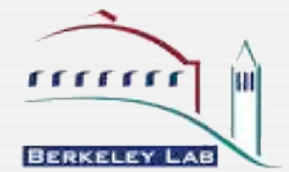
The Open Source Bro IDS
Overview and Recent Developments

Robin Sommer
International Computer Science Institute, &
Lawrence Berkeley National Laboratory

robin@icsi.berkeley.edu
http://www.icir.org/robin

2011 CACR Higher Education Cybersecurity Summit
Indiana University



Outline

Outline

Philosophy and Architecture

A *framework* for network traffic analysis.

Outline

Philosophy and Architecture

A *framework* for network traffic analysis.

Usage and Deployment

Logs, scripts, communication, cluster.

Outline

Philosophy and Architecture

A *framework* for network traffic analysis.

Usage and Deployment

Logs, scripts, communication, cluster.

Roadmap

The Bro world is moving quite rapidly.

What is Bro?

What is Bro?

Packet Capture

What is Bro?

Packet Capture

Traffic Inspection

What is Bro?

Packet Capture

Traffic Inspection

Attack Detection

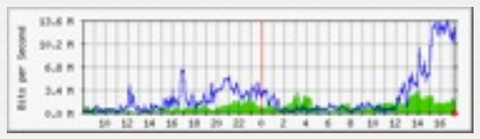
What is Bro?

Packet Capture

Traffic Inspection

Attack Detection

NetFlow



Log Recording

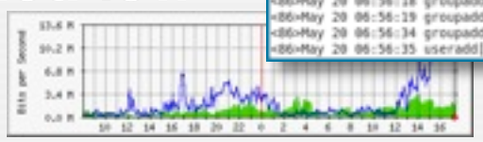
What is Bro?

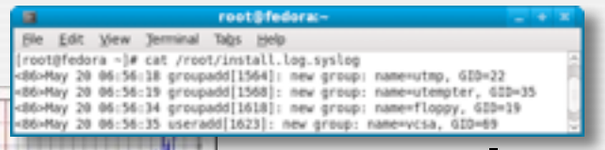
Packet Capture

Traffic Inspection

Attack Detection

NetFlow





syslog

Log Recording

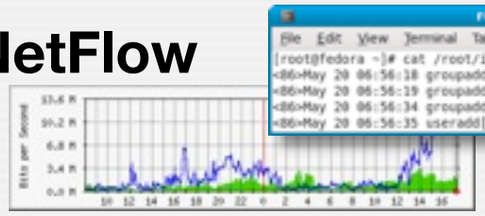
What is Bro?

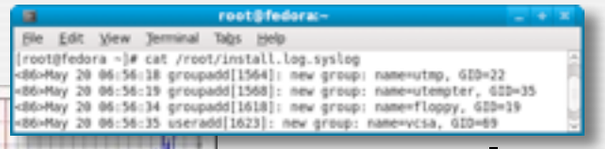
Packet Capture

Traffic Inspection

Attack Detection

NetFlow

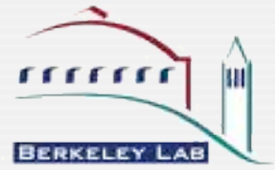




syslog

Log Recording

Flexibility
Abstraction
Data Structures



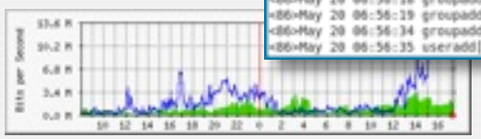
What is Bro?

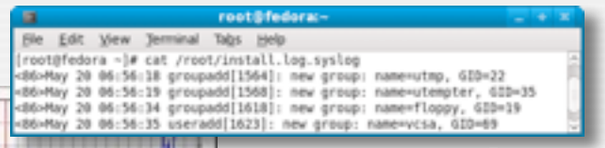
Packet Capture

Traffic Inspection

Attack Detection

NetFlow

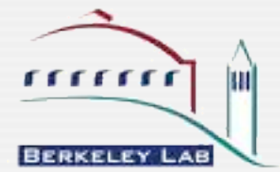




syslog

Log Recording

Flexibility
Abstraction
Data Structures



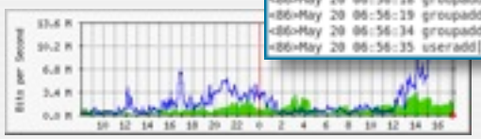
What is Bro?

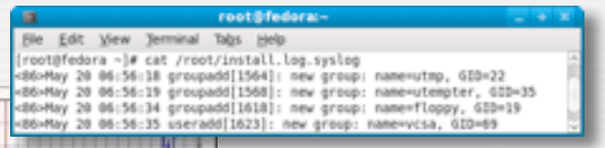
Packet Capture

Traffic Inspection

Attack Detection

NetFlow

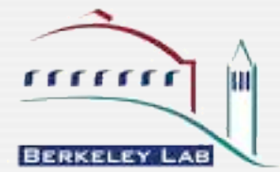




syslog

Log Recording

Flexibility
Abstraction
Data Structures



What is Bro?

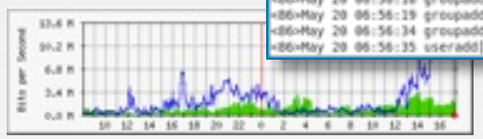
Packet Capture

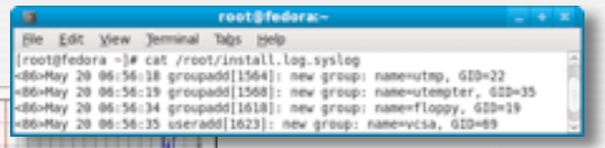
Traffic Inspection

"Domain-specific Python"

Attack Detection

NetFlow

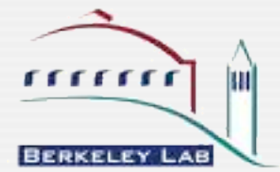




syslog

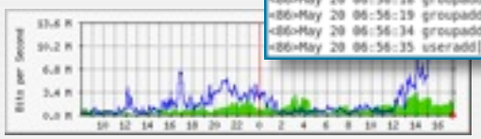
Log Recording

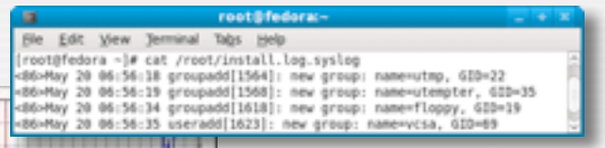
Flexibility
Abstraction
Data Structures



What is Bro?

NetFlow





syslog

Packet Capture

Traffic Inspection

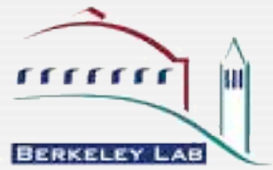
Attack Detection

Log Recording

Flexibility
Abstraction
Data Structures

Sum is more than the pieces

"Domain-specific Python"



Philosophy

Philosophy

Real-time network analysis *framework*

Primarily an IDS, but many use it for general traffic analysis.

Philosophy

Real-time network analysis *framework*

Primarily an IDS, but many use it for general traffic analysis.

Highly stateful

Tracks extensive application-layer network state.

Philosophy

Real-time network analysis *framework*

Primarily an IDS, but many use it for general traffic analysis.

Highly stateful

Tracks extensive application-layer network state.

Policy-neutral at the core

Can accommodate a range of detection approaches.

Philosophy

Real-time network analysis *framework*

Primarily an IDS, but many use it for general traffic analysis.

Highly stateful

Tracks extensive application-layer network state.

Policy-neutral at the core

Can accommodate a range of detection approaches.

Supports forensics

Extensively logs what it sees.

