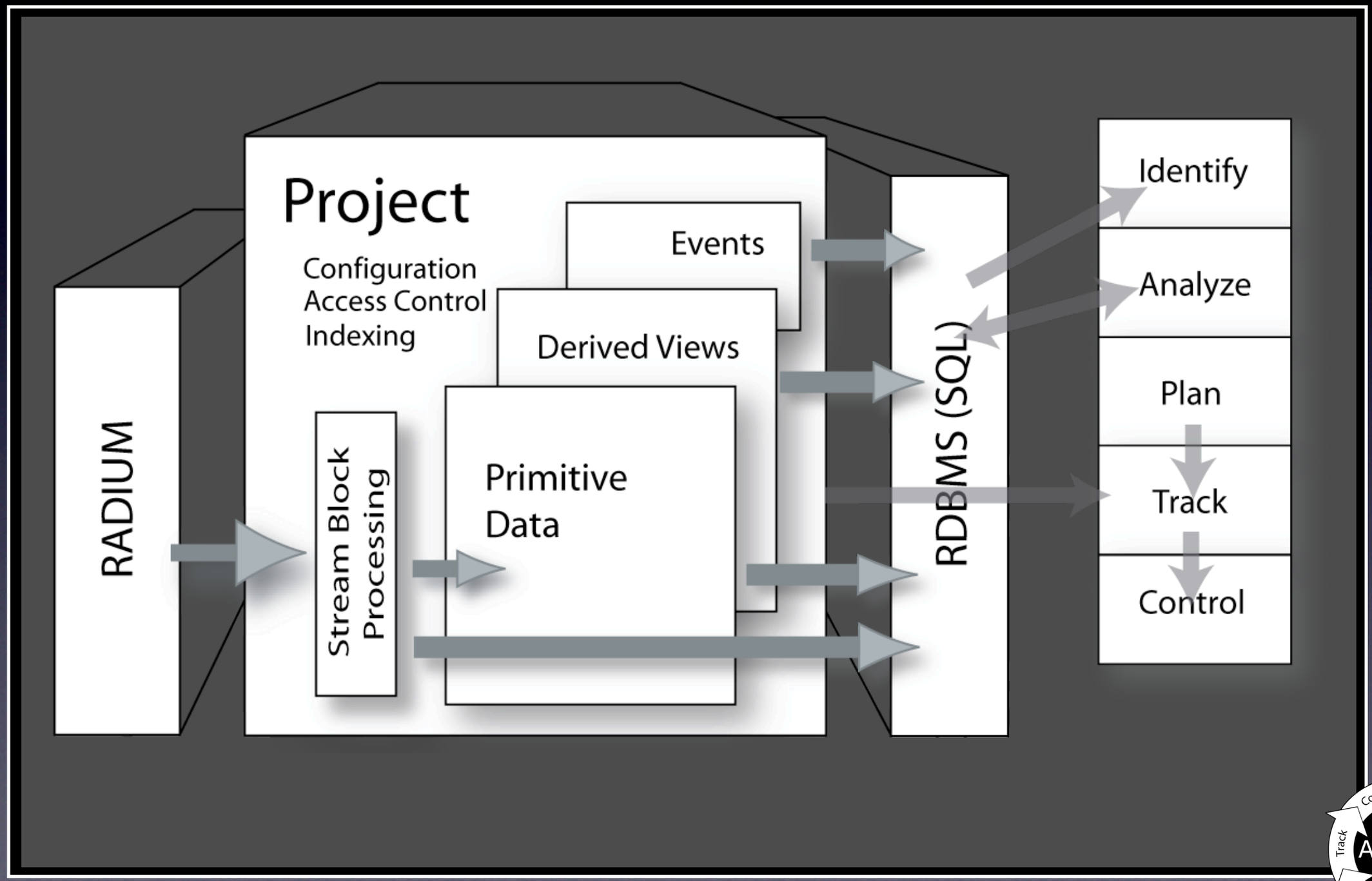
- Argus Repository Establishment
  - Formal Ingest/Disposition
- Repository Function
  - Primitive Data Repository
    - General Archive
    - Access Control
    - Retention Policies
    - Modification Policy (Compression)
  - Derived Data Repositories

