Following the Packets: A Walk Through Bro's Internal Processing Pipeline

Robin Sommer
robin@icir.org

Corelight, Inc.
International Computer Science Institute
Lawrence Berkeley National Laboratory

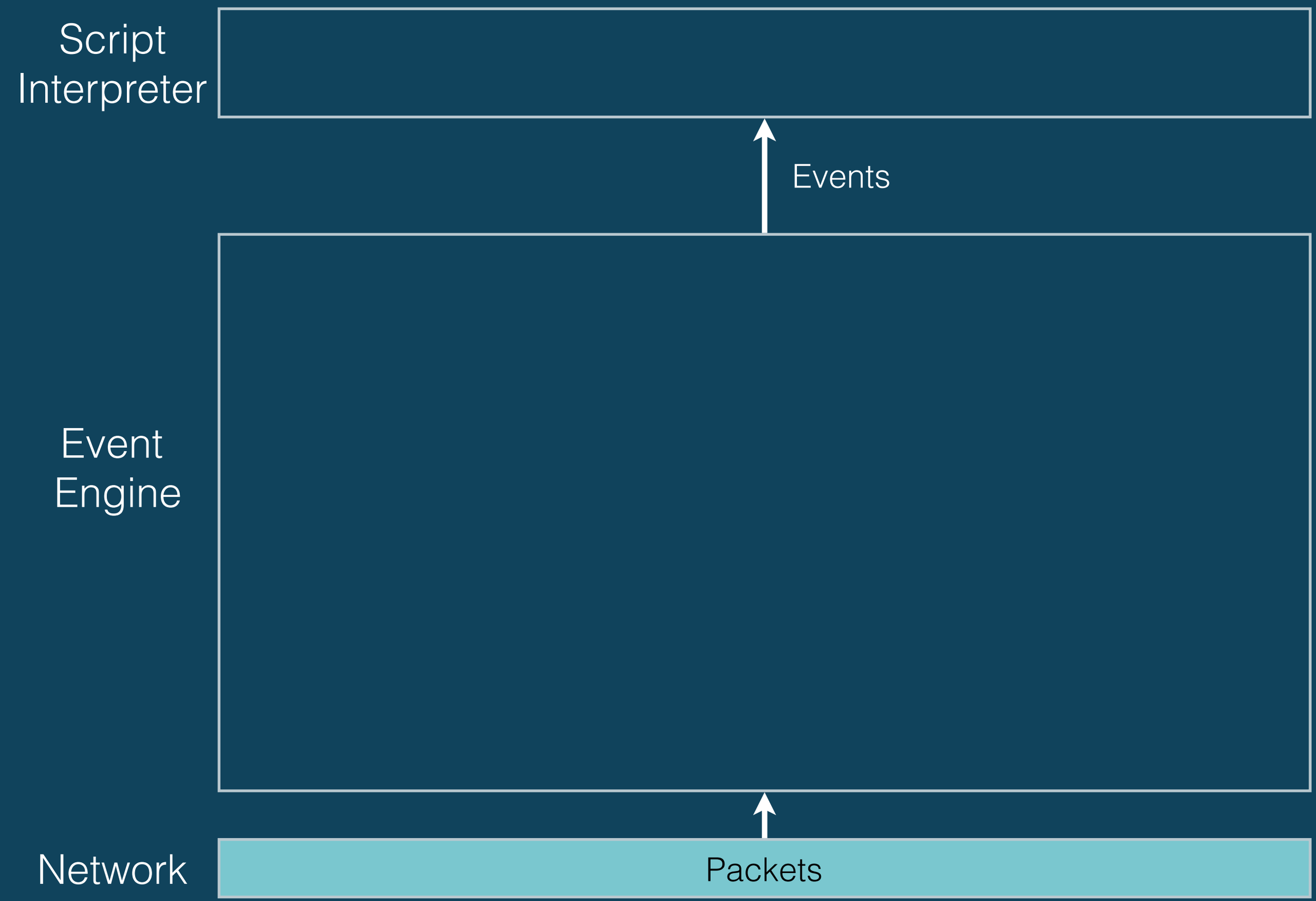
Outline

Bro's Architecture & Data Flow

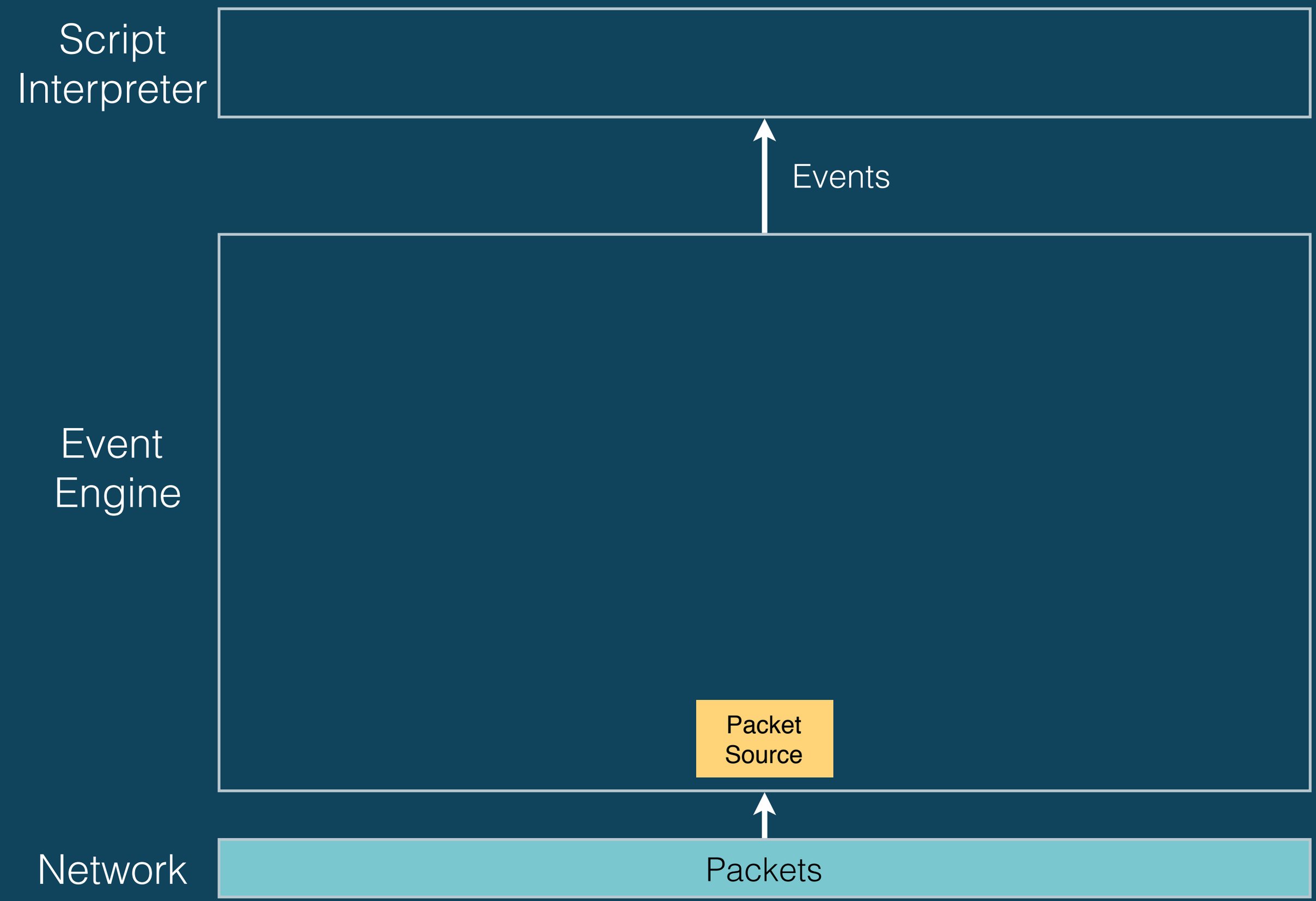