Target Audience

Target Audience

Scientific environments

Effective with liberal security policies

Target Audience

Scientific environments

Effective with liberal security policies

Network-savvy users

Requires understanding of your network

Target Audience

Scientific environments

Effective with liberal security policies

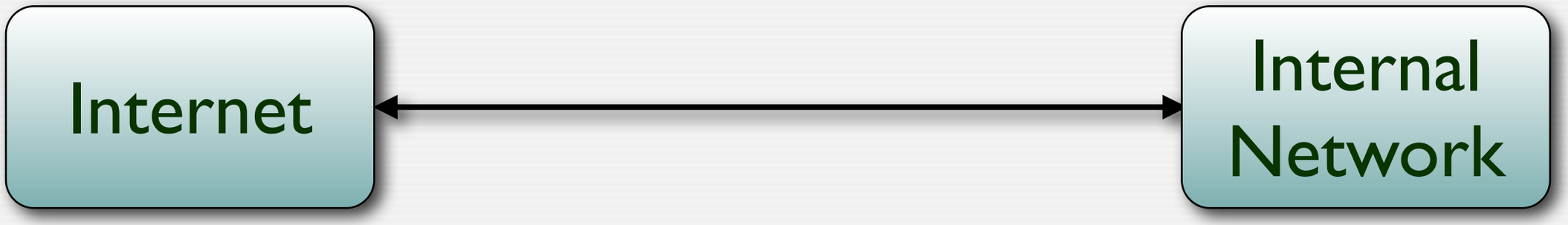
Network-savvy users

Requires understanding of your network

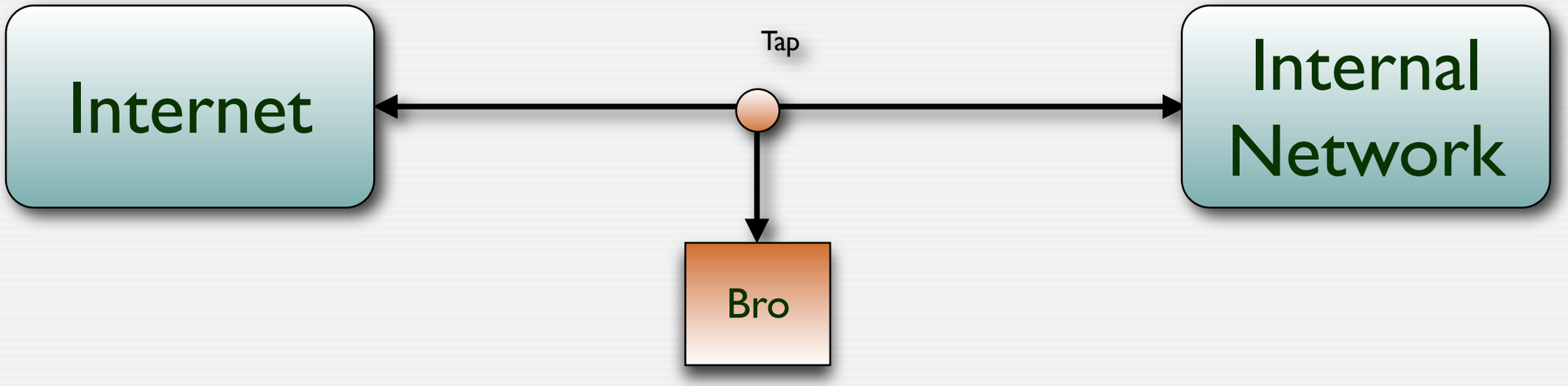
Unixy mindset

Command-line based, fully customizable

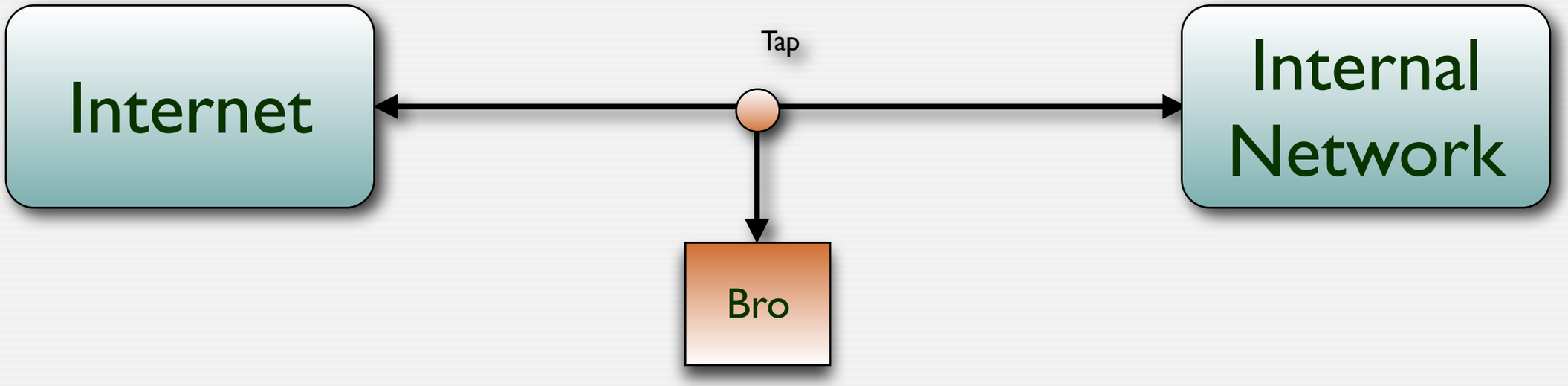
Deployment



Deployment



Deployment



Commodity platforms

Standard PC & NICs, FreeBSD/Linux

Activity Logs: TCP

Activity Logs: TCP

One-line summaries of all connections

Most basic, yet often extremely helpful output.

Activity Logs: TCP

One-line summaries of all connections

Most basic, yet often extremely helpful output.

```
> bro -i en0 tcp
[ ... wait ...]
> cat conn.log
```

Activity Logs: TCP

One-line summaries of all connections

Most basic, yet often extremely helpful output.

```
> bro -i en0 tcp
[ ... wait ...]
> cat conn.log
```

```
Time                  Duration  Source           Destination
1144876596.658302  1.206521  192.150.186.169  62.26.220.2 \
  http  53052     80         tcp     874         1841        SF      X
  Service SPort   DPort      Proto   SrcBytes    DstBytes    TCPState Local?
```

Activity Logs: TCP

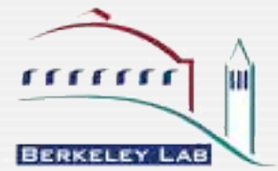
One-line summaries of all connections

Most basic, yet often extremely helpful output.

```
> bro -i en0 tcp
[ ... wait ...]
> cat conn.log
```

```
Time                  Duration  Source           Destination
1144876596.658302  1.206521  192.150.186.169  62.26.220.2 \
  http  53052     80        tcp     874         1841        SF      X
  Service SPort   DPort     Proto   SrcBytes    DstBytes    TCPState Local?
```

LBNL has connection logs for every connection attempt since June 94!



Activity Logs: HTTP

```
> cat http.log
[...]
1144876588.30 %2 start 192.150.186.169:53041 > 195.71.11.67:80
1144876588.30 %2 GET /index.html (200 "OK" [57634] www.spiegel.de)
1144876588.30 %2 > HOST:  www.spiegel.de
1144876588.30 %2 > USER-AGENT:  Mozilla/5.0 (Macintosh; PPC Mac OS ...
1144876588.30 %2 > ACCEPT:  text/xml,application/xml,application/xhtml ...
1144876588.30 %2 > ACCEPT-LANGUAGE:  en-us,en;q=0.7,de;q=0.3
[...]
1144876588.77 %2 < SERVER:  Apache/1.3.26 (Unix) mod_fastcgi/2.2.12
1144876588.77 %2 < CACHE-CONTROL:  max-age=120
1144876588.77 %2 < EXPIRES:  Wed, 12 Apr 2006 21:18:28 GMT
[...]
1144876588.77 %2 <= 1500 bytes: "<!-- Vignette StoryServer 5.0 Wed Apr..."
1144876588.78 %2 <= 1500 bytes: "r "http://spiegel.ivwbox.de" r..."
1144876588.78 %2 <= 1500 bytes: "icon.ico" type="image/ico">^M^J ..."
1144876588.94 %2 <= 1500 bytes: "erver 5.0 Mon Mar 27 15:56:55 ..."
[...]
```

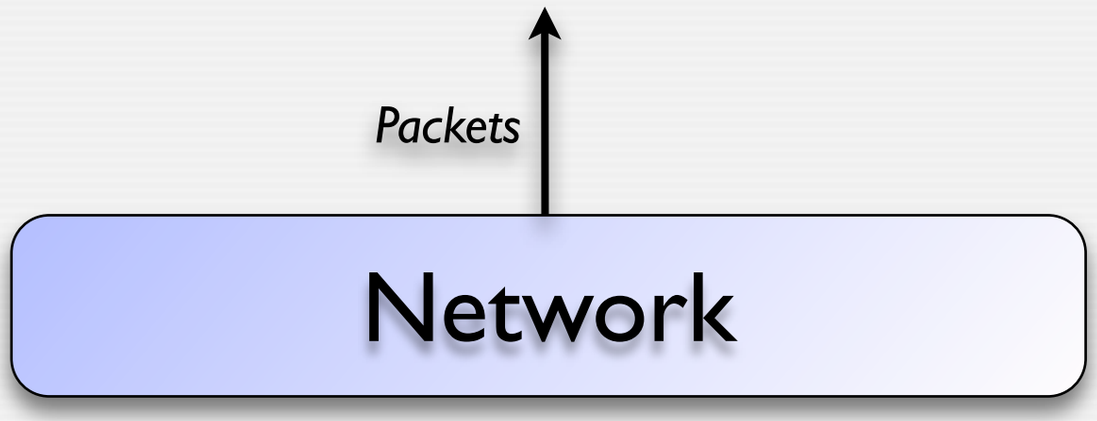
Activity Logs: HTTP

```
> cat http.log
[...]
1144876588.30 %2 start 192.150.186.169:53041 > 195.71.11.67:80
1144876588.30 %2 GET /index.html (200 "OK" [57634] www.spiegel.de)
1144876588.30 %2 > HOST:  www.spiegel.de
1144876588.30 %2 > USER-AGENT:  Mozilla/5.0 (Macintosh; PPC Mac OS ...
1144876588.30 %2 > ACCEPT:  text/xml,application/xml,application/xhtml ...
1144876588.30 %2 > ACCEPT-LANGUAGE:  en-us,en;q=0.7,de;q=0.3
[...]
1144876588.77 %2 < SERVER:  Apache/1.3.26 (Unix) mod_fastcgi/2.2.12
1144876588.77 %2 < CACHE-CONTROL:  max-age=120
1144876588.77 %2 < EXPIRES:  Wed, 12 Apr 2006 21:18:28 GMT
[...]
1144876588.77 %2 <= 1500 bytes: "<!-- Vignette StoryServer 5.0 Wed Apr..."
1144876588.78 %2 <= 1500 bytes: "r "http://spiegel.ivwbox.de" r..."
1144876588.78 %2 <= 1500 bytes: "icon.ico" type="image/ico">^M^J ..."
1144876588.94 %2 <= 1500 bytes: "erver 5.0 Mon Mar 27 15:56:55 ..."
[...]
```

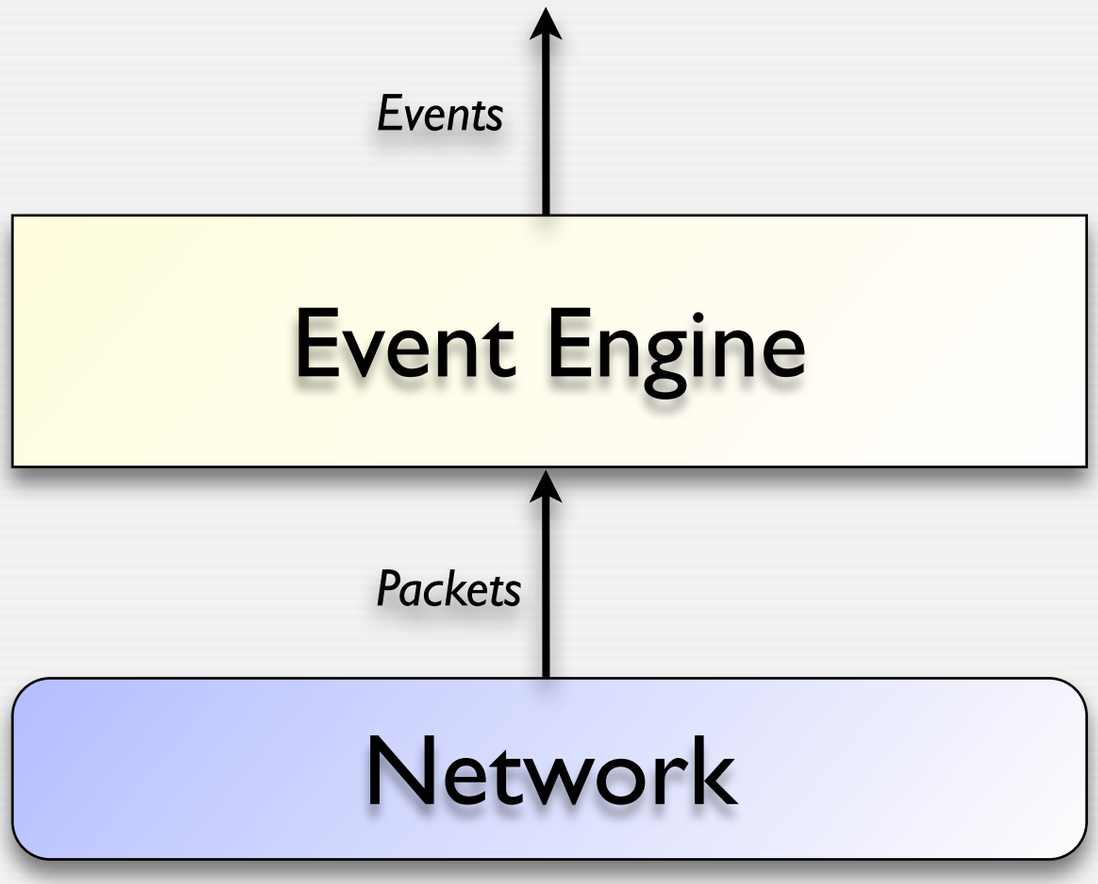
Supported Protocols

ARP, UDP, TCP, ICMP
BitTorrent, DNS, FTP, Gnutella, HTTP, IRC, NFS, POP3, RPC, SMB, SMTP, SSH, SSL
... and more

