Bro 2.0 and Beyond

Network Attack Detection and Defense Early Warning Systems
Schloss Dagstuhl, 2012


ICSI
INTERNATIONAL
COMPUTER SCIENCE
INSTITUTE

Outline

Bro Introduction

"Much different from the typical IDS you may know"

Hot off the Press: Bro 2.0

Focus on operational deployment

Current Research Projects

Real-time Intelligence

Performance for next-gen environments

What is Bro?

What is Bro?

Packet Capture

What is Bro?

Packet Capture

Traffic Inspection

What is Bro?

Packet Capture

Traffic Inspection

Attack Detection

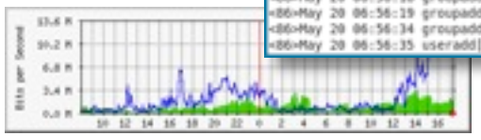
What is Bro?

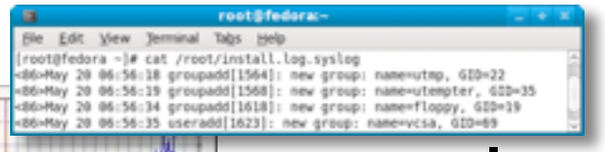
Packet Capture

Traffic Inspection

Attack Detection

NetFlow





syslog

Log Recording

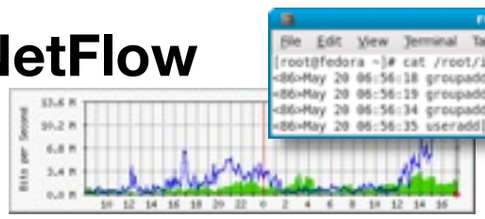
What is Bro?

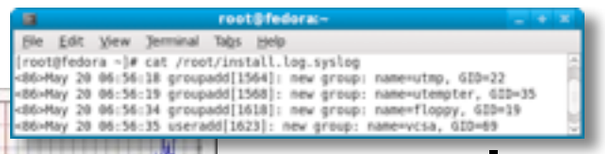
Packet Capture

Traffic Inspection

Attack Detection

NetFlow





syslog

Log Recording

Flexibility
Abstraction
Data Structures

What is Bro?

Packet Capture

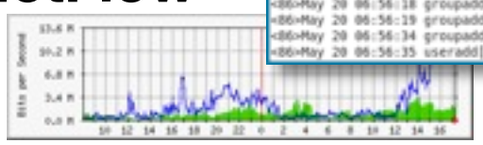
Traffic Inspection

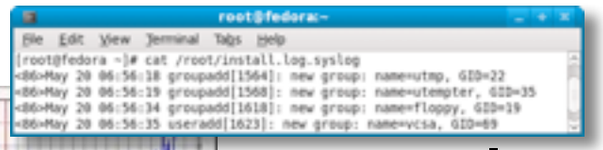
Attack Detection

Log Recording

Flexibility
Abstraction
Data Structures

NetFlow





syslog

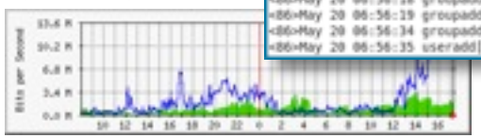
What is Bro?

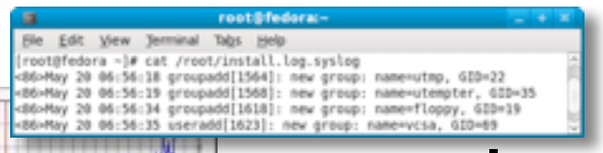
Packet Capture

Traffic Inspection

Attack Detection

NetFlow





syslog

Log Recording

Flexibility
Abstraction
Data Structures

What is Bro?

Packet Capture

Traffic Inspection

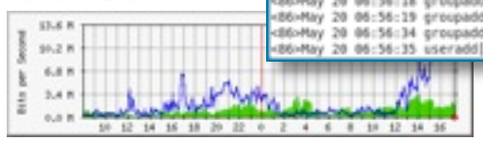
Attack Detection

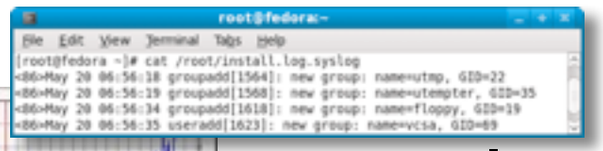
Log Recording

Flexibility
Abstraction
Data Structures

"Domain-specific Python"

NetFlow





syslog

NCSA

Philosophy

Fundamentally different from other IDS.

Reset your idea of an IDS before starting to use Bro.

Real-time network analysis *framework.*

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

Can accommodate a range of detection approaches.

Policy-neutral at the core.

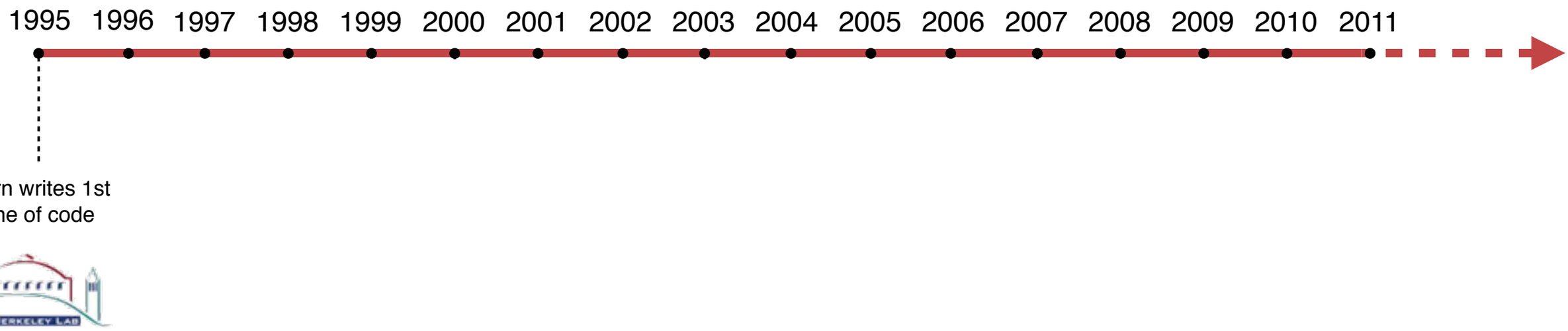
Highly stateful.

Tracks extensive application-layer network state.

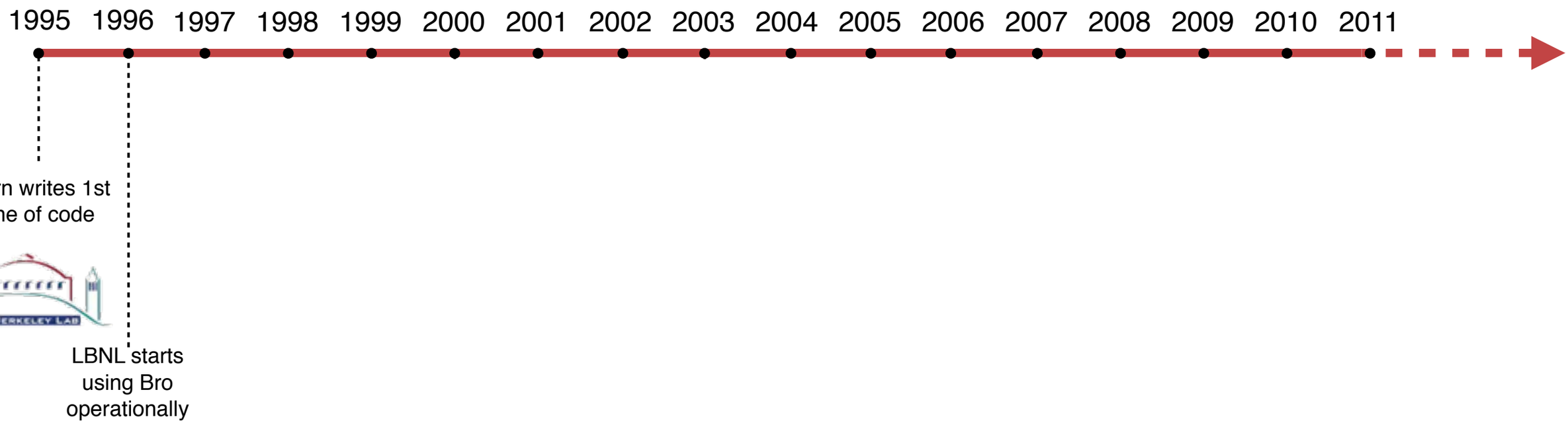
Supports forensics.

Extensively logs what it sees.

Bro History


1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
Vern writes 1st line of code

Bro History


1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
Vern writes 1st line of code
LBNL starts using Bro operationally

Bro History

1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011

Vern writes 1st line of code

LBNL starts using Bro operationally

v0.2
1st CHANGES entry

v0.4
HTTP analysis
Scan detector
IP fragments
Linux support

v0.6
RegExps
Login analysis

v0.7a48
Consistent CHANGES

v0.7a90
Profiling
State Mgmt

v0.7a175/0.8aX
Signatures
SMTP
IPv6 support
User manual

0.8a37
Communication
Persistence
Namespaces
Log Rotation

v0.8aX/0.9aX
SSL/SMB
STABLE releases
BroLite

v1.0
BinPAC
IRC/RPC analyzers
64-bit support
Sane version numbers

v1.1/v1.2
when Stmt
Resource tuning
Broccoli
DPD

v1.3
Ctor expressions
GeoIP
Conn Compressor

v1.4
DHCP/BitTorrent
HTTP entities
NetFlow
Bro Lite Deprecated

v1.5
BroControl

Bro Waters

Bro 2.0

Bro History

Academic Publications

1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011

Vern writes 1st line of code

LBNL starts using Bro operationally

USENIX Paper

v0.2
1st CHANGES entry

v0.4
HTTP analysis
Scan detector
IP fragments
Linux support

Stepping Stone Detector

v0.6
RegExps
Login analysis

v0.7a48
Consistent CHANGES

v0.7a90
Profiling
State Mgmt

Anonymizer
Active Mapping
Context Signat.

v0.7a175/0.8aX
Signatures
SMTP
IPv6 support
User manual

TRW
State Mgmt.
Indepen. State

0.8a37
Communication
Persistence
Namespaces
Log Rotation

Host Context
Time Machine
Enterprise Traffic

v0.8aX/0.9aX
SSL/SMB
STABLE releases
BroLite

v1.0
BinPAC
IRC/RPC analyzers
64-bit support
Sane version numbers

BinPAC
DPD
2nd Path

v1.1/v1.2
when Stmt
Resource tuning
Broccoli
DPD

Bro Cluster
Shunt

v1.3
Ctor expressions
GeoIP
Conn Compressor

Autotuning

v1.4
DHCP/BitTorrent
HTTP entities
NetFlow
Bro Lite Deprecated

Parallel Prototype

v1.5
BroControl

Bro Waters

Bro 2.0

“Who’s Using It?”