Components

- Protocol & file analysis
- Log writer & input readers
- Bro Plugins

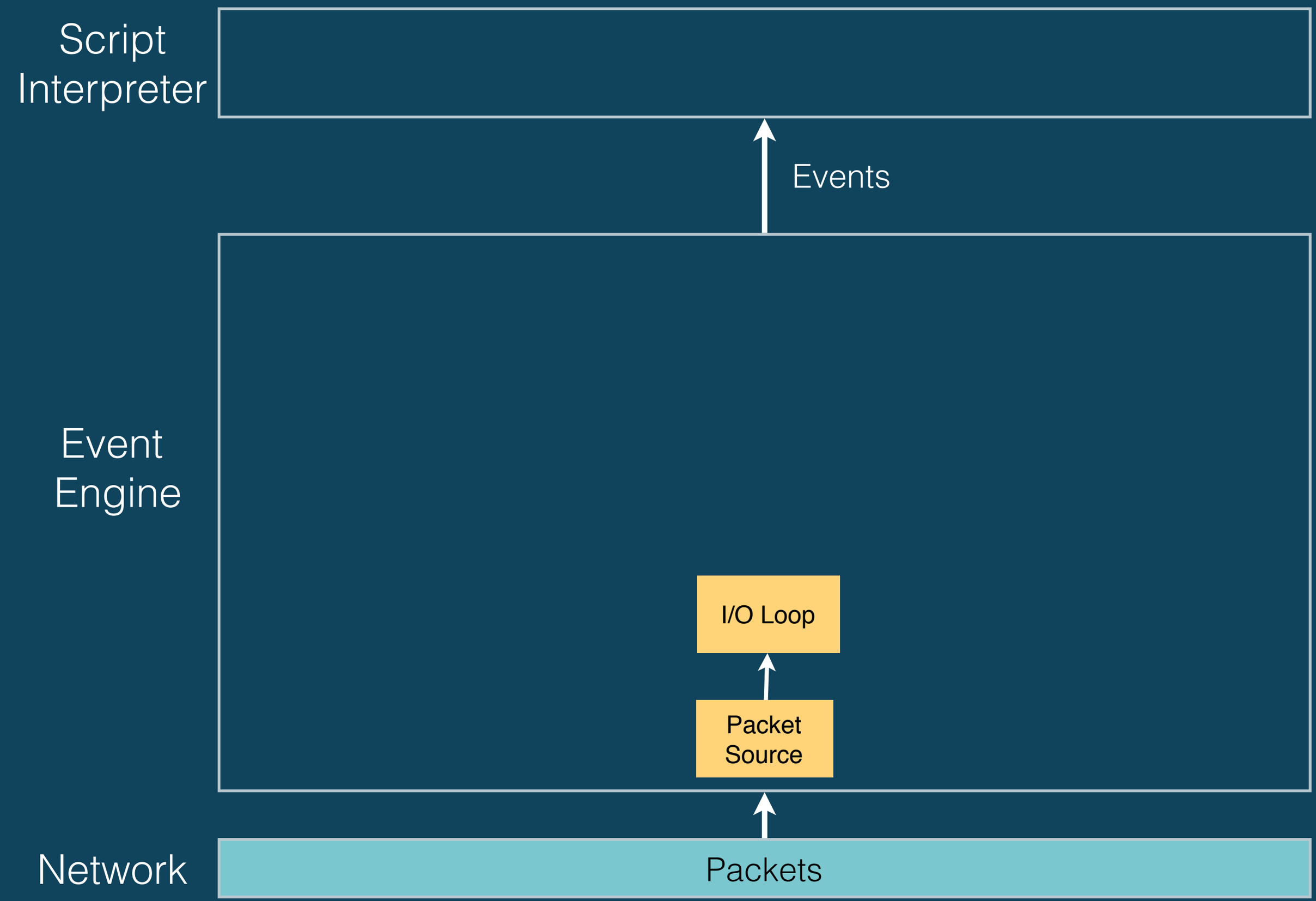