Argus Repositories

- Native File System
  - Simplicity
  - Performance
  - Compatibility
- Relational Database System (RDBMS)
  - Extensive Data Handling Capabilities
  - Complex Management Strategies
  - Performance Issues

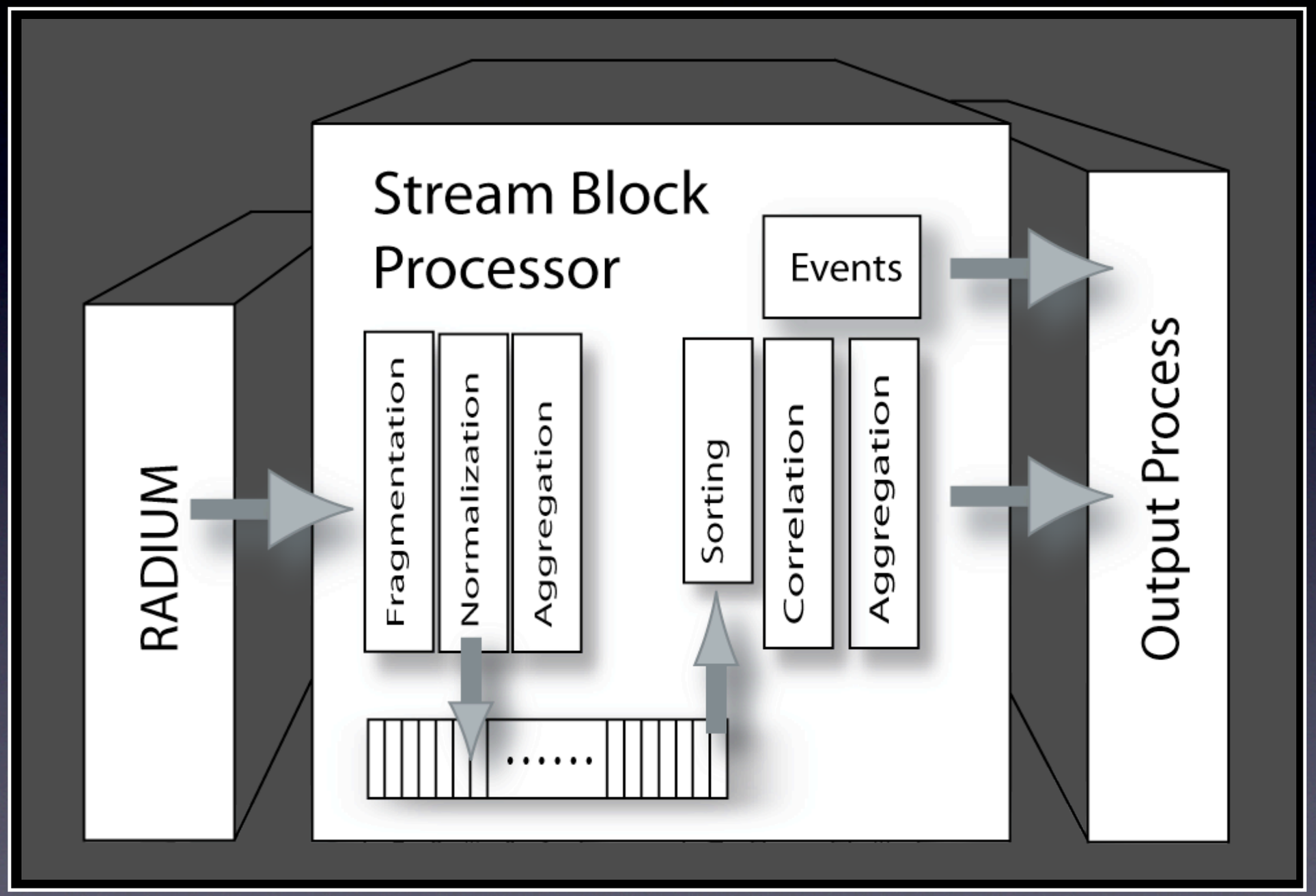
Argus Processing Design

Radium Stream Block Processor



Argus Processing Design

Stream Block Processor



Argus Repositories
Data Ingest Support

- Stream Block Processing
  - rasplit
  - rastream
  - rabins
  - rasqlinsert

Argus Repositories
Best Common Practices

- File system archives
  - Primitive and derived data file systems
  - RDBMS managed complex indexing
  - rastream
    - /sourceId/year/month/day file structure
    - 5 minute files
    - 288 entries per day
    - Matches native file system performance for searching
    - Analogous to Google's Big Table filesystem
- RDBMS based archives
  - Short term data held in RDBMS
  - Rolled into file based system after N days.
  - Binary data inserted into database
  - Primitive data schema includes 'autoid'
  - Table names provide explicit partitioning

Argus Repositories
Real Time Processing Strategies

- rasqlinsert based data insertion / management
  - Complete argus data analytic engine
    - Complex aggregation support
    - Semantic enhancement
    - Time and data correction

- Continuous flow status maintained in table
  - Configurable update refresh intervals
  - Idle timeout options provides windowed SA

- RDBMS handles concurrency: updates and access
- RDBMS enabled trigger support

Argus Repositories
Real Time Processing Strategies

- rasql based client data access
  - RDBMS handles multiple access
  - Maintains cache management and concurrency
  - Local and remote access through federation

Argus Client Programs

Argus Client Programs

- Basic Operations
  - Printing, Filtering, Sorting, Splitting, Aggregation
  - Collection, Archiving, Anonymizaiton
- Graphing/Visualization/GUI
- Data Enhancement
  - Labeling
  - Geolocation
- Database Support
- User Data Processing
- Analytics

Basic Operations