Example Logs

Example Logs

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

Example Logs

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

#fields ts	*id.orig_h*	*id.orig_p*	*id.resp_h*	*id.resp_p*	*proto*	*service*	*duration*
1144876741.1198	*192.150.186.169*	*53115*	*82.94.237.218*	*80*	*tcp*	*http*	*16.14929*
1144876612.6063	192.150.186.169	53090	198.189.255.82	80	tcp	http	4.437460
1144876596.5597	192.150.186.169	53051	193.203.227.129	80	tcp	http	0.372440
1144876606.7789	192.150.186.169	53082	198.189.255.73	80	tcp	http	0.597711
1144876741.4693	**192.150.186.169**	**53116**	**82.94.237.218**	**80**	**tcp**	**http**	**16.02667**
1144876745.6102	**192.150.186.169**	**53117**	**66.102.7.99**	**80**	**tcp**	**http**	**1.004346**
1144876605.6847	192.150.186.169	53075	207.151.118.143	80	tcp	http	0.029663

Example Logs

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

#fields ts	*id.orig_h*	*id.orig_p*	*id.resp_h*	*id.resp_p*	*proto*	*service*	*duration*
1144876741.1198	*192.150.186.169*	*53115*	*82.94.237.218*	*80*	*tcp*	*http*	*16.14929*
1144876612.6063	192.150.186.169	53090	198.189.255.82	80	tcp	http	4.437460
1144876596.5597	192.150.186.169	53051	193.203.227.129	80	tcp	http	0.372440
1144876606.7789	192.150.186.169	53082	198.189.255.73	80	tcp	http	0.597711
1144876741.4693	**192.150.186.169**	**53116**	**82.94.237.218**	**80**	**tcp**	**http**	**16.02667**
1144876745.6102	**192.150.186.169**	**53117**	**66.102.7.99**	**80**	**tcp**	**http**	**1.004346**
1144876605.6847	192.150.186.169	53075	207.151.118.143	80	tcp	http	0.029663

```
> cat http.log
```

Example Logs

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

#fields ts	*id.orig_h*	*id.orig_p*	*id.resp_h*	*id.resp_p*	*proto*	*service*	*duration*
1144876741.1198	*192.150.186.169*	*53115*	*82.94.237.218*	*80*	*tcp*	*http*	*16.14929*
1144876612.6063	192.150.186.169	53090	198.189.255.82	80	tcp	http	4.437460
1144876596.5597	192.150.186.169	53051	193.203.227.129	80	tcp	http	0.372440
1144876606.7789	192.150.186.169	53082	198.189.255.73	80	tcp	http	0.597711
1144876741.4693	**192.150.186.169**	**53116**	**82.94.237.218**	**80**	**tcp**	**http**	**16.02667**
1144876745.6102	**192.150.186.169**	**53117**	**66.102.7.99**	**80**	**tcp**	**http**	**1.004346**
1144876605.6847	192.150.186.169	53075	207.151.118.143	80	tcp	http	0.029663

```
> cat http.log
```

#fields ts	*id.orig_h*	*id.orig_p [...]*	*host*	*uri*	*status_code*	*user_agent [...]*
1144876741.6335	192.150.186.169	53116	docs.python.org	/lib/lib.css	200	Mozilla/5.0
1144876742.1687	192.150.186.169	53116	docs.python.org	/icons/previous.png	304	Mozilla/5.0
1144876741.2838	192.150.186.169	53115	docs.python.org	/lib/lib.html	200	Mozilla/5.0
1144876742.3337	192.150.186.169	53116	docs.python.org	/icons/up.png	304	Mozilla/5.0
1144876742.3337	192.150.186.169	53116	docs.python.org	/icons/next.png	304	Mozilla/5.0
1144876742.3337	192.150.186.169	53116	docs.python.org	/icons/contents.png	304	Mozilla/5.0
1144876742.3337	192.150.186.169	53116	docs.python.org	/icons/modules.png	304	Mozilla/5.0
1144876742.3338	192.150.186.169	53116	docs.python.org	/icons/index.png	304	Mozilla/5.0
1144876745.6144	192.150.186.169	53117	www.google.com	/	200	Mozilla/5.0

Example Logs

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

```
#fields ts          id.orig_h          id.orig_p  id.resp_h          id.resp_p proto  service  duration
1144876741.1198  192.150.186.169 53115      82.94.237.218   80        tcp    http     16.14929
1144876612.6063  192.150.186.169 53090      198.189.255.82  80        tcp    http     4.437460
```

```
[...] host                uri                      status_code  user_agent [...]
      docs.python.org     /lib/lib.css             200          Mozilla/5.0
      docs.python.org     /icons/previous.png      304          Mozilla/5.0
      docs.python.org     /lib/lib.html            200          Mozilla/5.0
      docs.python.org     /icons/up.png            304          Mozilla/5.0
      docs.python.org     /icons/next.png          304          Mozilla/5.0
      docs.python.org     /icons/contents.png      304          Mozilla/5.0
      docs.python.org     /icons/modules.png       304          Mozilla/5.0
      docs.python.org     /icons/index.png         304          Mozilla/5.0
       www.google.com     /                        200          Mozilla/5.0
```

```
1144876742.3338  192.150.186.169  53116        docs.python.org  /icons/index.png     304          Mozilla/5.0
1144876745.6144  192.150.186.169  53117         www.google.com  /                    200          Mozilla/5.0
```

Script Example: Matching URLs

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

Script Example: Matching URLs

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

```
event http_request(c: connection,           # Connection.
                   method: string,          # HTTP method.
                   original_URI: string,    # Requested URL.
                   unescaped_URI: string,   # Decoded URL.
                   version: string)         # HTTP version.
{
    if ( method == "GET" && unescaped_URI == /.*passwd/ )
        NOTICE(...); # Alarm.
}
```

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 count &default=0;

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

Distributed Scripts

Distributed Scripts

Bro comes with >10,000 lines of script code.

Prewritten functionality that's just loaded.

Scripts also generate the logs.

Amendable to extensive customization and extension.

Version 2.0

Version 2.0

Default scripts rewritten from scratch.

Focus ease of use and operational deployment.

New logging infrastructure.

New build and packaging system.

New auto-documentation system (Broxygen).

Lots of bugs fixed.

Obsolete code removed.

New development infrastructure.

New regression testing framework.

New web server.

New mailing lists.

New logo.

Upcoming

Upcoming

Bro 2.1

Overhauled IPv6 support.

New user's guide.

Logging extensions.

- Binary logging/Postgresql/CouchDB/SQLite(?) / Threads.

Input framework.

Reaction framework.

New/improved analyzers.

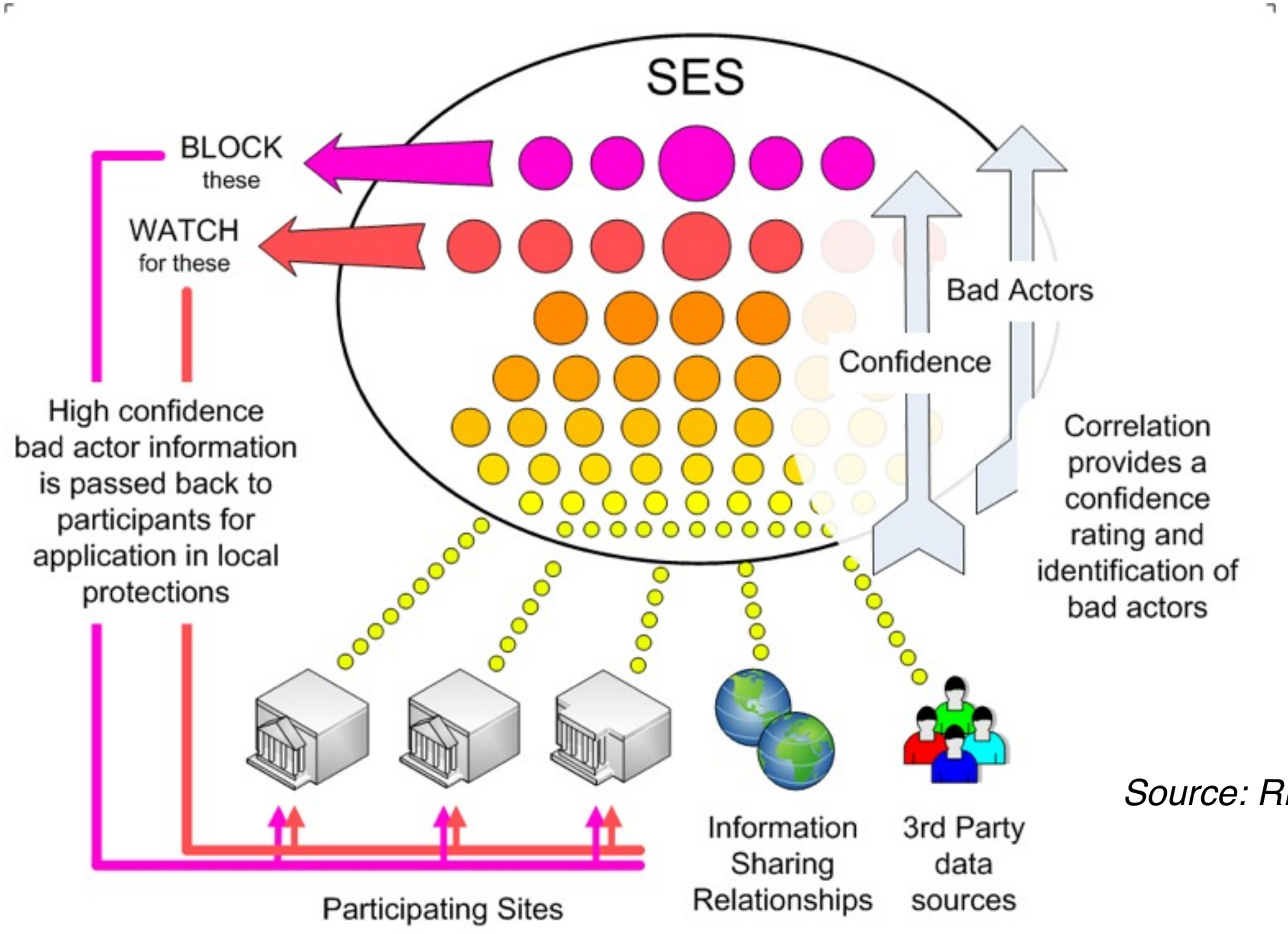
- Syslog/GridFTP/NFS/SMB/BitTorrent.