Architecture



Architecture

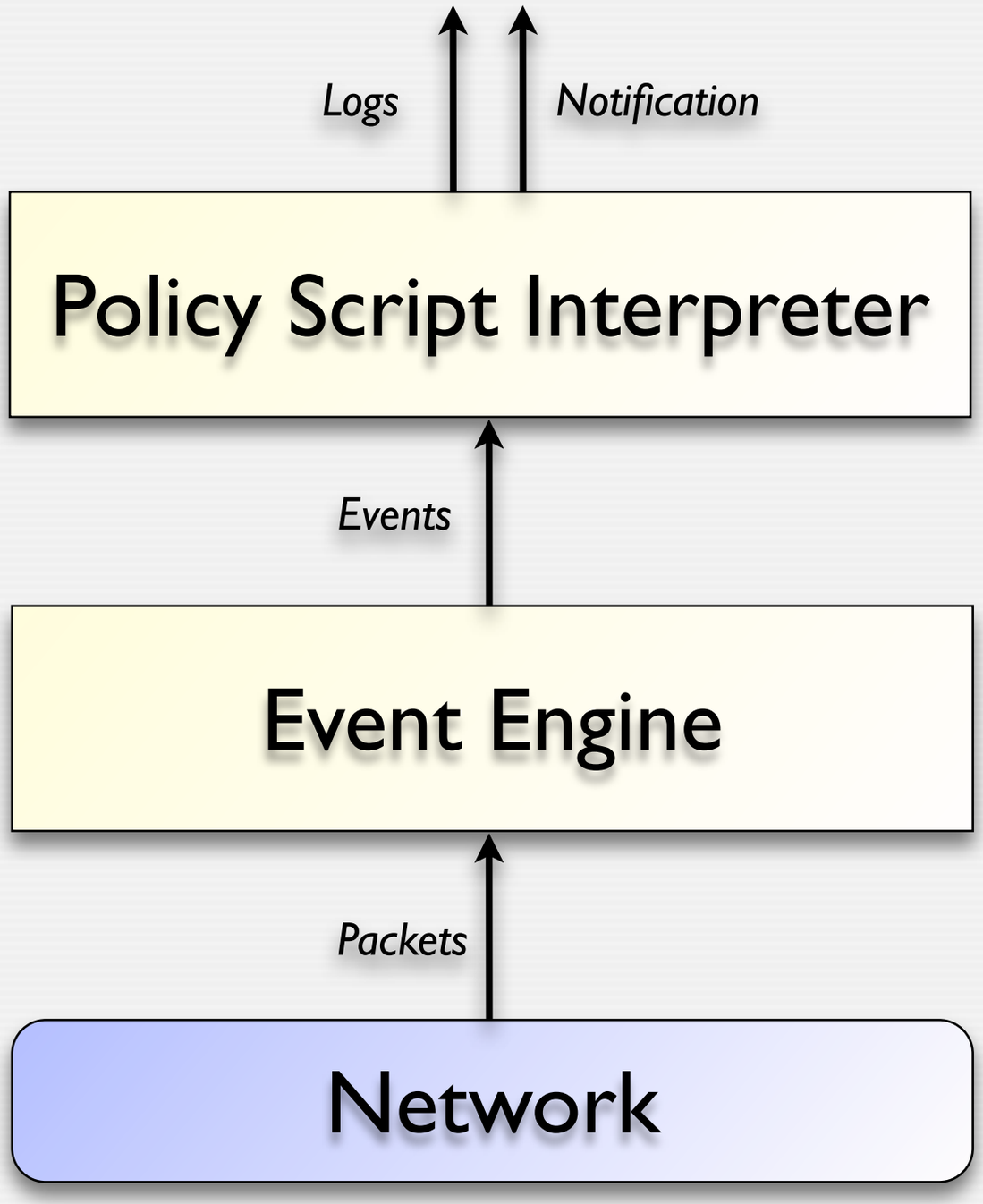


Protocol Decoding

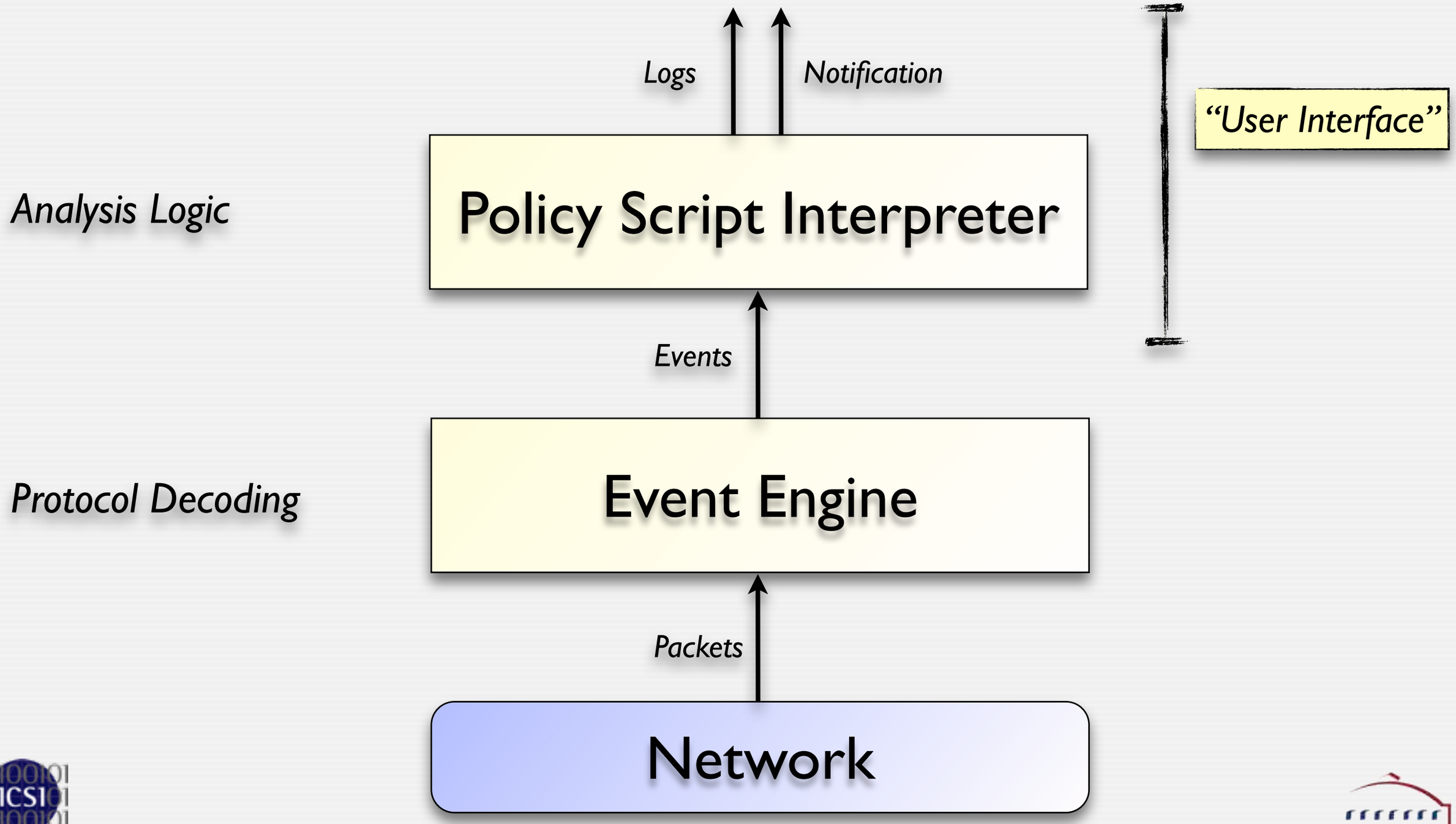
Architecture

Analysis Logic

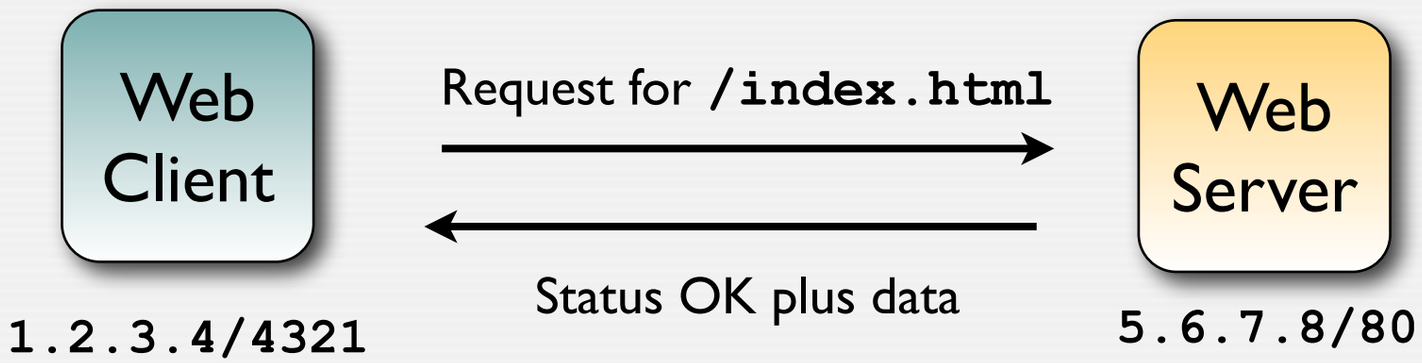
Protocol Decoding



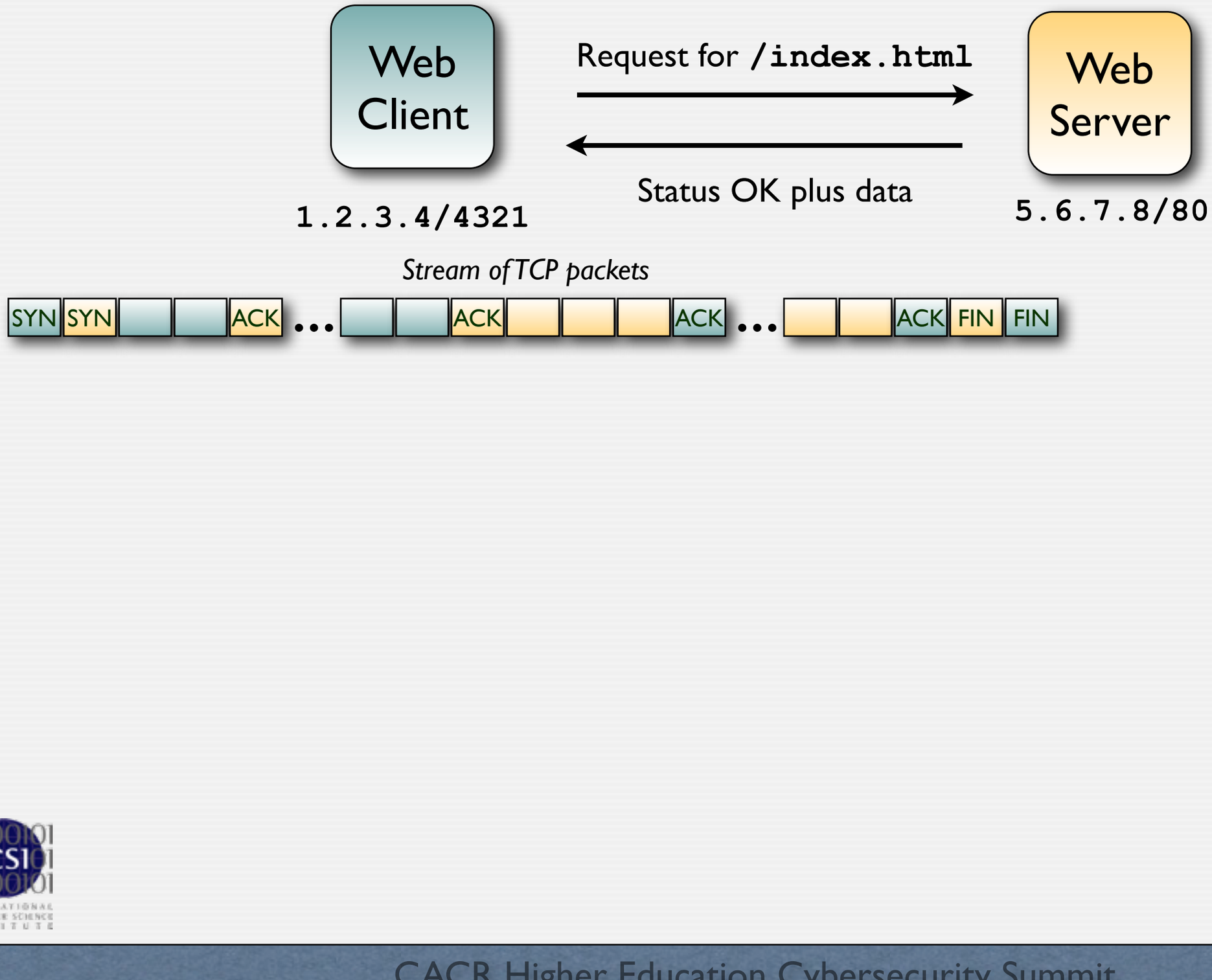
Architecture



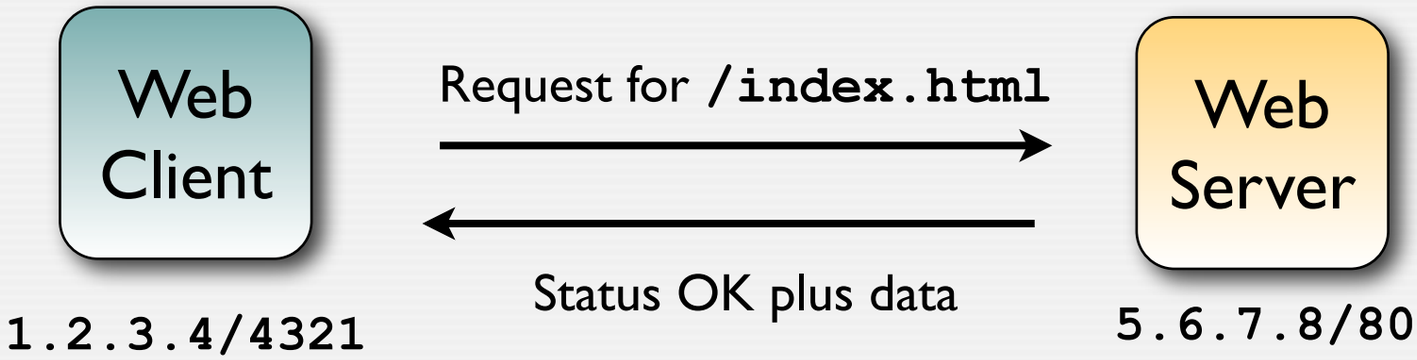
Event Model


Web Client
Request for /index.html
Web Server
Status OK plus data
1.2.3.4/4321
5.6.7.8/80

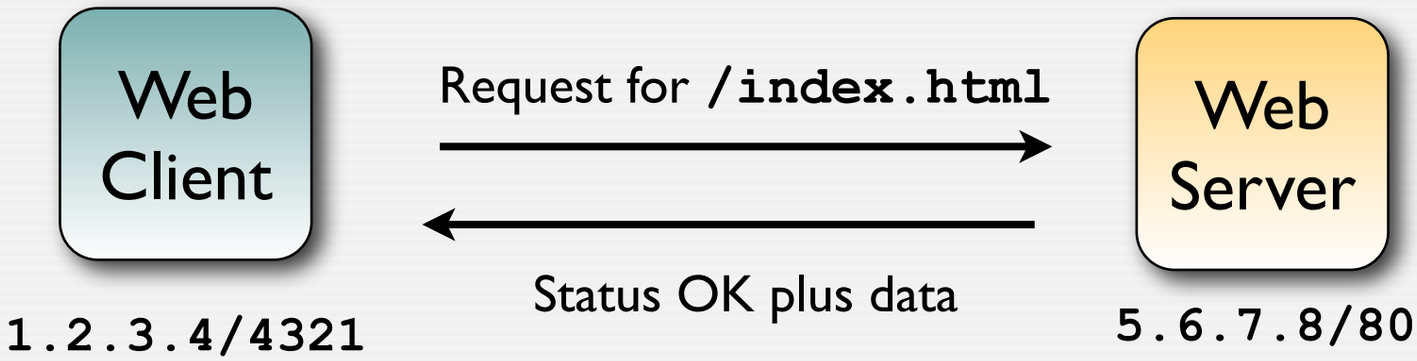
Event Model


Web Client
Request for /index.html
Web Server
Status OK plus data
1.2.3.4/4321
5.6.7.8/80
Stream of TCP packets
SYN