Bro Architecture



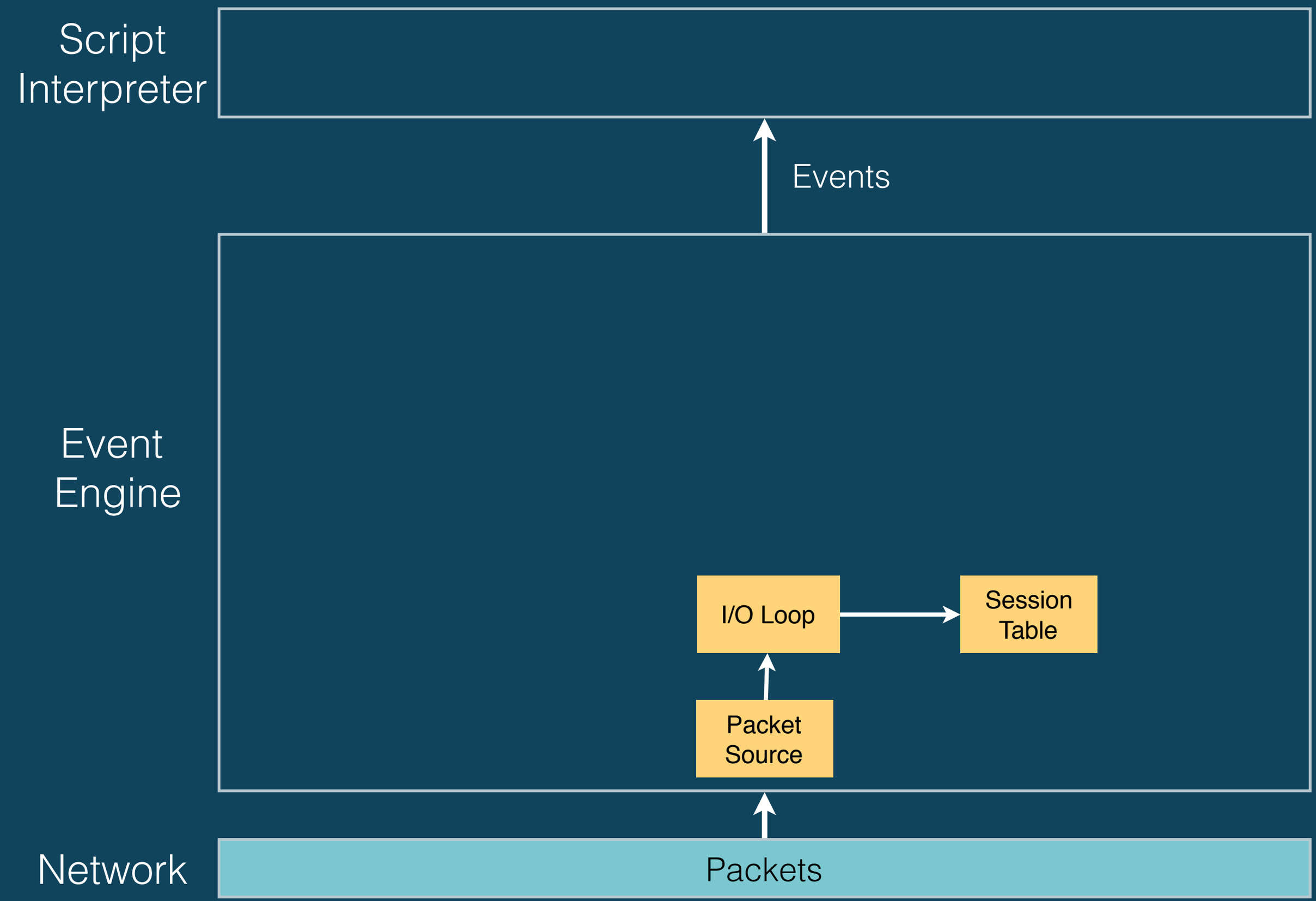
Bro Architecture