- ra - provides data file creation, printing, filtering, content selection and data stripping
- rasort
- racluster - provides flexible data aggregation
- rabins -
- racount - simple record counting
- radium -
- rasplit -
- ranonymize

Argus Client Program Configuration

- Extensive command line options
- .rarc configuration
- Runtime Environment
- Data Access
- Printing Support
- Security

Argus Client Program Configuration

- Runtime Environment
  - RA_SET_PID
  - RA_PID_PATH
  - RA_OUTPUT_FILE
  - RA_TIME_RANGE
  - RA_TZ
  - RA_DEBUG_LEVEL
  - RA_TIMEOUT_INTERVAL
  - RA_UPDATE_INTERVAL
  - RA_DELEGATED_IP
  - RA_RELIABLE_CONNECT

Argus Client Program Configuration

- Data Access
  - RA_ARGUS_SERVER
  - RA_SOURCE_PORT
  - RA_CISCONETFLOW_PORT
  - RA_TIME_RANGE
  - RA_RUN_TIME
  - RA_FILTER

Argus Client Program Configuration

- Printing Support
  - RA_PRINT_MAN_RECORDS
  - RA_PRINT_LABELS
  - RA_FIELD_SPECIFIER
  - RA_FIELD_DELIMITER
  - RA_FIELD_WIDTH
  - RA_PRINT_NAMES
  - RA_PRINT_RESPONSE_DATA
  - RA_PRINT_UNIX_TIME
  - RA_TIME_FORMAT
  - RA_USEC_PRECISION
  - RA_USERDATA_ENCODE

Argus Client Program Configuration

- Security
  - RA_USER_AUTH
    - This is the user name and, depending on the MECH, a group name, for the SASL account to be used for authentication.
  - RA_AUTH_PASS
    - This is the password for the SASL account to be used for authentication. This plain-text entry does pose some issues, so be sure and protect your .rarc file if this method is used.
  - RA_MIN_SSF
    - This is the minimum security strength factor for the connection. An SSF of 0 allows for no protection. An SSF of 1 will supply integrity protection without privacy.
  - RA_MAX_SSF
    - The MAX_SSF is normally used to specify the strength of encryption. 56, as an example, specifies 56-bit DES. This value should not be less than the MIN_SSF.

ra()

ra is the basis of all ra* programs, in that it is the simplest of the client programs written against the client library. It is simply, "read the data source and print each record, one record at a time".