SYN
ACK
ACK
ACK
ACK
FIN
FIN

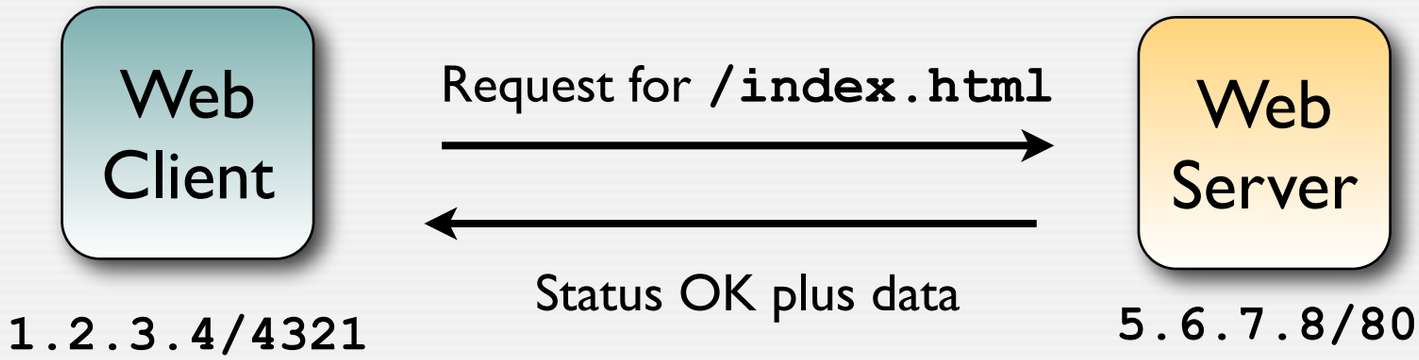
Event Model


Web Client
Request for /index.html
Web Server
Status OK plus data
1.2.3.4/4321
5.6.7.8/80
Stream of TCP packets
SYN
SYN
ACK
ACK
ACK
ACK
FIN
FIN
Event
connection established(1.2.3.4/4321⇒5.6.7.8/80)

Event Model


Web Client
Request for /index.html
Web Server
Status OK plus data
1.2.3.4/4321
5.6.7.8/80
Stream of TCP packets
SYN
SYN
ACK
ACK
ACK
ACK
FIN
FIN
Event
connection established(1.2.3.4/4321⇒5.6.7.8/80)
TCP stream reassembly for originator
Event
http_request(1.2.3.4/4321⇒5.6.7.8/80, "GET", "/index.html")