Extended test-suite.

Aiming for 3-4 months release cycle.

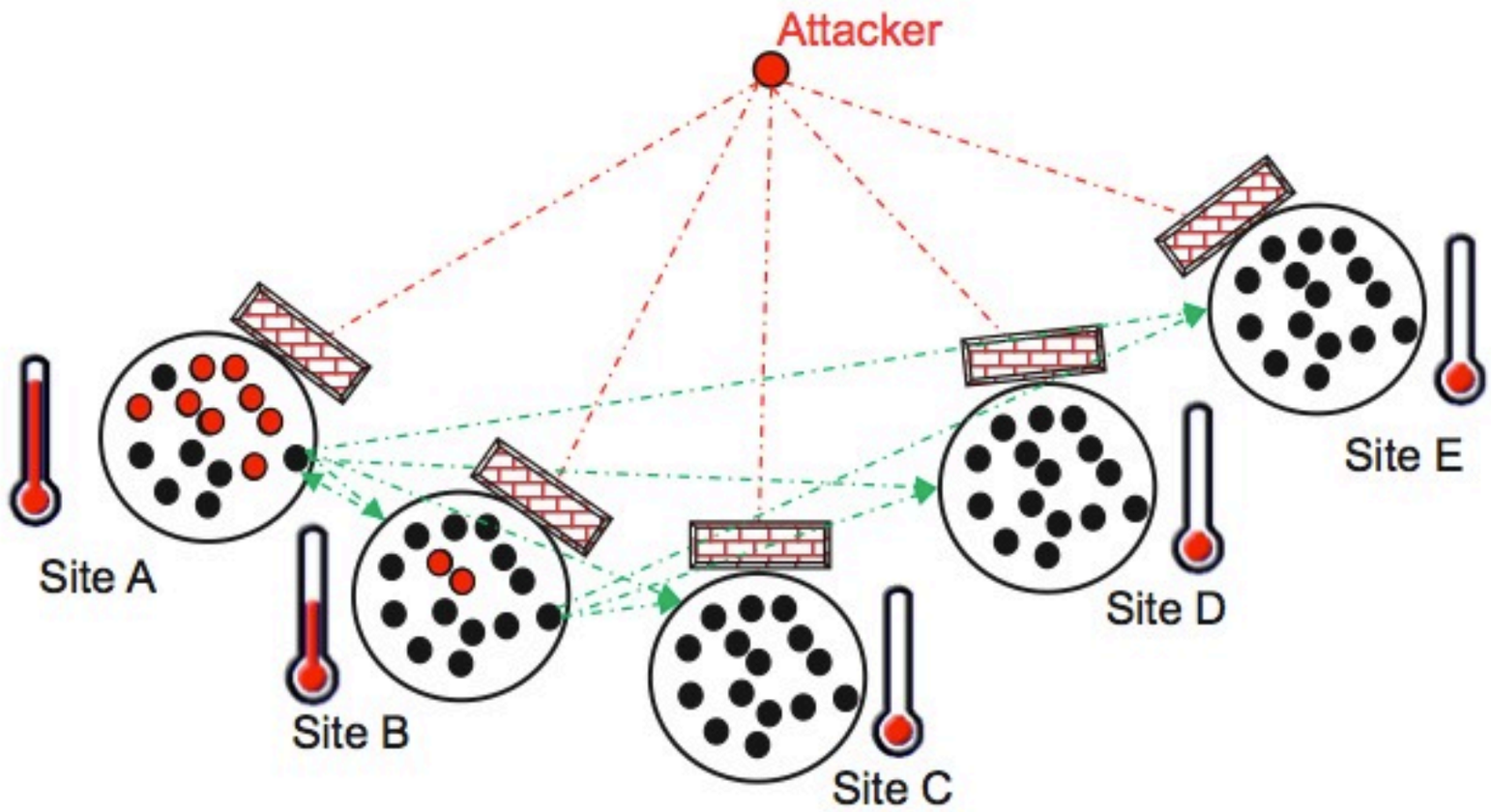
Current Research
Real-Time Intelligence

REN-ISAC's Security Event System



Source: REN-ISAC

Argonne Federated Model



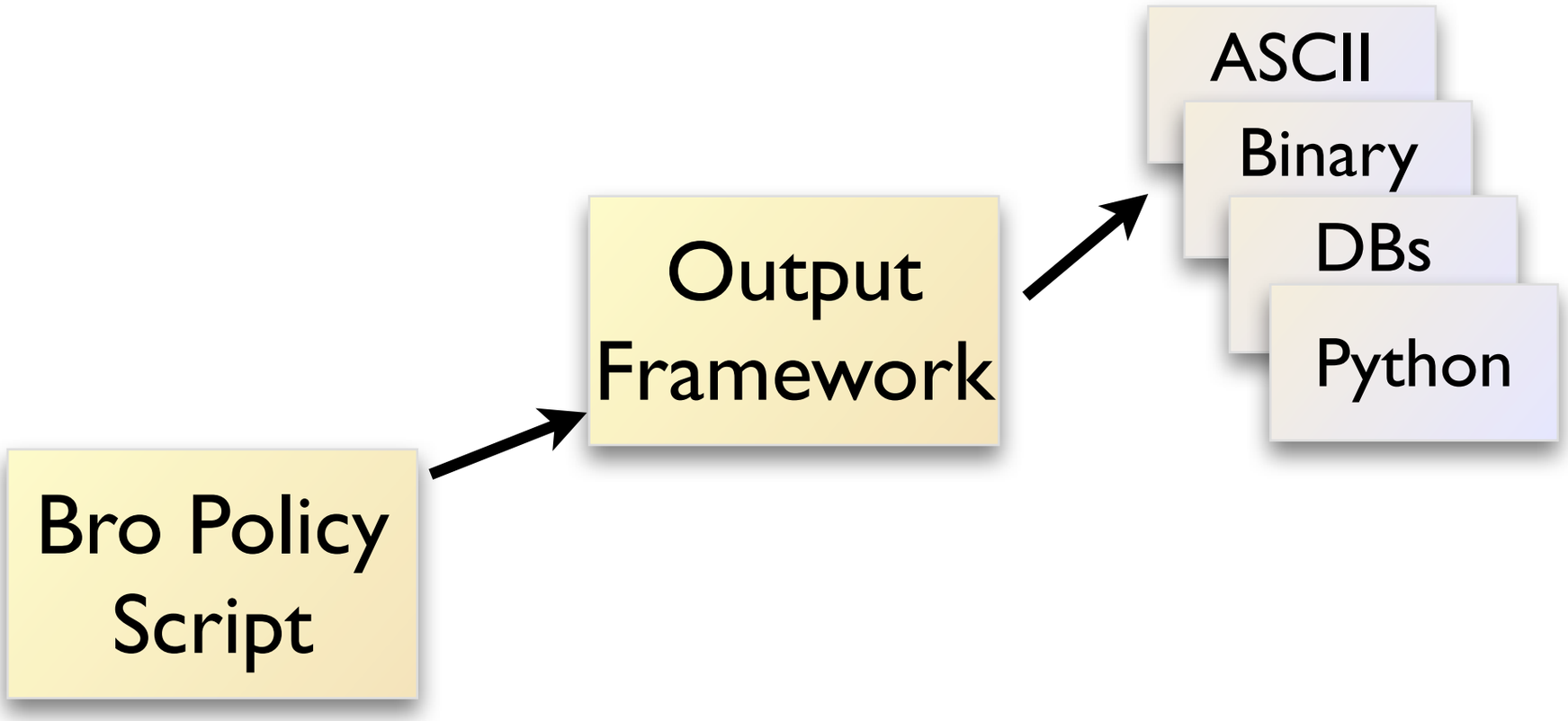
Cyber Fed Model

Collaborative Effort to Combat Internet Attackers