- All ra* programs do what ra() does
- Use ra to inspect individual records
- ra.1 is your primary documentation
- If you want to develop argus clients
  - ra.c should be your first example

ratop()

ratop is the top() equivalent for argus data. It is becoming the argus data editor as time goes on.

ratop is also an argus data aggregator.

- ratop is the 1° argus client
- it incorporates all basic functions into a single program.
- it is the best program example for near-realtime situational awareness

Aggregation

Network data aggregation is a MASSIVE topic. It drives most of the data analysis and report generation and is the heart of all the interesting programs.

- racluster, rabins, ratop
  - Each field has its own aggregation methods
  - Semantic preservation

Aggregation

- Configuration

```
RACLUSTER_MODEL_NAME=Test Configuration
RACLUSTER_PRESERVE_FIELDS=yes
RACLUSTER_REPORT_AGGREGATION=yes
RACLUSTER_AUTO_CORRECTION=yes

filter="tcp or udp"  model="saddr daddr proto dport"       status=120 idle=3600 cont
label ="Class-Video" model="srcid saddr daddr proto dport" status=5   idle=10
filter="tcp or udp"  model="saddr daddr proto dport"       status=30  idle=120
filter="icmp"        model="saddr daddr proto dport sport" status=60  idle=30
filter="arp"         model="saddr daddr proto dport sport" status=120 idle=60
filter=""            model="saddr daddr proto"             status=300 idle=3600
```

Data Splitting

Argus can generate a lot of data. Tools that help in data disposition are very, very helpful. Here we are providing basic file processing tools, like the unix command split().

- rasplit
  - First designed to support sorting
  - Split based on time, size, count, and contents
  - Time based splitting basis for 1° archive methods
- rastream
  - rasplit + file processing on close

Graphing/Visualization

Network data graphing is a powerful communication tool for report generation, etc..., but it is also the best way to verify and validate the correctness of data processing.

- ragraph
- Perl script processing rabins() output
  - Generating rrd and running rrd_graph
    - Single object / multi metric graphing
    - Strict time-series representations
    - 1 second minimum resolution
  - Variations use GNU Plot, Mathmatica, MatLab
- CSV file generation
  - Data support for many 3rd party systems
    - Numbers, Excel, AfterGlow, PicViz, etc...
    - Issues with date formats - ./support/Config/excel.rc

User Data Processing

- All ra* programs can grep() user data.
- radump
  - Port of tcpdump decoding logic to argus user data buffers.
  - Use this as a prototype for content sensitive analytics.

- rauserdata/raservices
  - Upper protocol discovery, classification, identification, verification
  - Will provide guess for unknown protocols.
  - Experimental, but works very well.

Semantic Enhancement

- ralabel
  - Geospatial Information Merging
    - Country Codes, City, Area Code, Postal Codes, Physical Address, Lat/Lon
  - Netspatial Information Fusion
    - Origin AS Number
    - Domain Name
    - Path Information
- Flow Classification
- Tagging
- ralabel.conf

Anonymization

Network data anonymity is a big topic when considering sharing for research and collaboration.

The strategies used by argus are intended to preserve the information needed to convey the value of the data, and change or throw everything else away.