Event Model



Stream of TCP packets

Event → `connection_established(1.2.3.4/4321⇒5.6.7.8/80)`

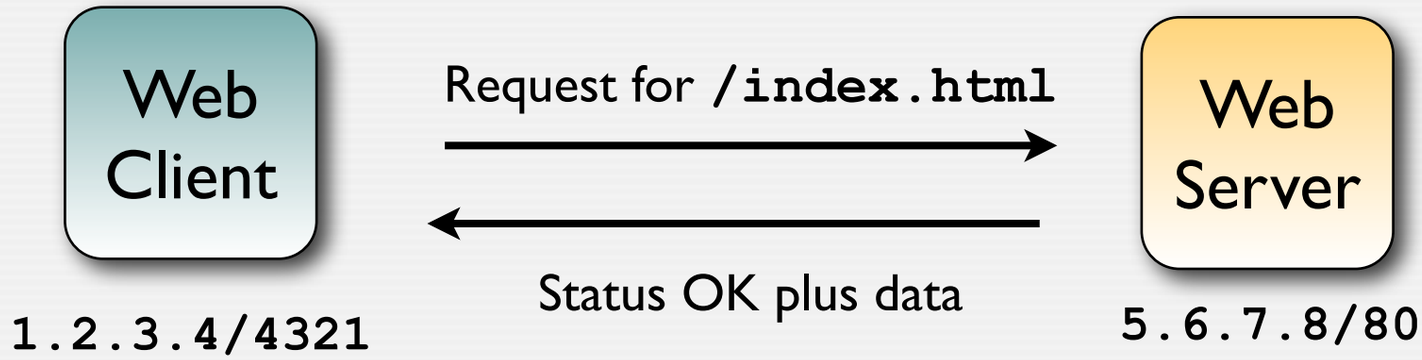
TCP stream reassembly for originator

Event → `http_request(1.2.3.4/4321⇒5.6.7.8/80, "GET", "/index.html")`

TCP stream reassembly for responder

Event → `http_reply(1.2.3.4/4321⇒5.6.7.8/80, 200, "OK", data)`

Event Model



Stream of TCP packets

Event → `connection established(1.2.3.4/4321⇒5.6.7.8/80)`

TCP stream reassembly for originator

Event → `http_request(1.2.3.4/4321⇒5.6.7.8/80, "GET", "/index.html")`

TCP stream reassembly for responder

Event → `http_reply(1.2.3.4/4321⇒5.6.7.8/80, 200, "OK", data)`

Event → `connection_finished(1.2.3.4/4321, 5.6.7.8/80)`

Script Example: Matching URLs

Task: Report all Web requests for files called "`passwd`".

Example: *`http_request(1.2.3.4/4321⇒5.6.7.8/80, "GET", "/index.html")`*

Script Example: Matching URLs

Task: Report all Web requests for files called "`passwd`".

```
event http_request(c: connection, method: string, path: string)
{
    if ( method == "GET" && path == /.*passwd/ )
        NOTICE(SensitiveURL, c, path); # Alarm.
}
```

(Syntax simplified.)

Example: *`http_request(1.2.3.4/4321⇒5.6.7.8/80, "GET", "/index.html")`*

Script Example: Matching URLs

Task: Report all successful Web requests for files called "`passwd`".

```
event http_request(c: connection, method: string, path: string)
{
    if ( method == "GET" && path == /.*passwd/ )
        NOTICE(SensitiveURL, c, path); # Alarm.
}
```

(Syntax simplified.)

Example: *`http_request(1.2.3.4/4321⇒5.6.7.8/80, "GET", "/index.html")`*

Script Example: Matching URLs

Task: Report all successful Web requests for files called "`passwd`".

```
global potentially_sensitive: table[connection] of string;

event http_request(c: connection, method: string, path: string)
{
    if ( method == "GET" && path == /.*passwd/ )
        potentially_sensitive[c] = path; # Add to table.
}
```

(Syntax simplified.)

Example: *http_request(1.2.3.4/4321⇒5.6.7.8/80, "GET", "/index.html")*

Script Example: Matching URLs

Task: Report all successful Web requests for files called "`passwd`".

```
global potentially_sensitive: table[connection] of string;

event http_request(c: connection, method: string, path: string)
{
    if ( method == "GET" && path == /.*passwd/ )
        potentially_sensitive[c] = path; # Add to table.
}

event http_reply(c: connection, response: int, reason: string)
{
    if ( response == OK && c in potentially_sensitive )
        NOTICE(SensitiveURL, c, potentially_sensitive[c]);
}
```

(Syntax simplified.)

Example: *http_request(1.2.3.4/4321⇒5.6.7.8/80, "GET", "/index.html")*

Script Example: Scan Detector

Task: *Count failed connection attempts per source address*.

Script Example: Scan Detector

Task: *Count failed connection attempts per source address.*

```
global attempts: table[addr] of int &default=0;

event connection_rejected(c: connection)
{
    local source = c$orig_h;        # Get source address.
    local n = ++attempts[source]; # Increase counter.
    if ( n == SOME_THRESHOLD )     # Check for threshold.
        NOTICE(Scanner, source);   # If so, report.
}
```

(Syntax simplified.)

Script Example: Tracking Services

Task: *Report hosts accepting an SSH connection for the first time*

Script Example: Tracking Services

Task: *Report hosts accepting an SSH connection for the first time*

```
global ssh_hosts: set[addr];

event connection_established(c: connection)
{
    local responder = c$id$resp_h; # Responder's address
    local service = c$id$resp_p;   # Responder's port

    if ( service != 22/tcp )
        return; # Not SSH.

    if ( responder in ssh_hosts )
        return; # We already know this one.

    NOTICE(SSHHostFound, responder); # Alarm
    add ssh_hosts[responder]; # Found a new host
}
```

(Syntax simplified.)

Distributed Scripts

Distributed Scripts

Bro comes with >20,000 lines of script

Prewritten functionality that can just be loaded

Distributed Scripts

Bro comes with >20,000 lines of script

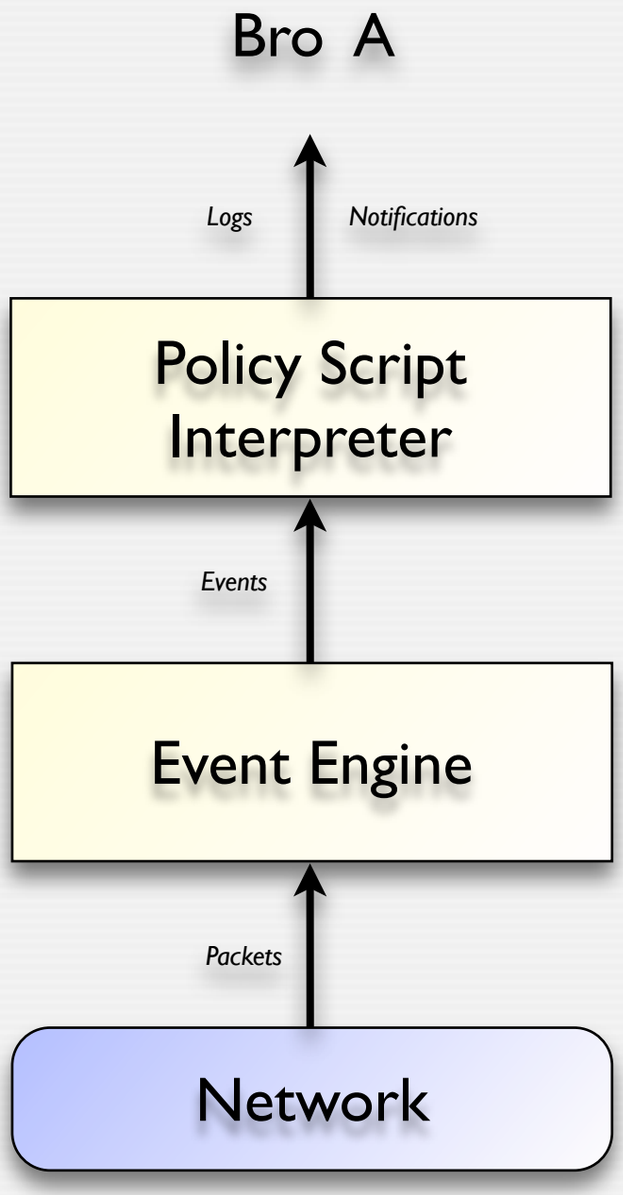
Prewritten functionality that can just be loaded

Scripts generate logs and alarms

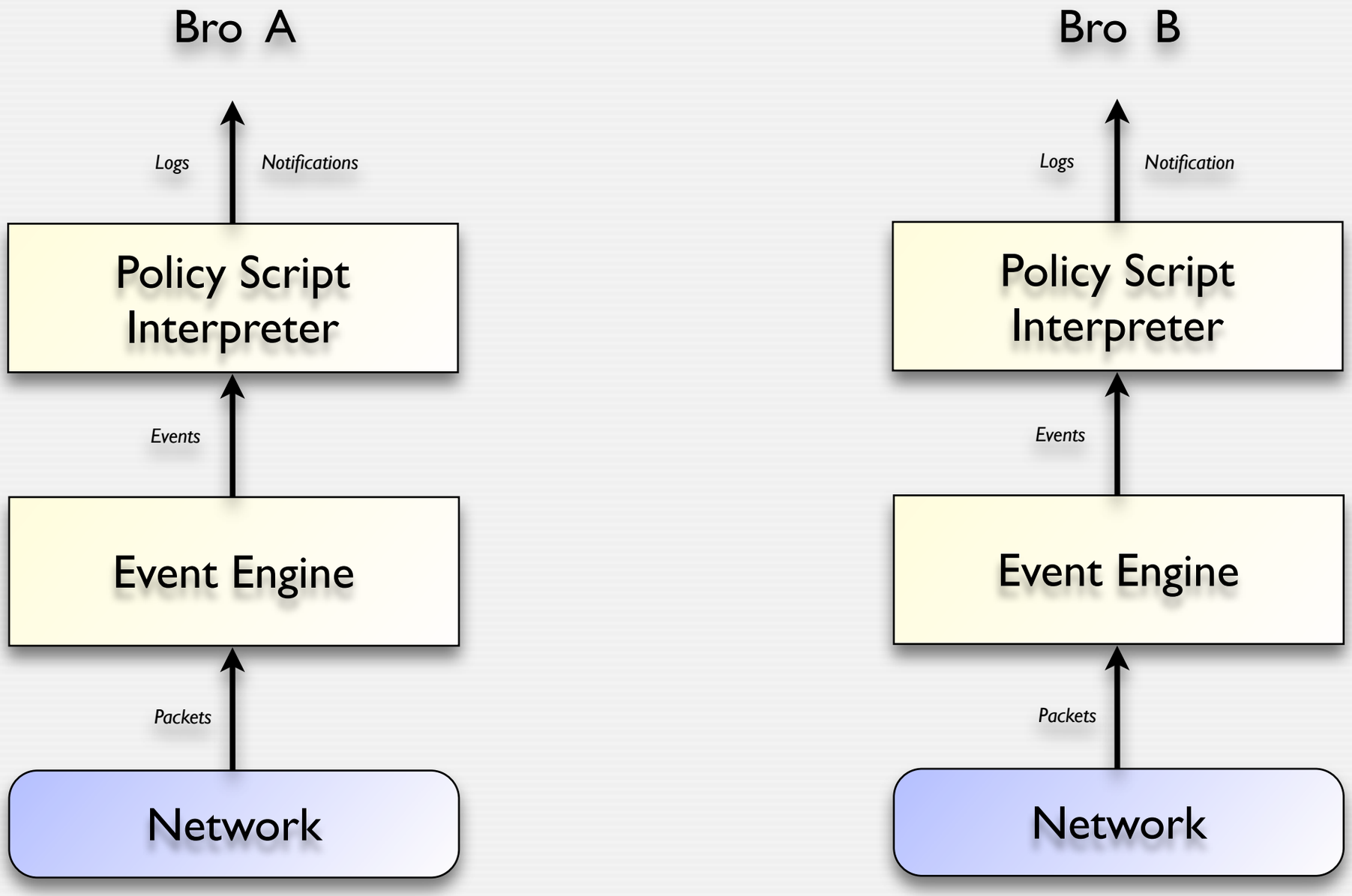
Amendable to customization and extension