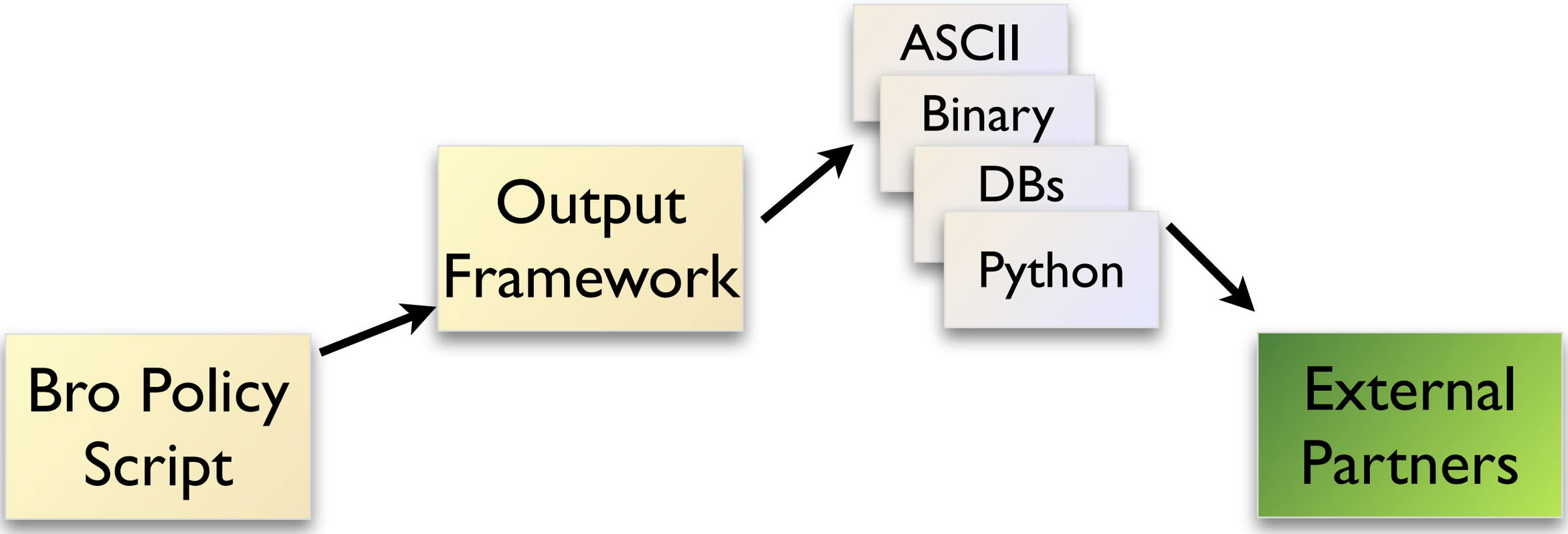
Source: Argonne National Lab

Real-time Intelligence with Bro

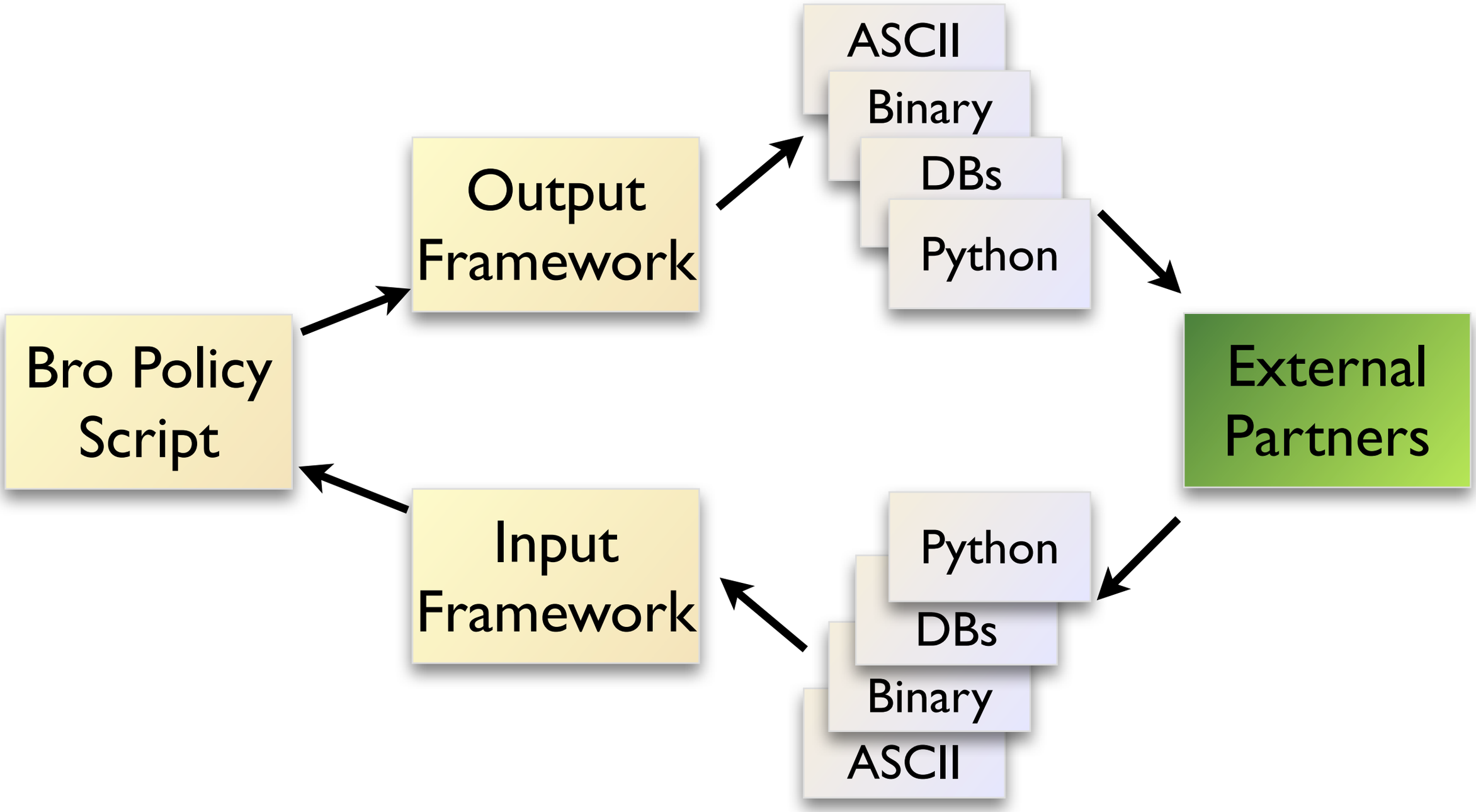
Real-time Intelligence with Bro



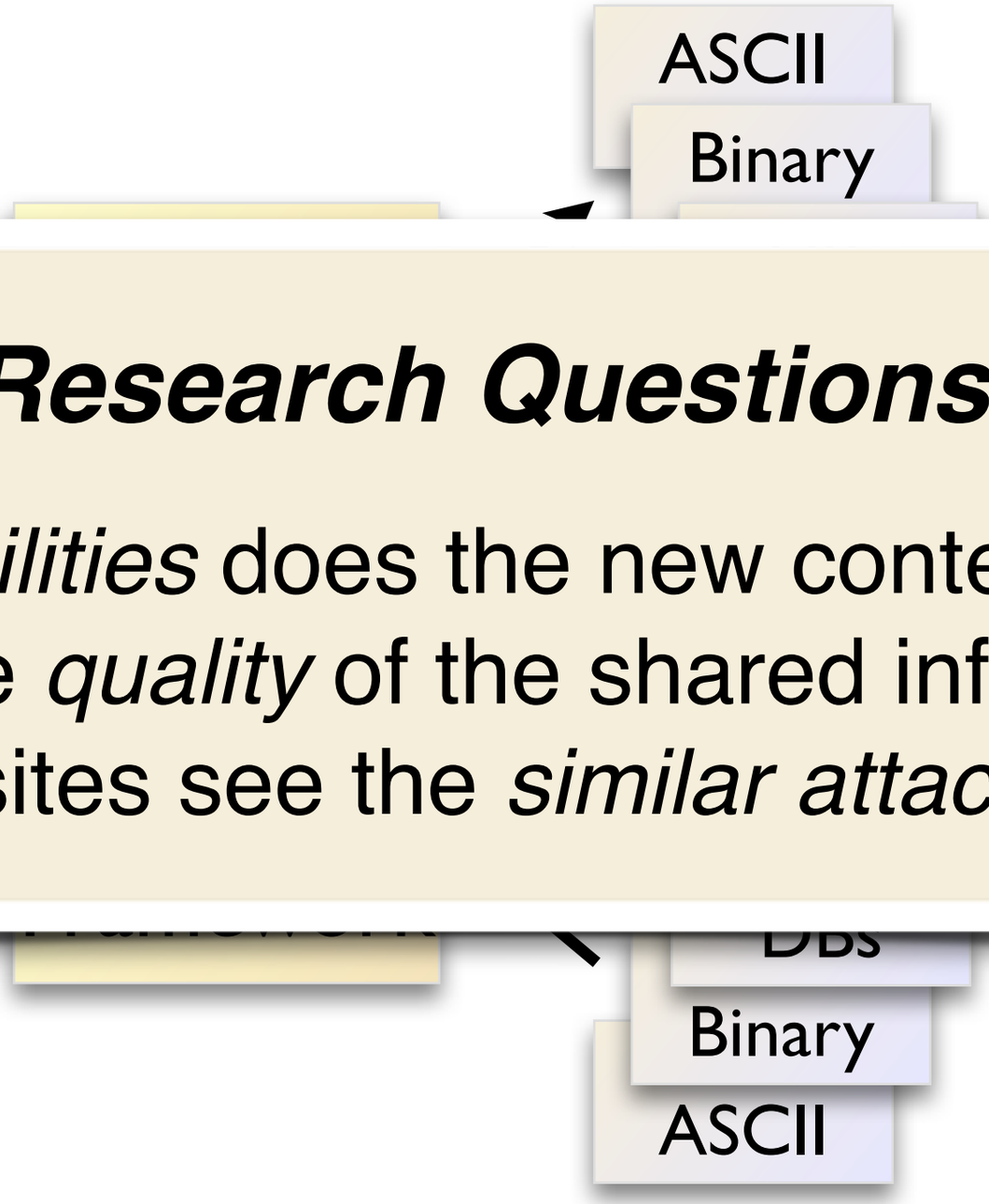
Real-time Intelligence with Bro



Real-time Intelligence with Bro



Real-time Intelligence with Bro



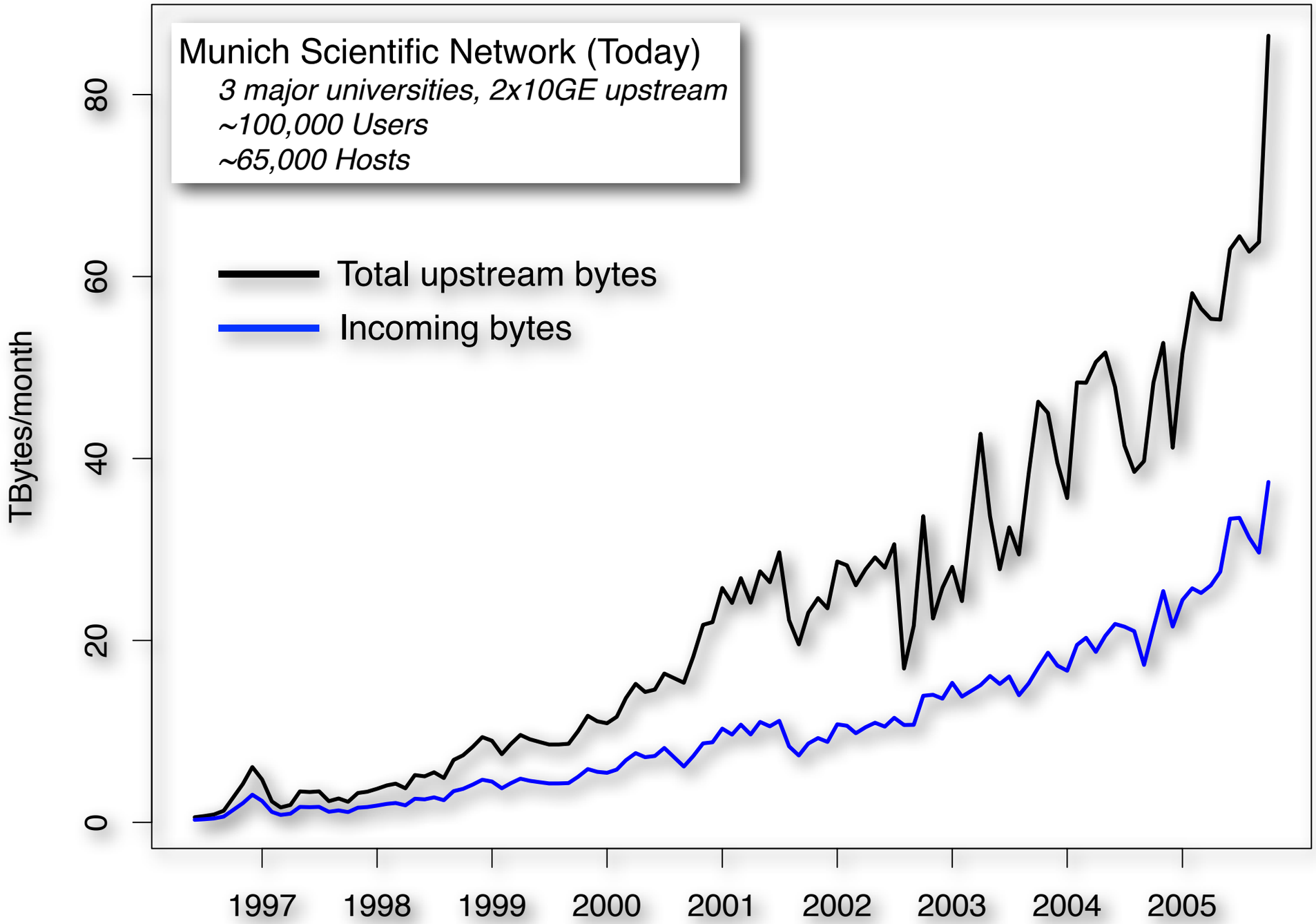