- ranonymize
  - User Data Capture
  - Time
  - Network Object Identifiers
    - Network Addresses
    - Service Access Point Identifiers (ports)
  - Sequence Numbers

Situational Awareness

Situational Awareness

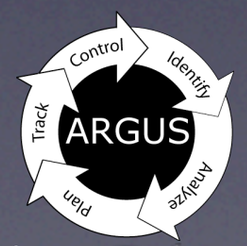
Level 1 SA - Perception

- The perception of elements in the environment within a volume of time and space
- Involves timely sensing, data generation, distribution, collection, combination, filtering, enhancement, processing, storage, retention and access.

Level 2 SA - Comprehension

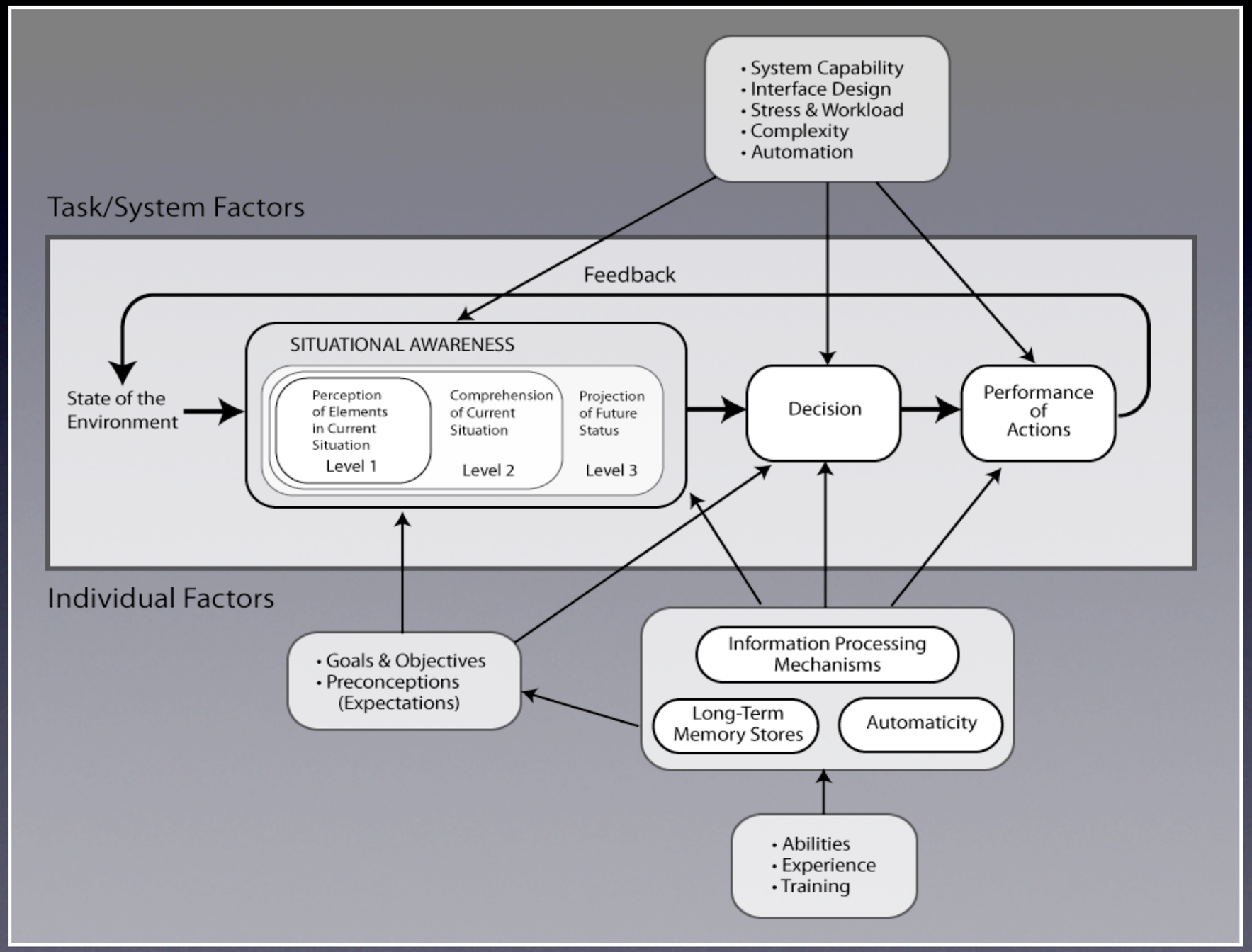
- Understanding significance of perceived elements in relation to relevant goals and objectives.
- Involves integration, correlation, knowledge generation.