Communication Architecture

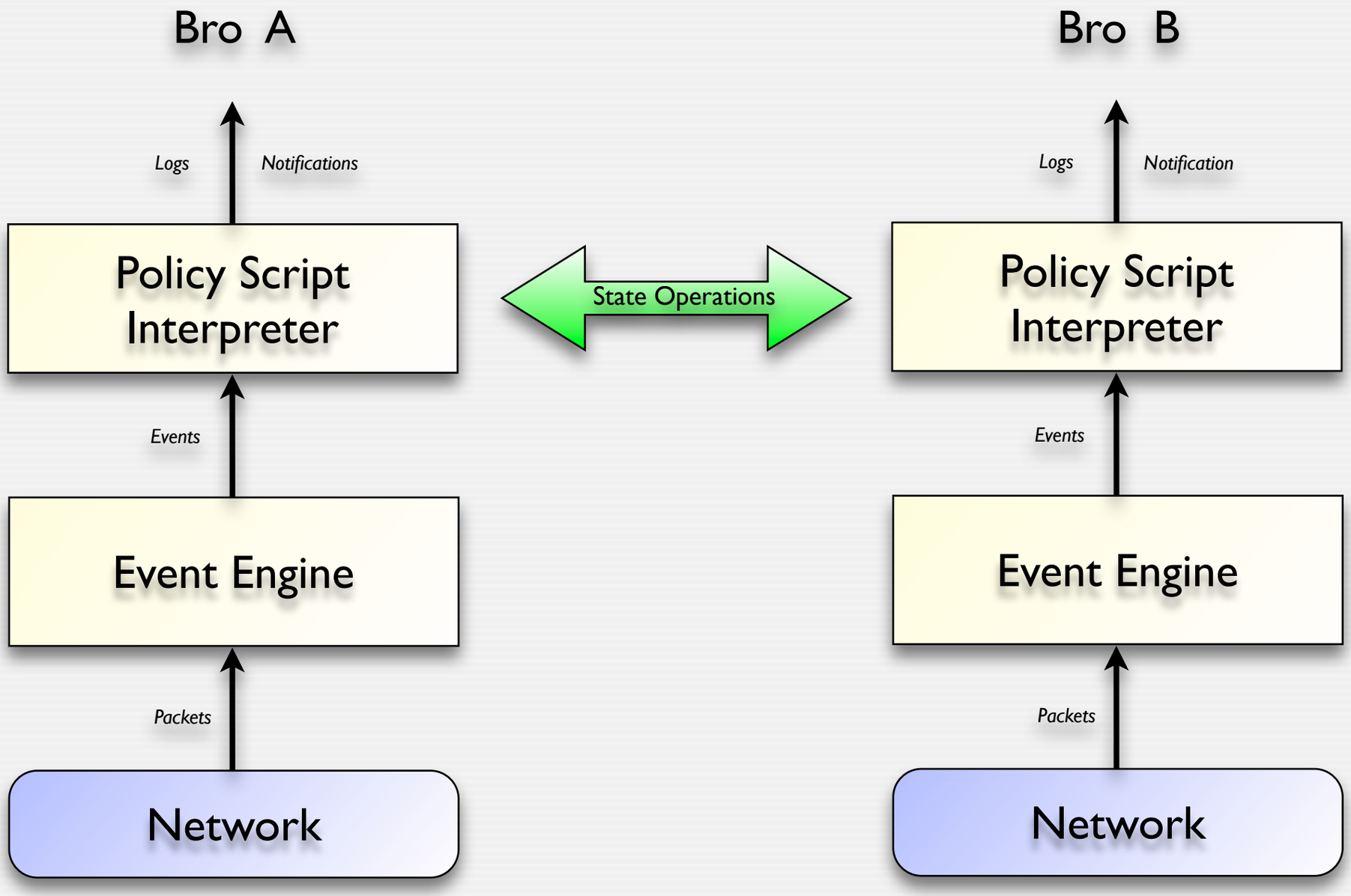
Communication Architecture



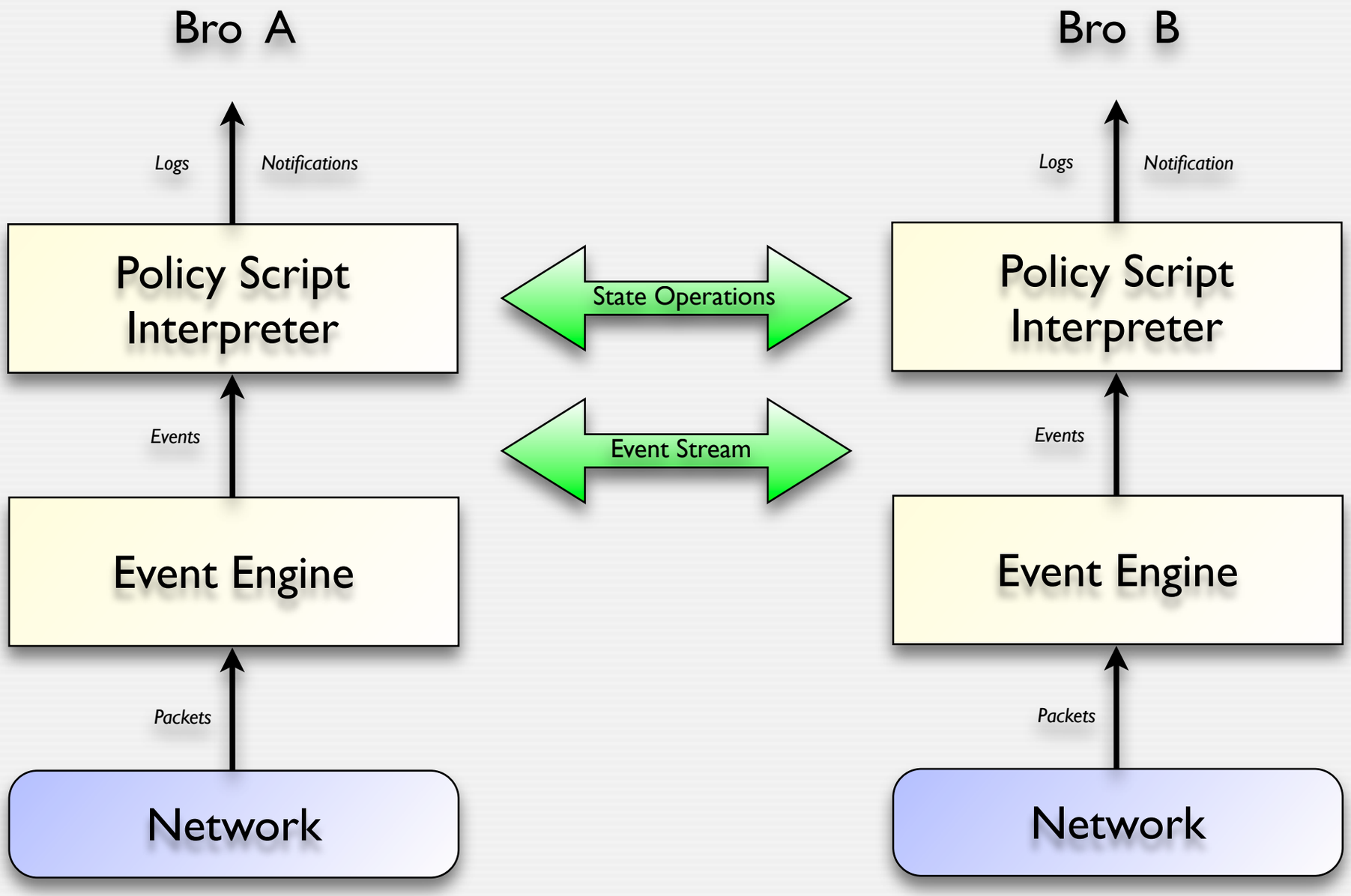
Communication Architecture



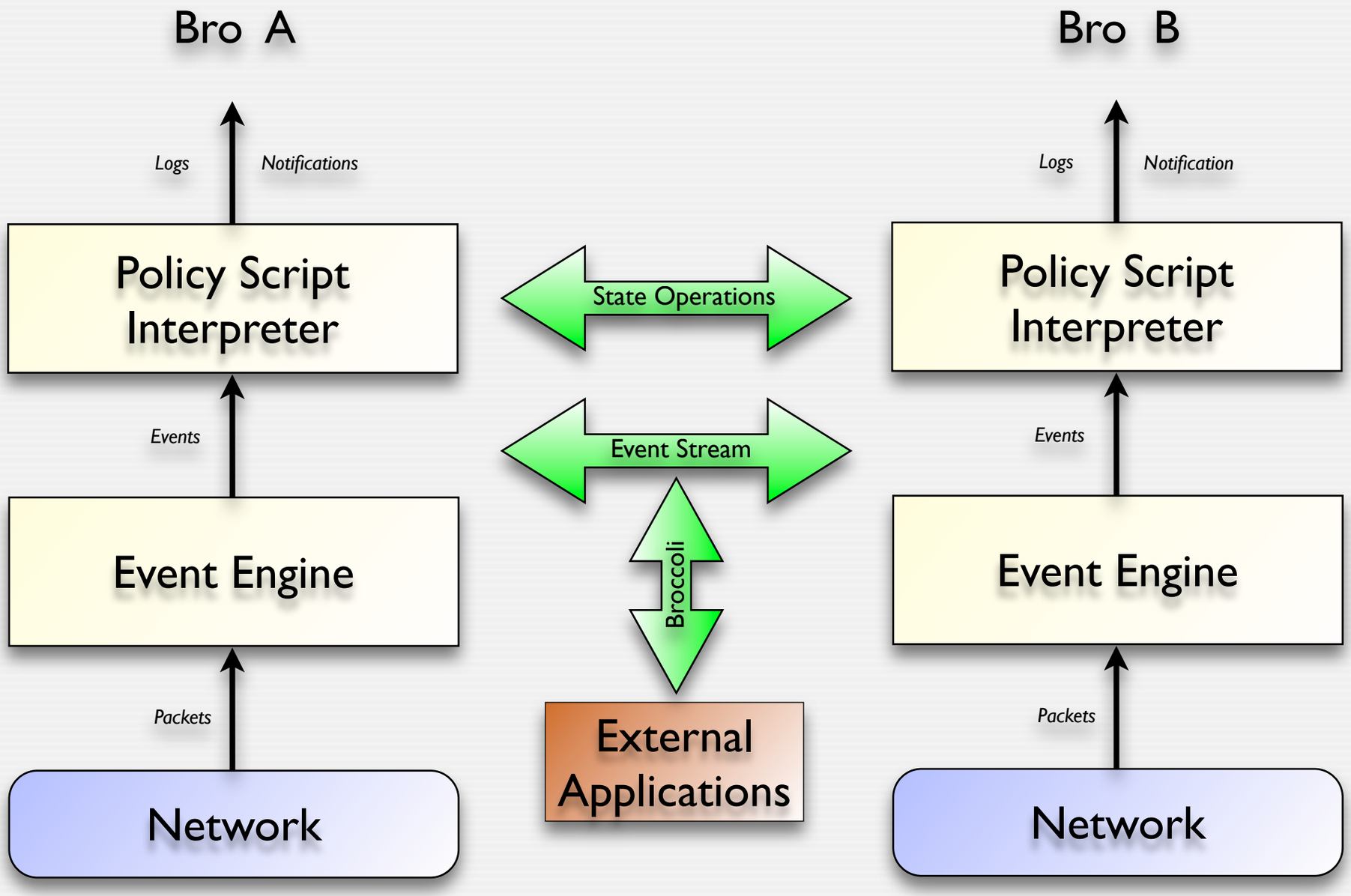
Communication Architecture



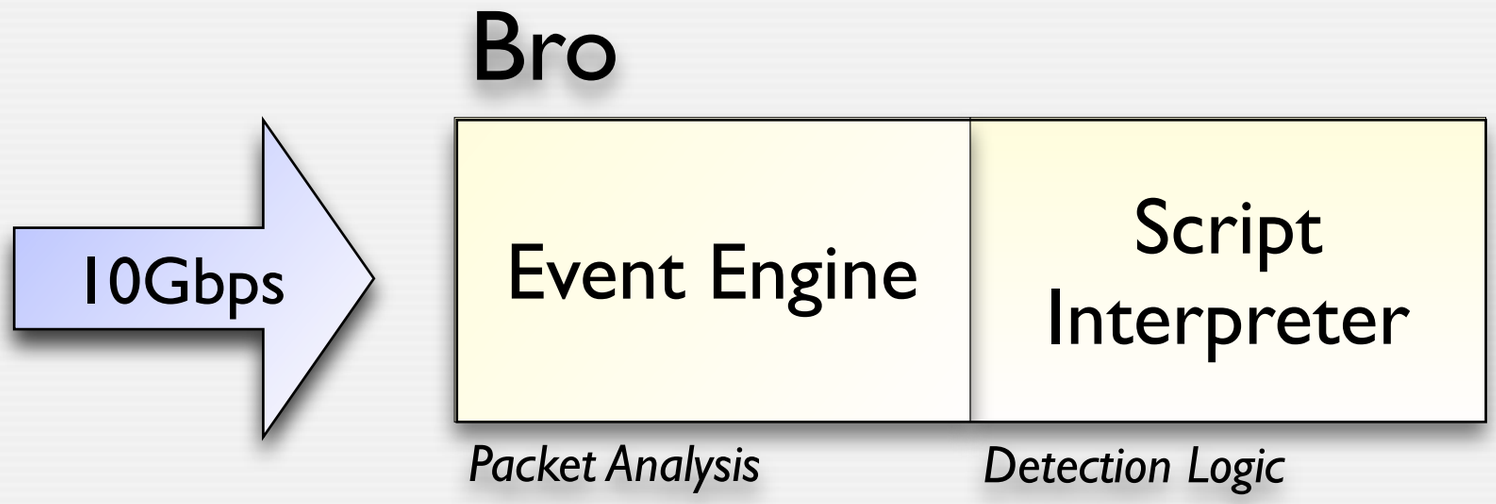
Communication Architecture



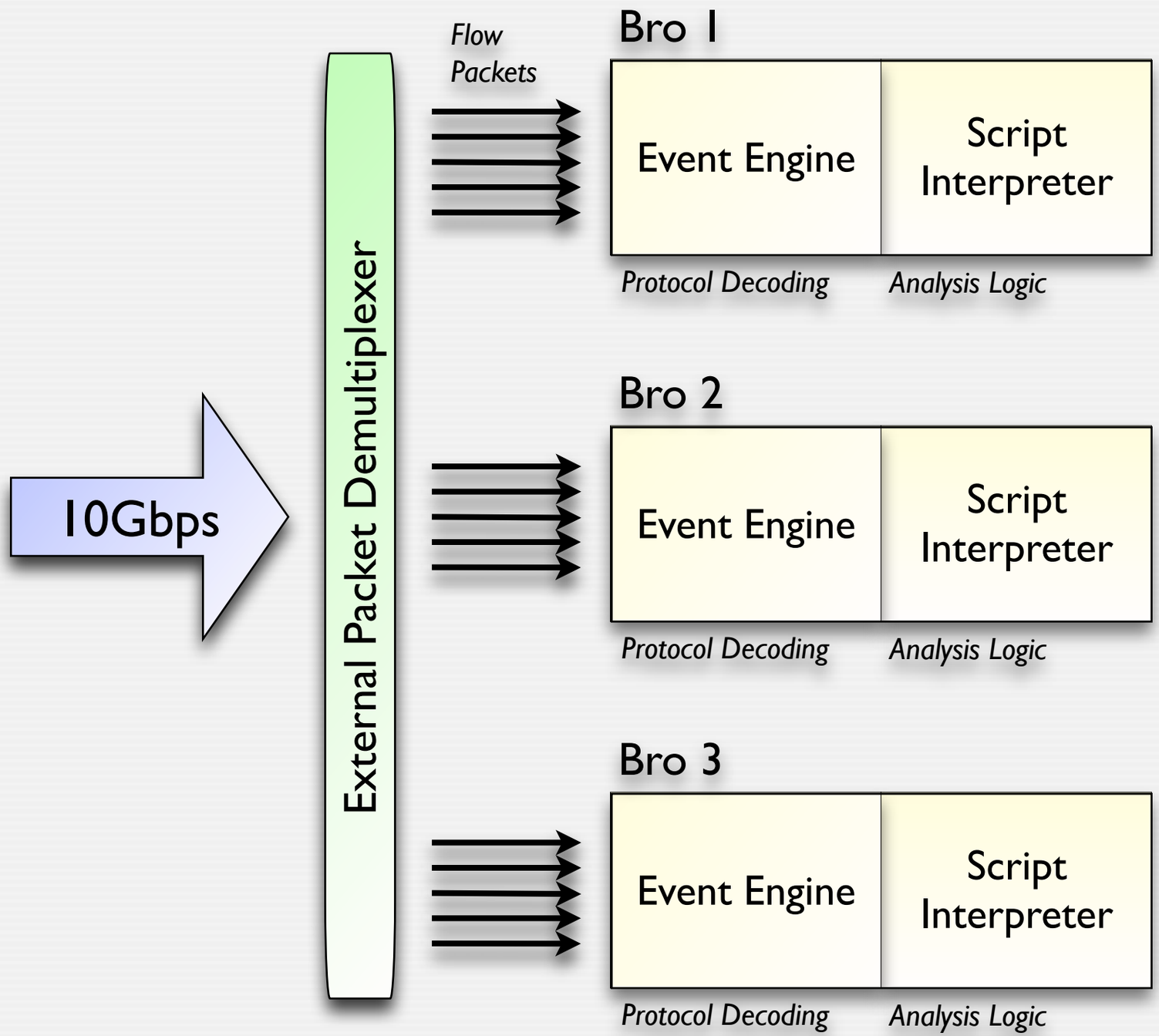
Communication Architecture



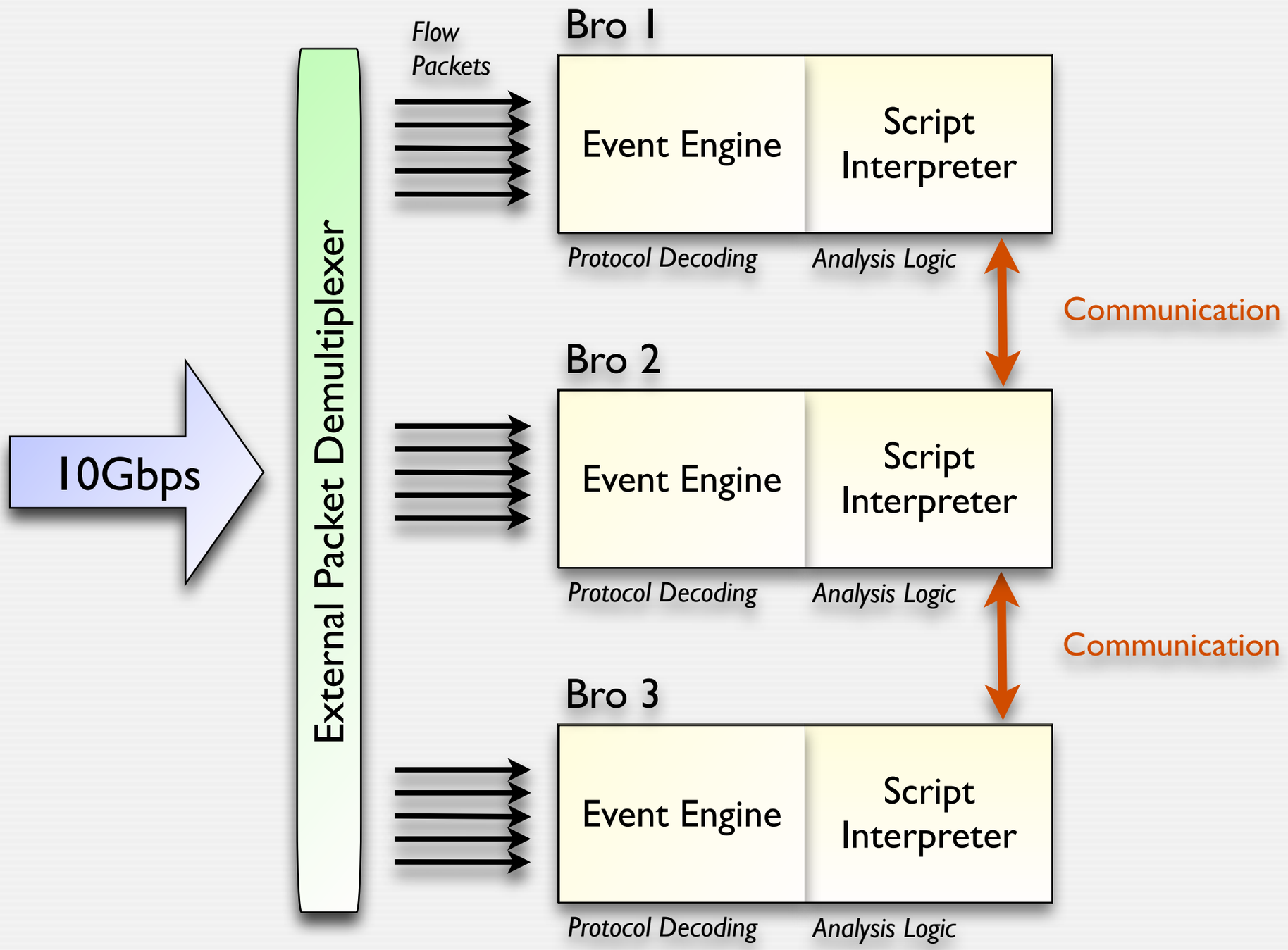
Load-Balancer Approach



Load-Balancer Approach

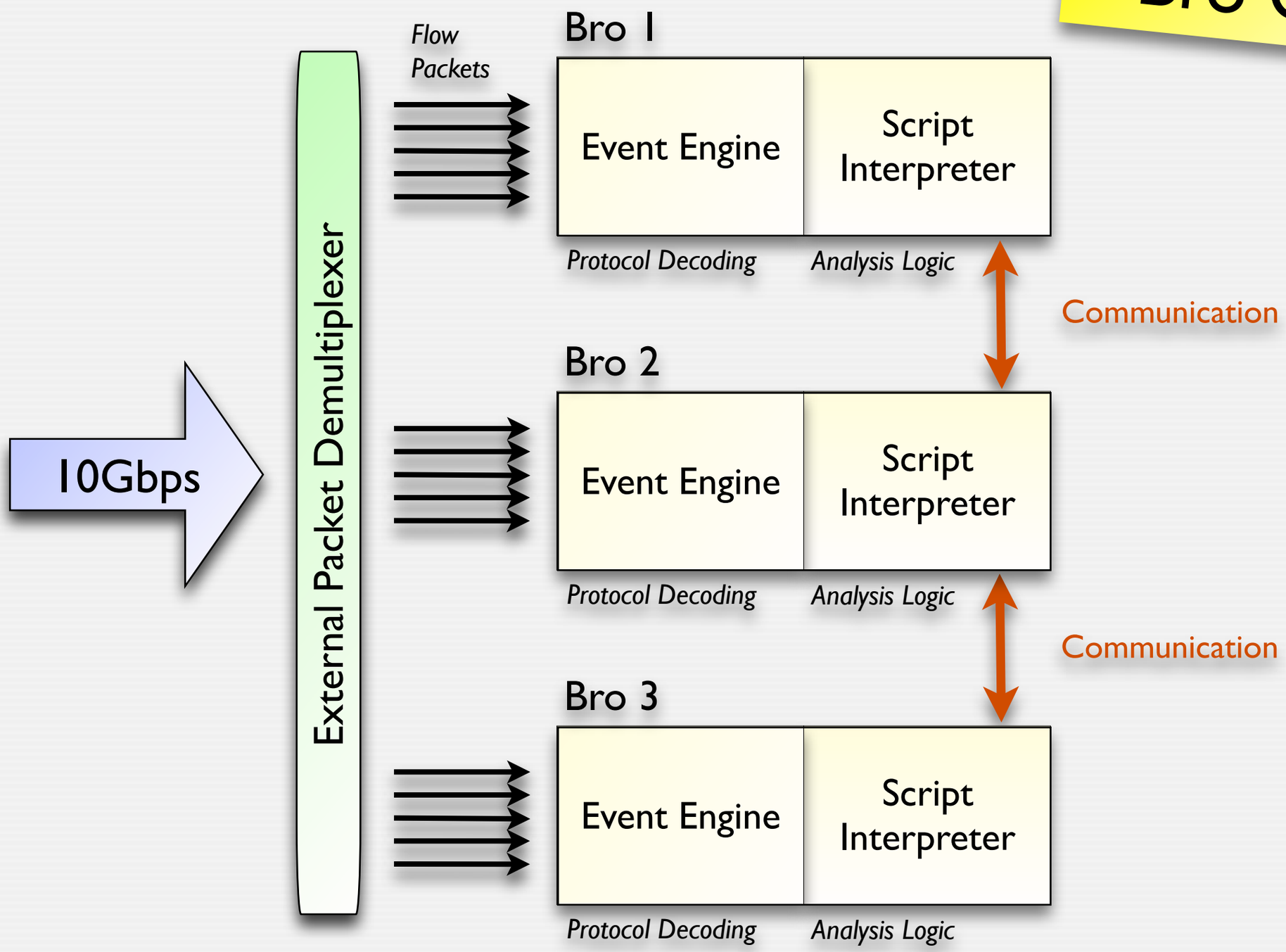


Load-Balancer Approach



Load-Balancer Approach

"Bro Cluster"



Cluster Challenges

Cluster Challenges

Communication capability required

- Bro has communication primitives built-in

Cluster Challenges

Communication capability required

- Bro has communication primitives built-in

Management of multi-machine setup is tedious

- Management interface transparently hides complexity (*BroControl*)

Cluster Challenges

Communication capability required

Bro has communication primitives built-in

Management of multi-machine setup is tedious

Management interface transparently hides complexity (*BroControl*)

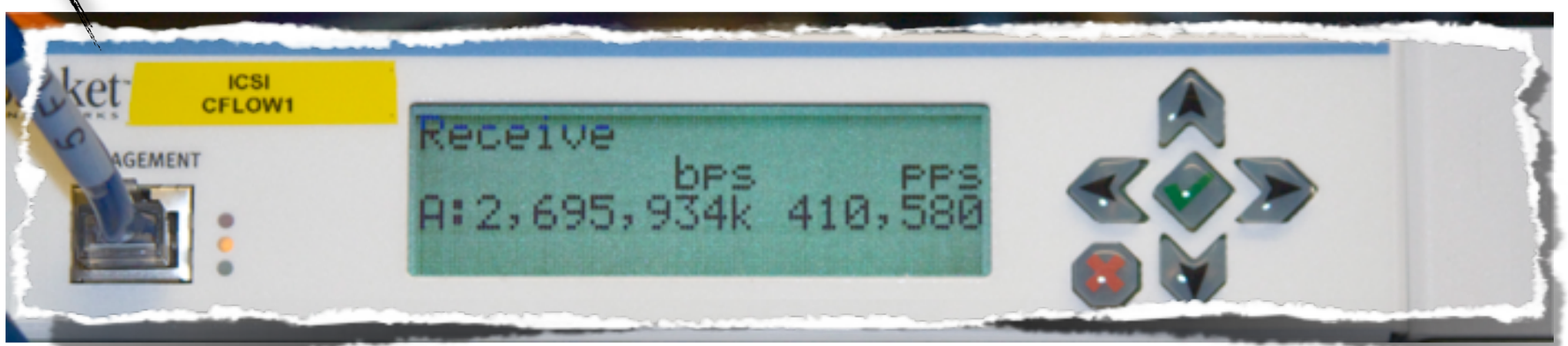
Demultiplexer needs to operate at line-rate

A number of solutions, including a commercially available appliance

UC Berkeley Research Cluster

UC Berkeley Research Cluster

UC Berkeley Research Cluster



Roadmap

Research Heritage

Research Heritage

Bro development has focused on research