Research Questions

What *capabilities* does the new context give us?
What is the *quality* of the shared information?
Do sites see the *similar attacks*?

Current Research
Performance

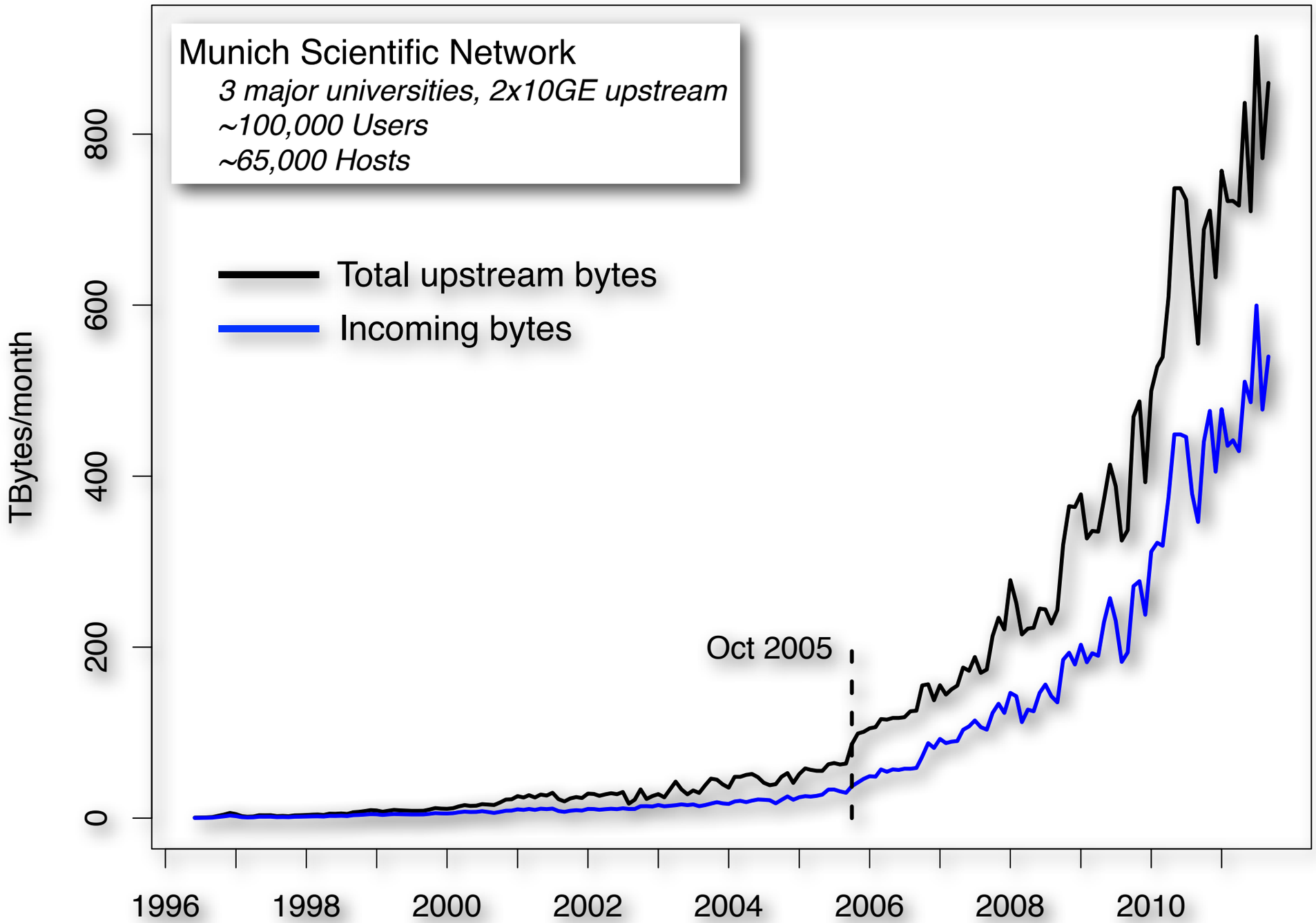
Back in 2005 ...

Back in 2005 ...



Data: Leibniz-Rechenzentrum, München

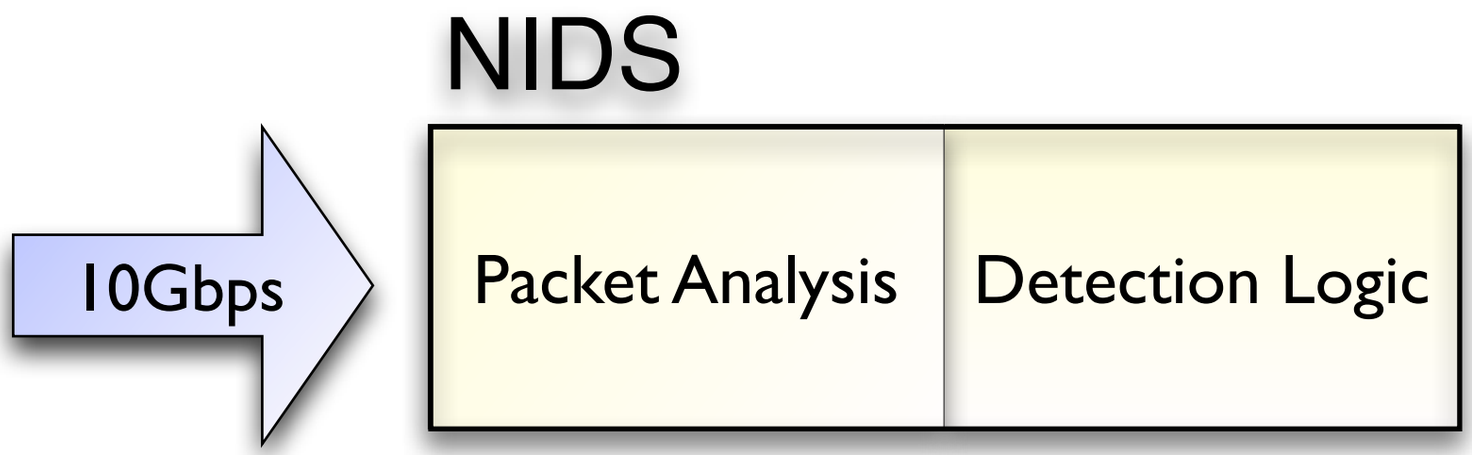
Today ...