Level 3 SA - Projection of Future Status



Endsley, M. R. (1995b). Toward a theory of situation awareness in dynamic systems. Human Factors 37(1), 32-64.

Model of Situational Awareness in Dynamic Decision Making



Situational Awareness System

Basic design is local sensing, data collection and management, with local near real time data processing and large scale data sharing to support multi-dimensional control plane comprehension.

- Federated Database Model
  - Access controlled by local administrative domain (scoping)
  - Cloud-like distributed processing and query support
  - Flexible data management strategies
  - Large numbers of simultaneous users
- Near real-time information availability
  - Register for information of interest
  - Complex data processing / aggregation / enhancement
  - Large scale data correlation processing
  - Anonymization

Network Situational Awareness

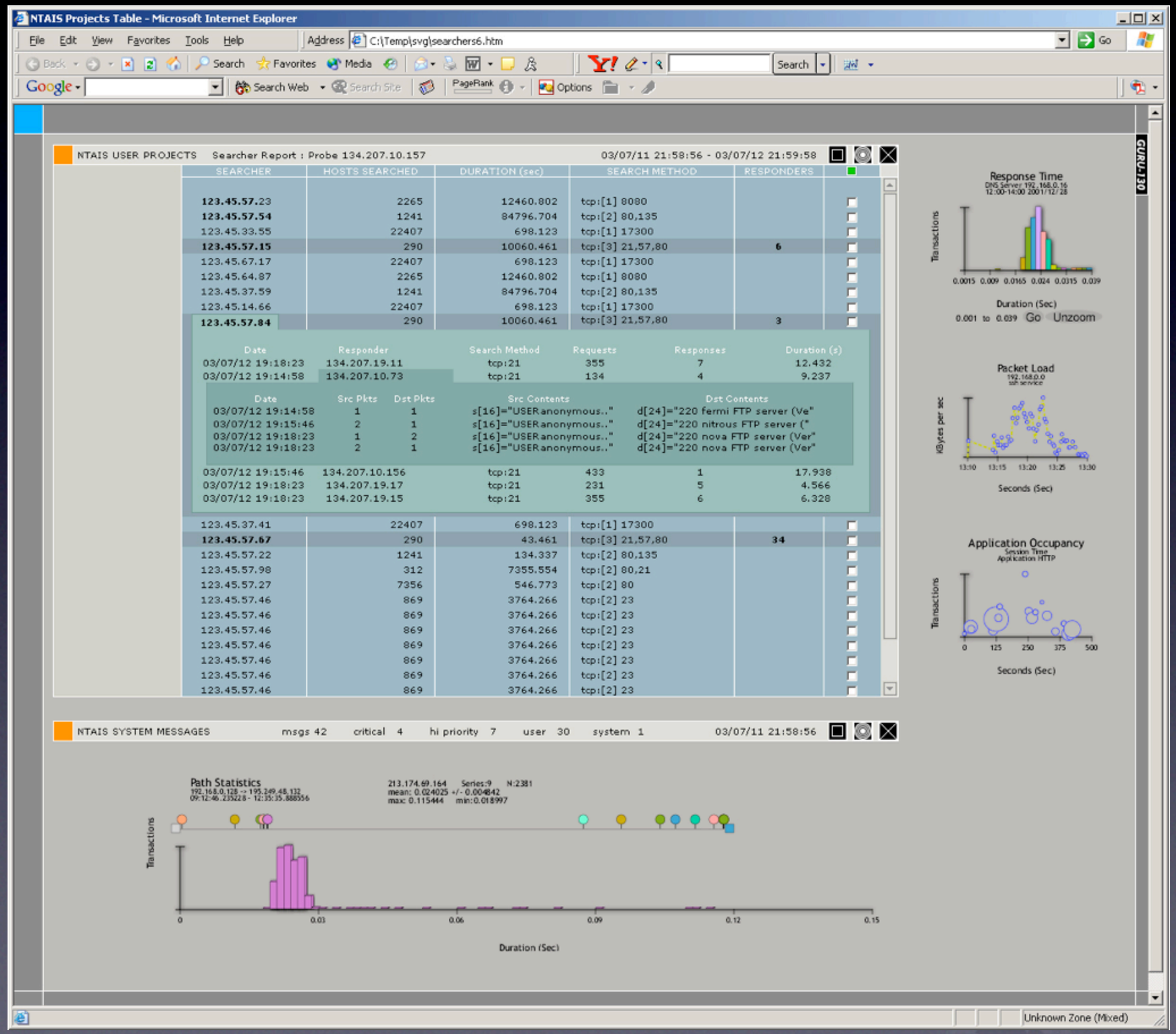
- Argus is designed to be THE network SA sensor
  - Ubiquitously deployable DPI traffic sensor
  - Comprehensive (non-statistical) traffic awareness
  - Provides engineering data, not business intelligence
    - Detailed network transactional performance
    - Network fault identification, discrimination and mitigation
      - Reachability, connectivity, availability, latency, path, flow control etc....
    - Customer gets the primitive data, not just reports/alerts
    - Near realtime and historical capabilities
    - Packet capture replacement
- Supporting a large number of SA applications
  - Advanced Network Functional Assurance (Operations)
    - End-to-End transactional performance tracking (data and control plane)
    - Network component functional assurance (NAT, reachability, encryption)
    - Policy enforcement verification/validation (Access control, path, QoS)
  - Advanced Network Optimization (Security and Performance)
    - Supports network entity and service identification, analysis, planning tracking and control, including baselining, anomaly detection, behavioral analysis and exhaustive forensics