- We were lacking resources for documentation, polishing, functionality

Research Heritage

Bro development has focused on research

We were lacking resources for documentation, polishing, functionality

NSF is now funding Bro *development*

Full-time engineers working on user experience and performance

Office of Cyberinfrastructure

Target: Blue Waters @ NCSA

Target: Blue Waters @ NCSA

10 PF/s peak performance

>1 PF/s sustained on applications
>300,000 cores
>1 Petabyte memory
>10 Petabyte disk storage
>0.5 Exabyte archival storage

Hosted in 88,000-square-foot facility

Next Release

Next Release

Bro 1.6

- New build, installation, and test framework.
- New quick-start guide.
- Overhauled default policy scripts.
- Comprehensive script documentation.
- New interface for extending scripts
- New logging framework.
- Code cleanup.
- Lots of little things fixed.
- Git repositories.
- New web server.

Roadmap

Roadmap

Medium-term

- Comprehensive user and reference manuals.
- High-performance binary logging.
- Integration of external "intelligence sources".
- Improved IPv6 support.
- New protocols.
- Database interface.
- Community script repository.
- More code cleanup.
- More things fixed.

Roadmap

Medium-term

- Comprehensive user and reference manuals.
- High-performance binary logging.
- Integration of external "intelligence sources".
- Improved IPv6 support.
- New protocols.
- Database interface.