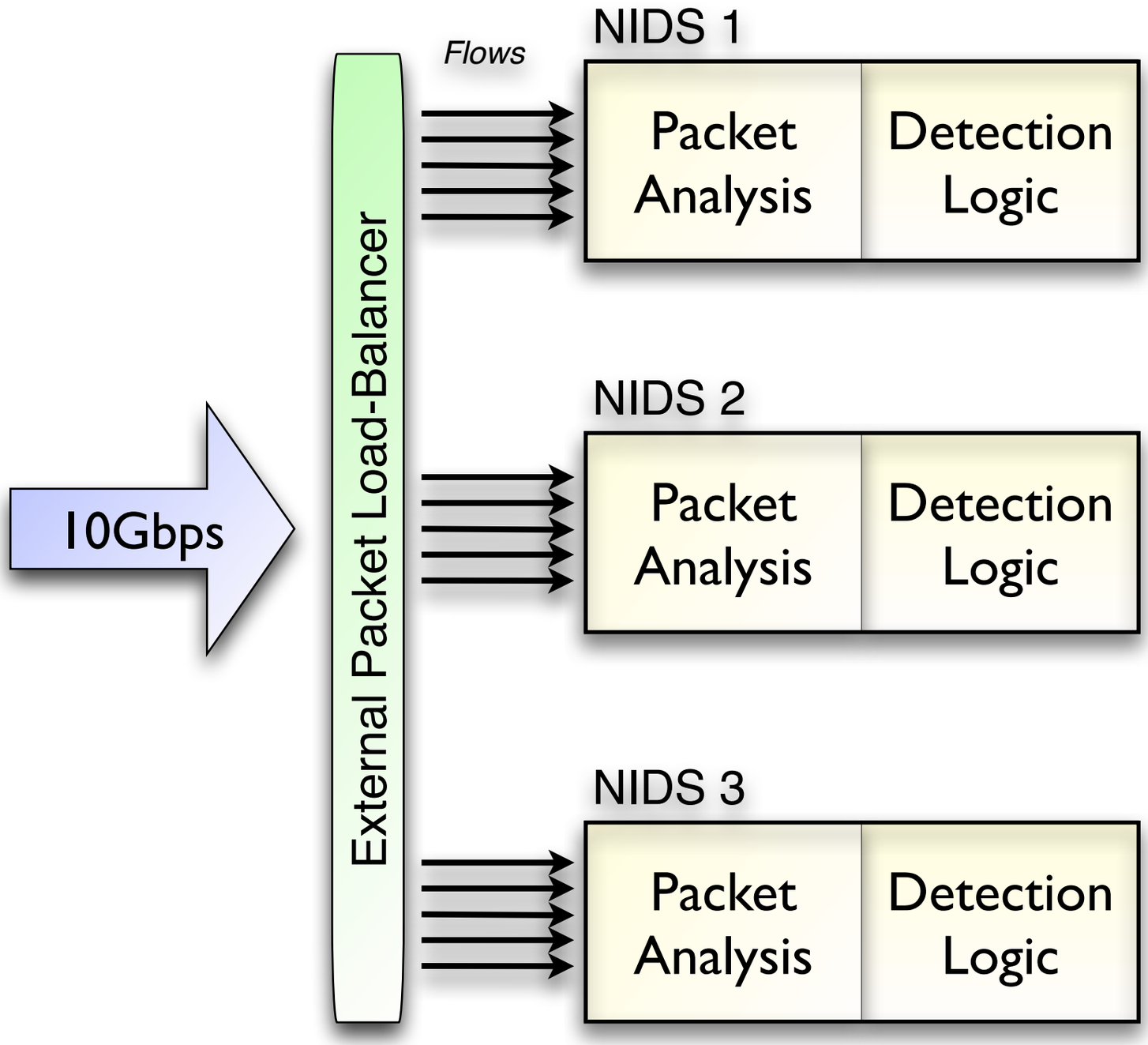
Data: Leibniz-Rechenzentrum, München

Load-balancing Architecture

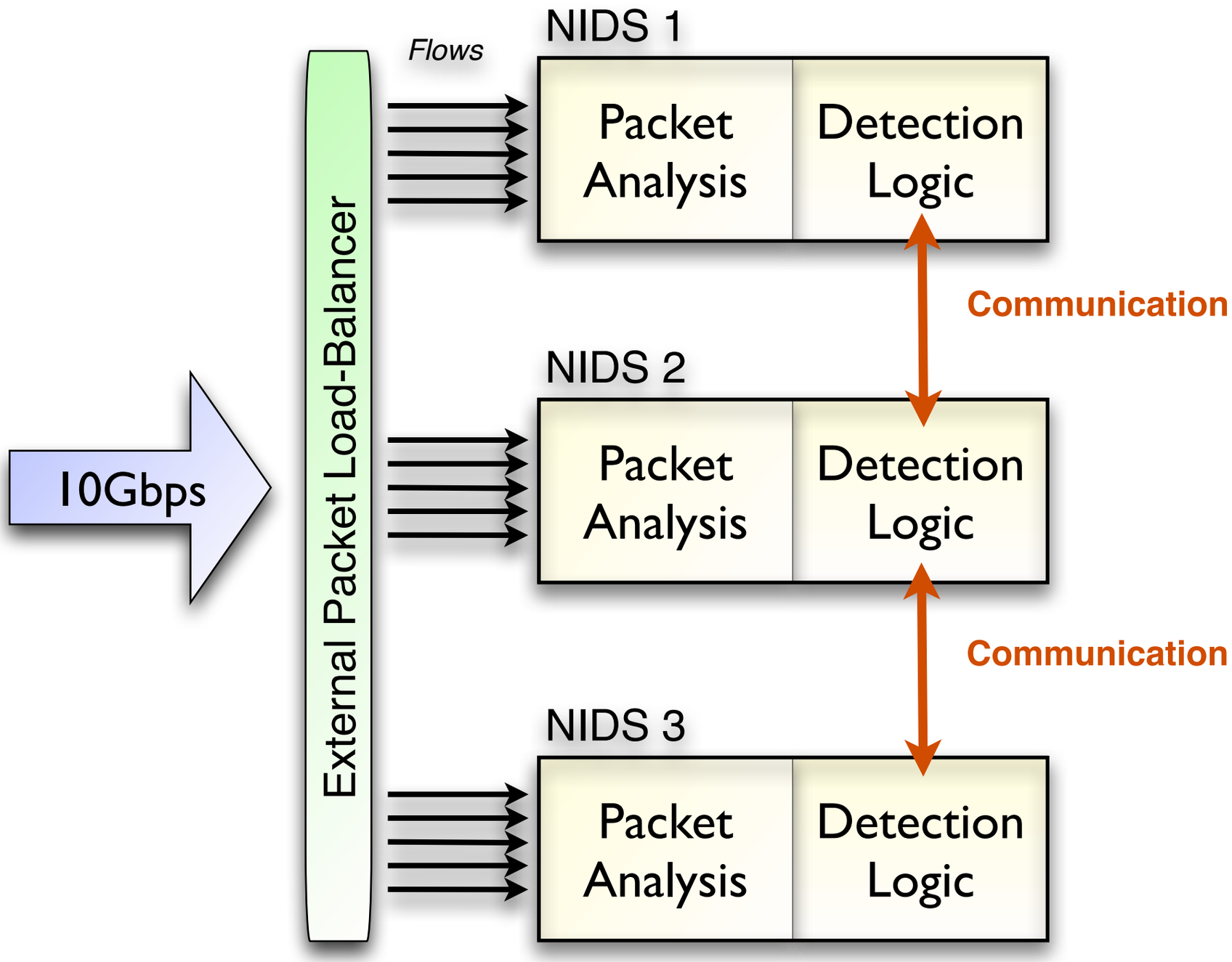
Load-balancing Architecture



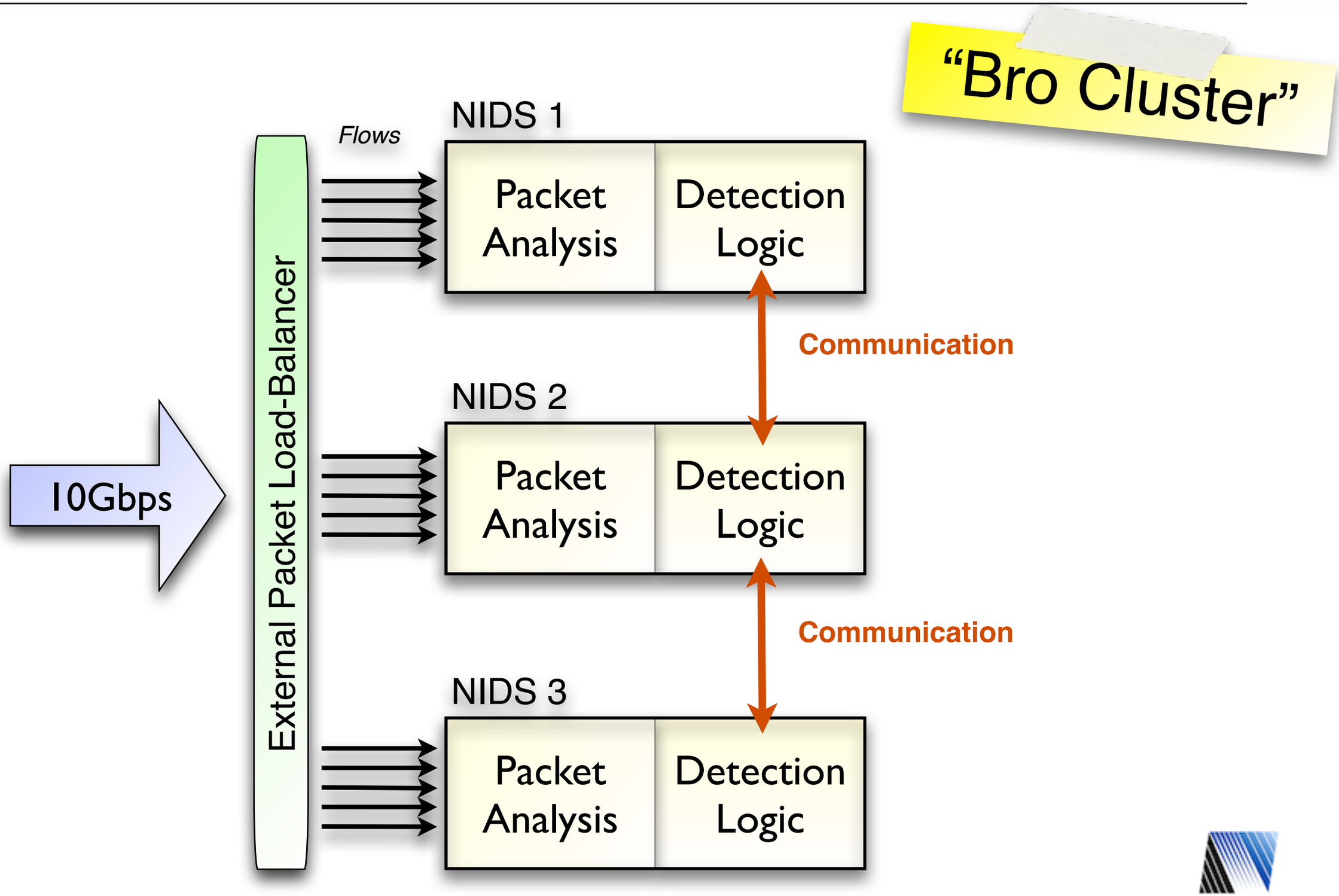
Load-balancing Architecture



Load-balancing Architecture

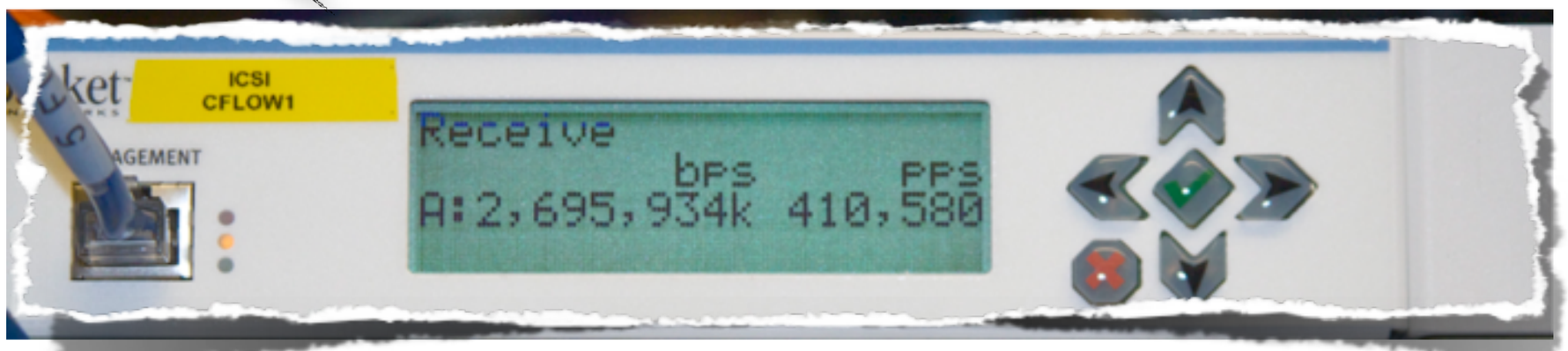


Load-balancing Architecture