Situational Awareness

- ratop
  - Develop near-realtime view of what is going on in this network, right now.
  - Complex aggregation rule set for data representation
- rasqlinsert
  - ratop functions with state cached in mysql
  - goal is to have the last X minutes available

Situational Awareness

- GUI Strategies
  - Real Time Access
  - Web Based vs Native
  - Drill Down
  - Complex Data Methods
    - Time Series
      - Traditional
    - Spatial Information



Discovery Detection

Network scan detection is not as important as it was decades ago, but understanding who responds to scans, and what they respond with, is still a very important thing to know.

- radark.pl
  - Track IP addresses that attempt to connect to non-existent hosts (a network explorer)
  - If these network addresses ever get a response from existing nodes on non-public service SAPs, then report these accesses.
  - Include what the responder responded with.

Network Perception Goals

- Total Semantic Capture (Comprehension)
  - State initializations and transitions
  - Policy dissemination and enforcement
  - Topology, resource allocations, error conditions
- Context Awareness
  - Multi-Layer Identifiers (ethernet, MPLS labels, etc....)
  - Globally synchronized uSec timestamps
- Enable Near Real-Time State Awareness
  - Large scale access and data sharing
  - Multi-dimensional Correlation
- Complete Historical Reconstruction

Network Sensing Strategy

- Third-party Control/Data Plane monitor/sensors
  - Can't rely on the network switch/router vendors to do this.
- Each network device must provide complete Data and Control Plane packet capture
  - Any packet that originates from or terminates on the device must be captured in its entirety.
  - Data must include port of origination/transmission, direction and UTC time stamp.
  - Before and after any encryption/decryption.
- Packet data is converted to flow data for sharing, status reporting, and archival.
- Now we have the data we need to drive Data/ Control Plane Situational Awareness.