- Community script repository.
- More code cleanup.
- More things fixed.

Long-Term

- Script compilation.
- Multi-threading.
- GUI.

Community

Community

We aim to involve the community more

Feedback and contributions welcome!

Community

We aim to involve the community more

Feedback and contributions welcome!

We can help you getting started

Get in touch and ask us!

Community

We aim to involve the community more

Feedback and contributions welcome!

We can help you getting started

Get in touch and ask us!

Let us know if you're using Bro

Operationally, experimentally, research?

Summary

Summary

Philosophy and Architecture

"A Python for Network Traffic Analysis"

Summary

Philosophy and Architecture

"A Python for Network Traffic Analysis"

Usage and Deployment

Activity logs, scripting examples, Bro Cluster

Summary

Philosophy and Architecture

"A Python for Network Traffic Analysis"

Usage and Deployment

Activity logs, scripting examples, Bro Cluster

Roadmap

Milestones and community

Get in Touch

Homepage	**www.bro-ids.org**
EMail	**info@bro-ids.org** **mailman.icsi.berkeley.edu/mailman/listinfo/bro**
Development	**git.bro-ids.org** **mailman.icsi.berkeley.edu/mailman/listinfo/bro-dev**

Get in Touch

Homepage	**www.bro-ids.org**
EMail	**info@bro-ids.org** **mailman.icsi.berkeley.edu/mailman/listinfo/bro**
Development	**git.bro-ids.org** **mailman.icsi.berkeley.edu/mailman/listinfo/bro-dev**

Thanks for your attention!