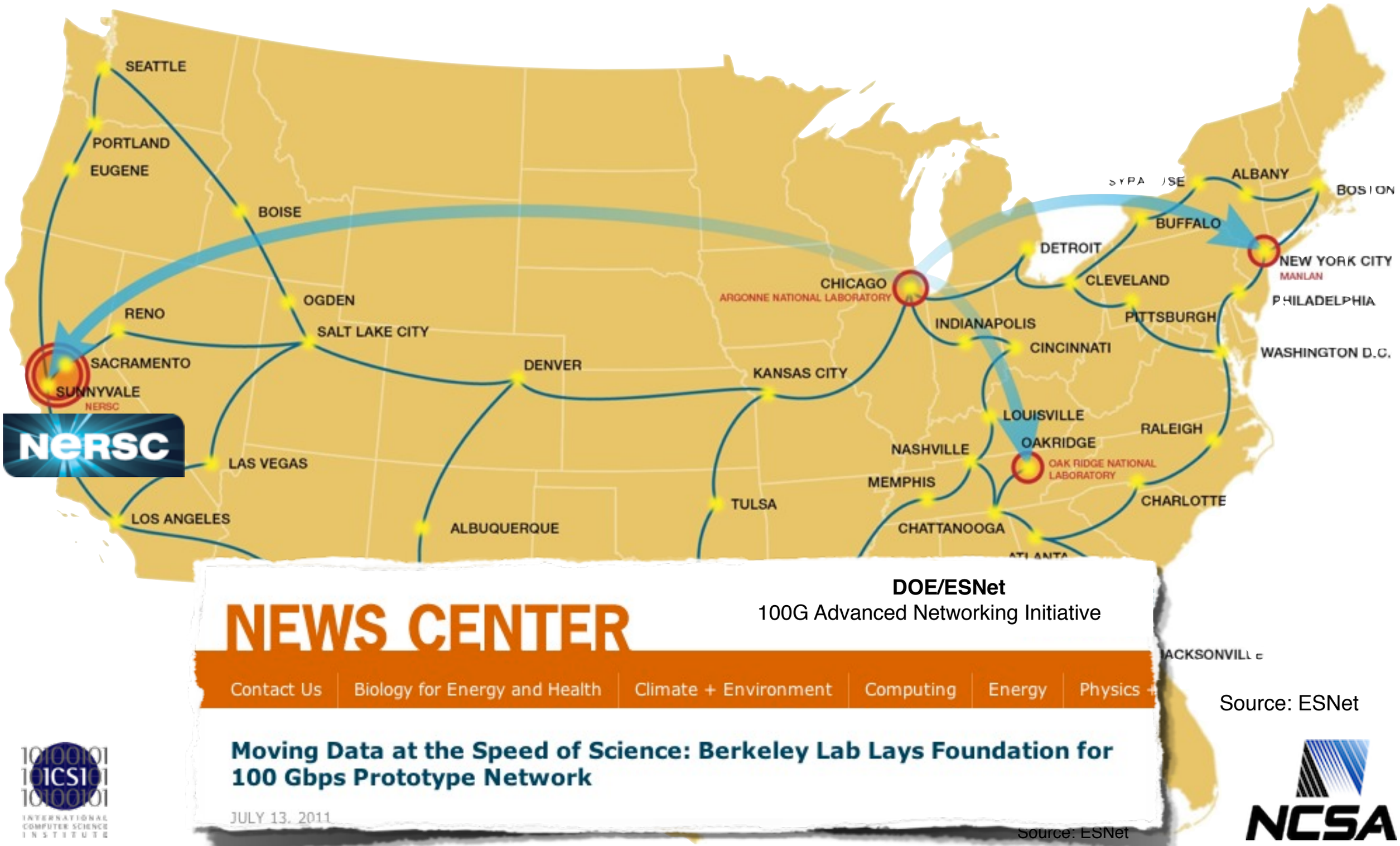


cPacket's cFlow 10G

cPacket's cFlow 10G



Next Stop: 100 Gb/s



DOE/ESNet
100G Advanced Networking Initiative

NEWS CENTER

Contact Us | Biology for Energy and Health | Climate + Environment | Computing | Energy | Physics +

Moving Data at the Speed of Science: Berkeley Lab Lays Foundation for 100 Gbps Prototype Network