Packet / Flow Strategies

- Packet data for complete comprehension
- Flow data provides multi-tiered data model.
  - Data Reduction / Semantic Preservation
    - Service Oriented Transactional Abstractions
    - Complex Data Representations
    - Flexible Compression Strategies
    - Multiple Flow Content Representations
  - Semantic Access Control Schemes
    - Inter/Intra Domain Data Sharing
    - Complex Data Aggregation Scoping
    - Anonymization
  - Cross Domain/Dimensional Correlation
    - Unified Object Specifications
    - Self-Synchronization Methodologies
    - High Resolution Timestamping

Network Information Model

- Multi-tiered Information Model
  - Not every application needs the same type of information
  - System needs to allow "customer" to define what it wants
- And, as conditions change, level of detail and frequency of status reports also needs to change

1. Data/Control Plane Service Existence Flow Strategy
   1.1. Matrix Flow with Service Identifiers
   1.2. Operational/Security Fault Status Flow Records
2. Data/Control Plane Service Performance Strategy
   2.1. Transactional Flow with Ops and Performance Attributes
   2.2. Operational Fault Status Flow Records
3. Total Packet Content Flow Strategy
   3.1. Transactional Flow with Aggregated Content
   3.2. Complete Remote Packet Capture

Building Real Time Systems

Supporting Slides

Argus Approach

- Use formal network activity audit models
  - Network service oriented
    - Initiation, status, termination state indications for complex flows
    - Bidirectional realtime network flow traffic monitoring
    - For all services - ARP, DHCP, DNS, TCP, UDP,VoIP, P2P....
  - Convey as much about the traffic as possible
    - Deep packet inspection strategies to extract traffic semantics
    - Tunnel identifiers (MPLS,VLAN, Ethernet, IPnIP, GRE, RTP, Teredo)
    - Reachability, connectivity, availability, rate, load, latency, loss, jitter, packet size distribution
    - Security issue reporting (protocol issues i.e. fragmentation overlap attack, tunnel hopping)
    - Controlled content capture
- Flexible transport, collection and storage strategies
- Data processing tools
  - Aggregation, analysis, anonymization, filtering, graphing, … , zipping.
  - Native OS archive management, MySQL database, 3rd party integration
- Realizable audit data resource requirements
  - CMU generates ~100-300 GB/day of 'primitive' data
  - Naval Research Lab retains ~ 50-100 GB/day
  - QoSient.com manages 120 MB/day

Benefits of Argus

- Argus provides excellent data
  - Data drives many applications
    - Advanced network activity metrics
    - Rich security information model
  - Real time access
- Deployable throughout the infrastructure
  - High Performance - 10 Gbps using Endace, Bivio, etc....
  - End Systems - Unix, Linux, Mac OS X, Windows (Cygwin), AIX, IRIX, HPUX, Ericsson ViPR
  - Router Systems - VxWorks, DIY Routers, OpenWRT
- Rich data collection, management and processing
  - Key differentiator from commercial offerings
    - Many advanced sites want their own data analysis
    - All want data processing and reporting extensibility

Where are we headed?

- Distributed Network Auditing
  - Very Large Scale Situational Awareness
    - Auditing system scalability using cloud architectures
  - Complete end-to-end capability
  - Automated Attribution
  - New security mechanisms
- Sensor Improvements
  - Higher performance - multi-core
  - More Control Plane Auditing
    - OSPF, BGP, , SIP ...
  - Wireless
- Audit system applications
  - Real-time situational awareness
  - Security forensics tools

Network Activity Driven Feedback
Directed Optimization

Function	Description	
Identify	Discover and Identify comprehensive network behavior	Collect and process network behavioral data
Analyze	Collect and transform data into optimization metrics, establish baselines occurrence probabilities and prioritize events.	
Plan	Establish optimization criteria, both present and future and implement actions, if needed	Provide information and feedback internal and external to the project on the optimization outcomes as events.
Track	Monitor network behavioral indicators to realize an effect.	
Control	Correct for deviations from criteria.