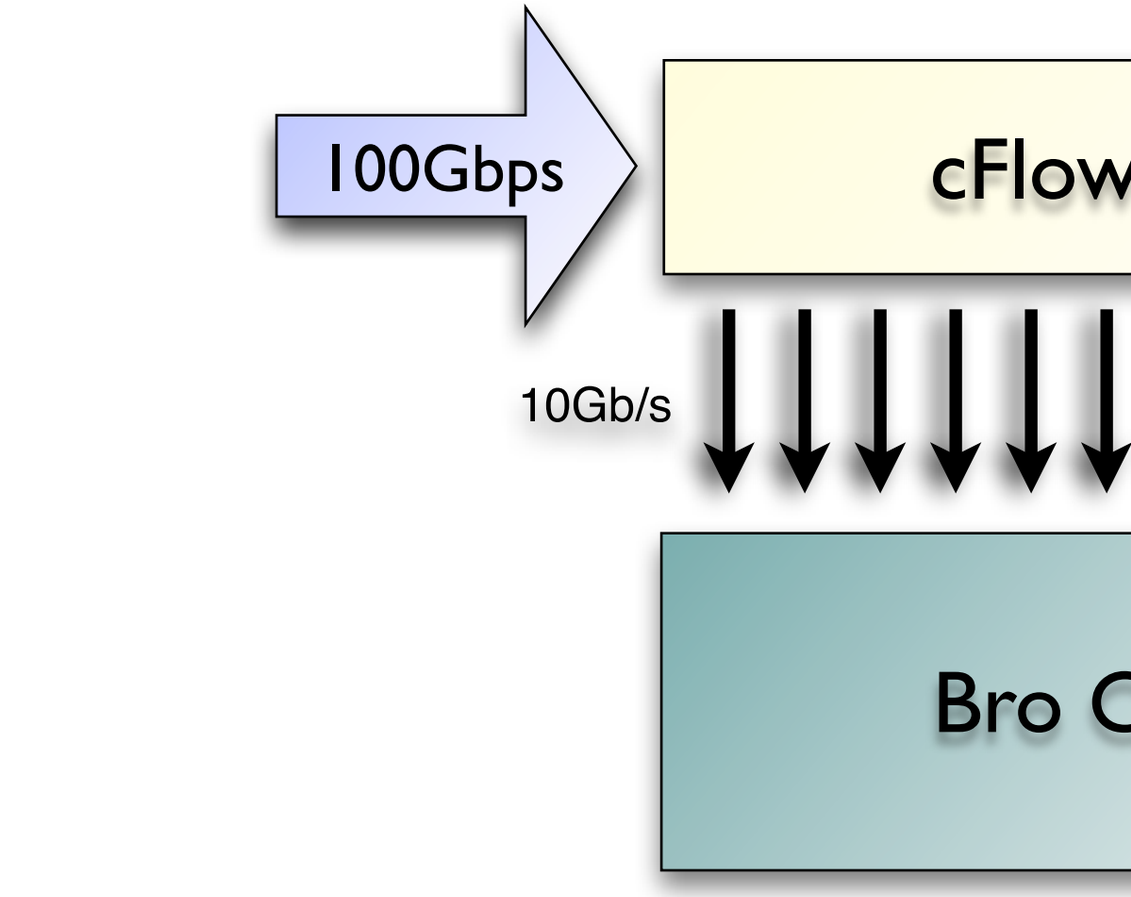
JULY 13, 2011

Source: ESNet

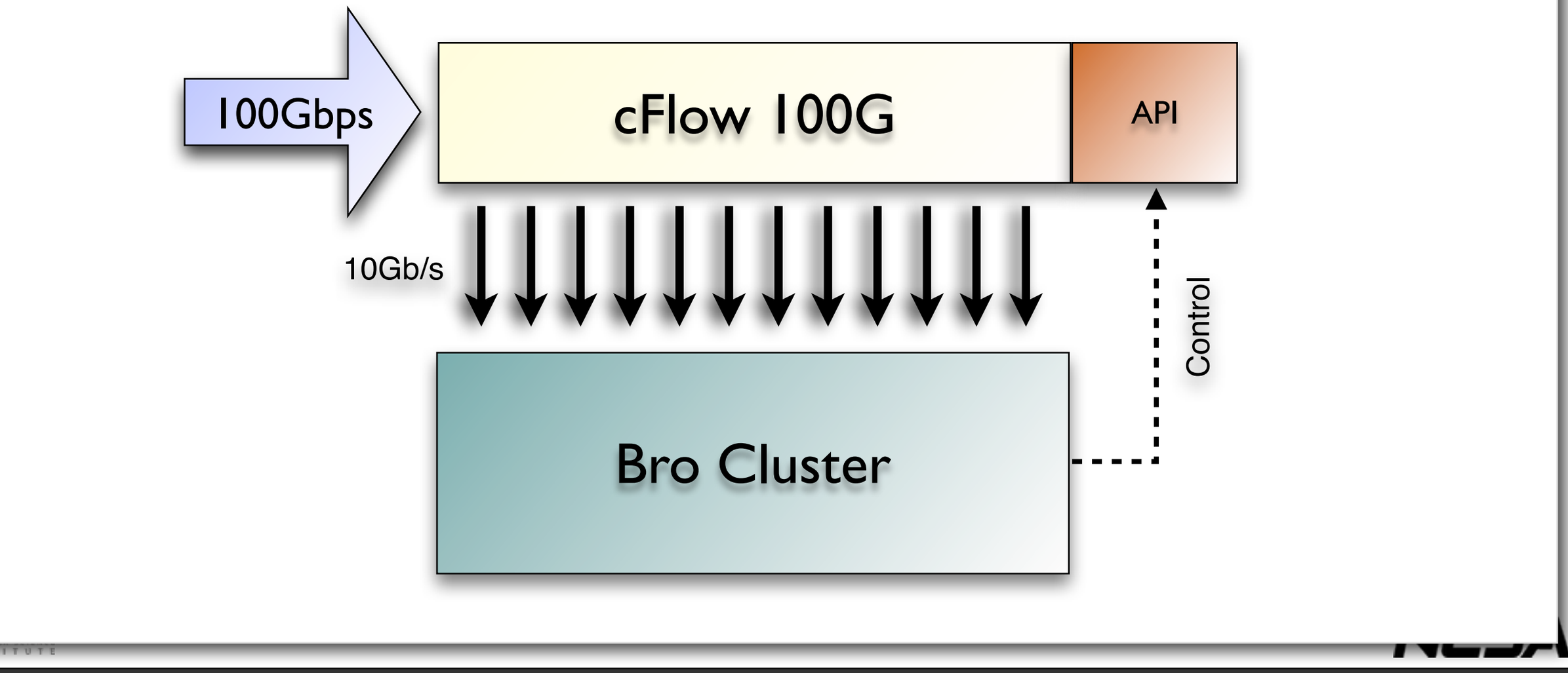
Source: ESNet

100 Gb/s Load-balancer

100 Gb/s Load-balancer

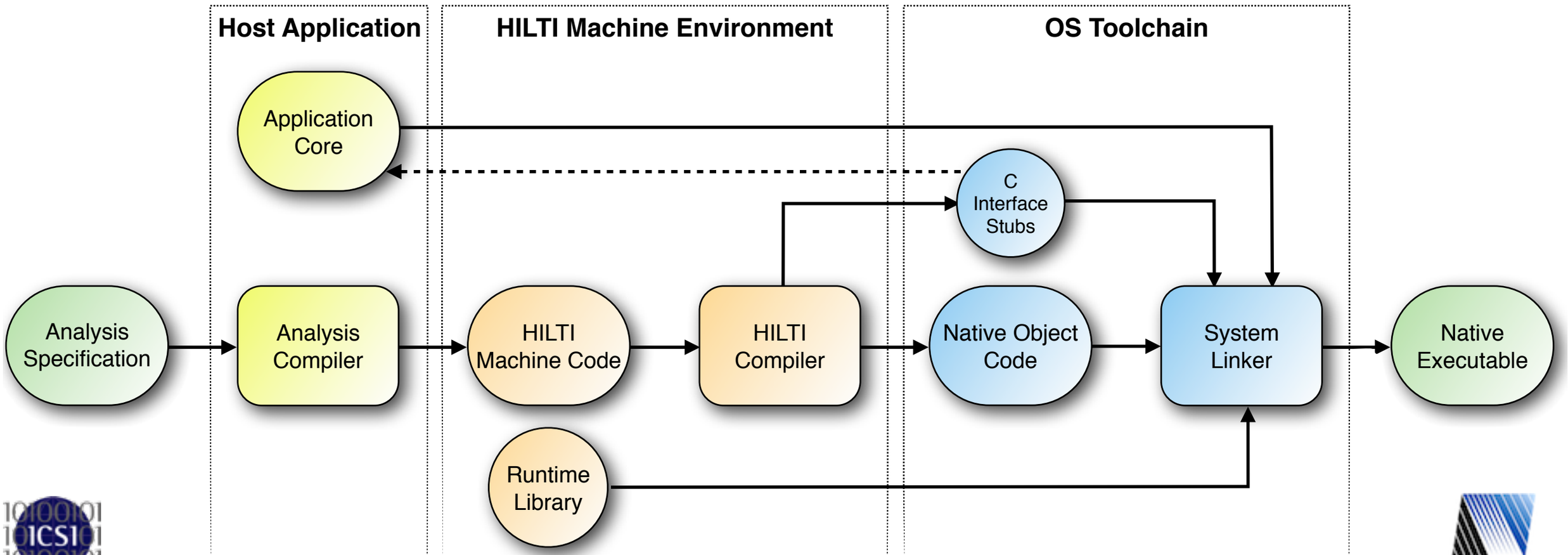


100 Gb/s Load-balancer



Improving Bro's Performance

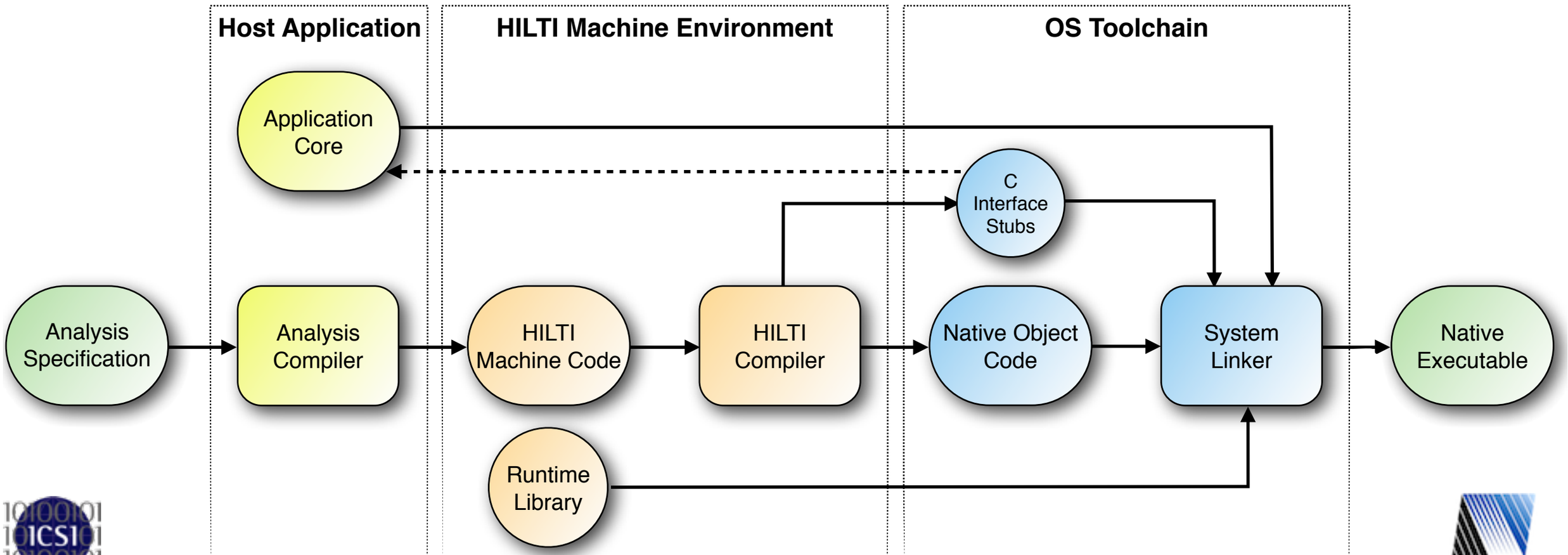
A **H**igh-Level **I**ntermediary **L**anguage for **T**raffic **I**nspection



Improving Bro's Performance

Bottlenecks: *Script interpretation* & *single-thread structure*

A **H**igh-Level **I**ntermediary **L**anguage for **T**raffic **I**nspection



Summary

Summary

```
www.bro-ids.org
git.bro-ids.org
tracker.bro-ids.org
@Bro_IDS on Twitter
```

Summary

Bro 2.0 is a major step forward.

- From research to operations.
- Crucial engineering resources available.
- Aiming to setup a long-term development model.

```
www.bro-ids.org
git.bro-ids.org
tracker.bro-ids.org
@Bro_IDS on Twitter
```

Summary

Bro 2.0 is a major step forward.

- From research to operations.
- Crucial engineering resources available.
- Aiming to setup a long-term development model.

Bro remains a research platform.

- Real-time intelligence
- Performance for next-gen environments

```
www.bro-ids.org
git.bro-ids.org
tracker.bro-ids.org
@Bro_IDS on Twitter
```