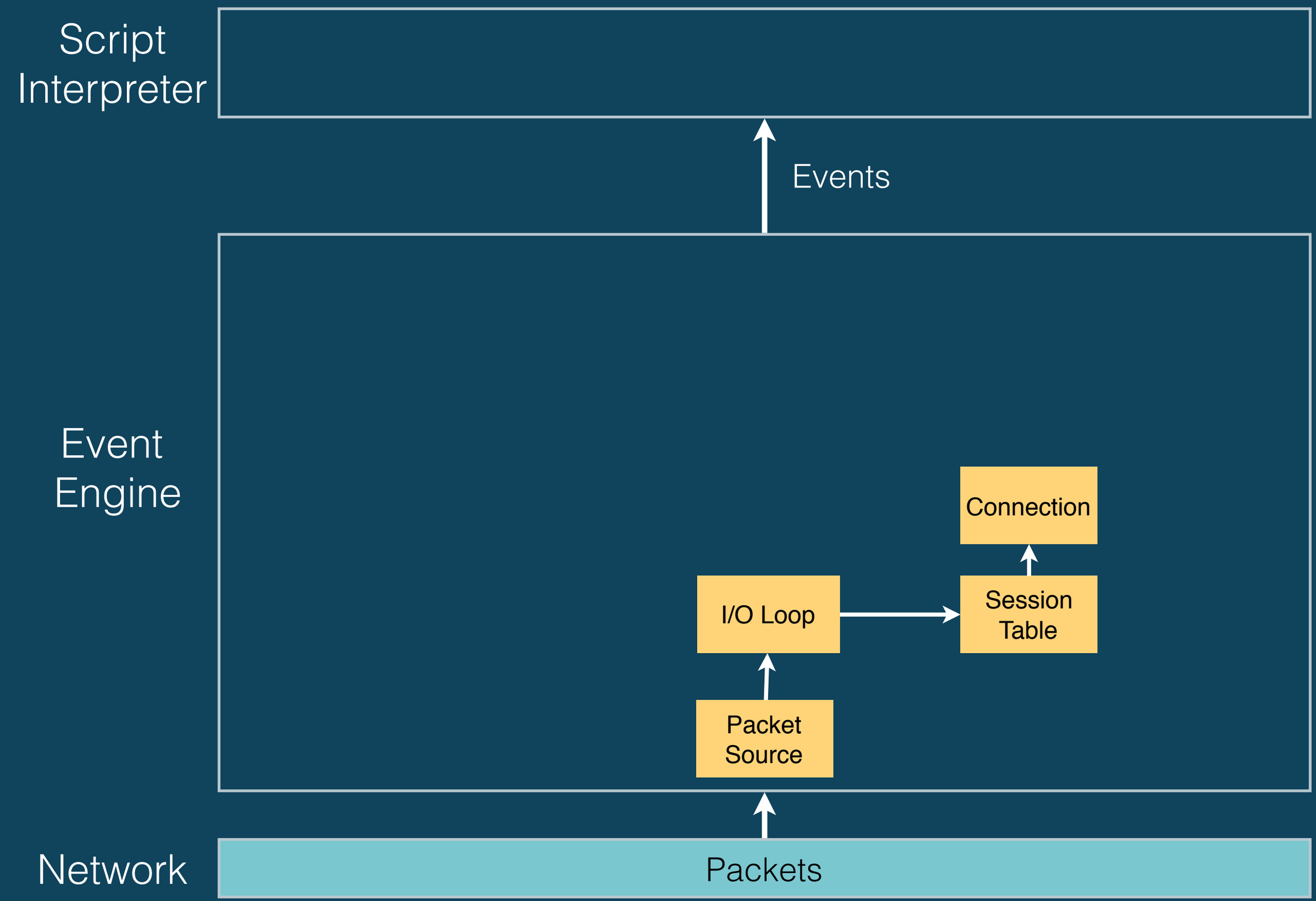
Bro Architecture



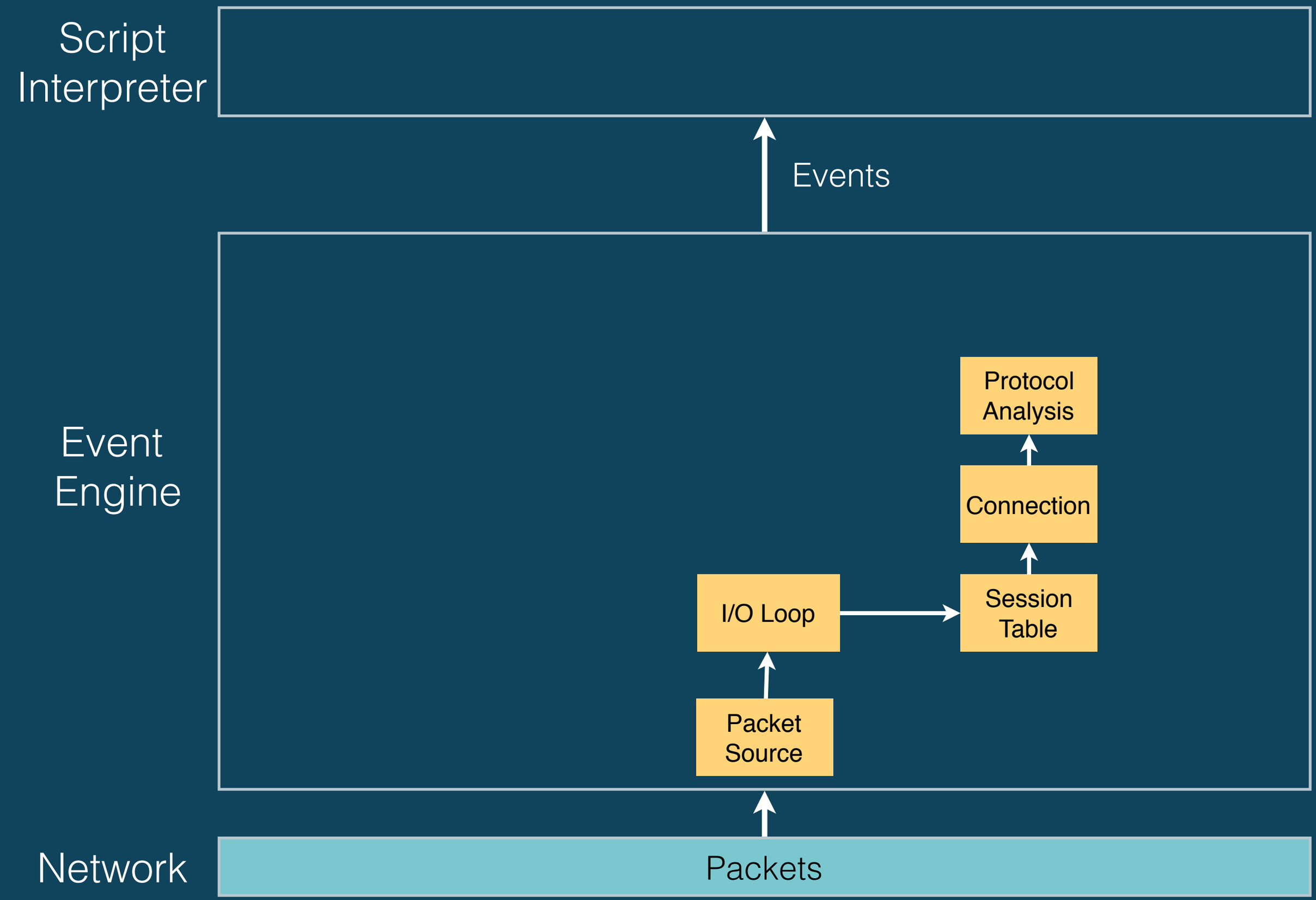
Bro Architecture



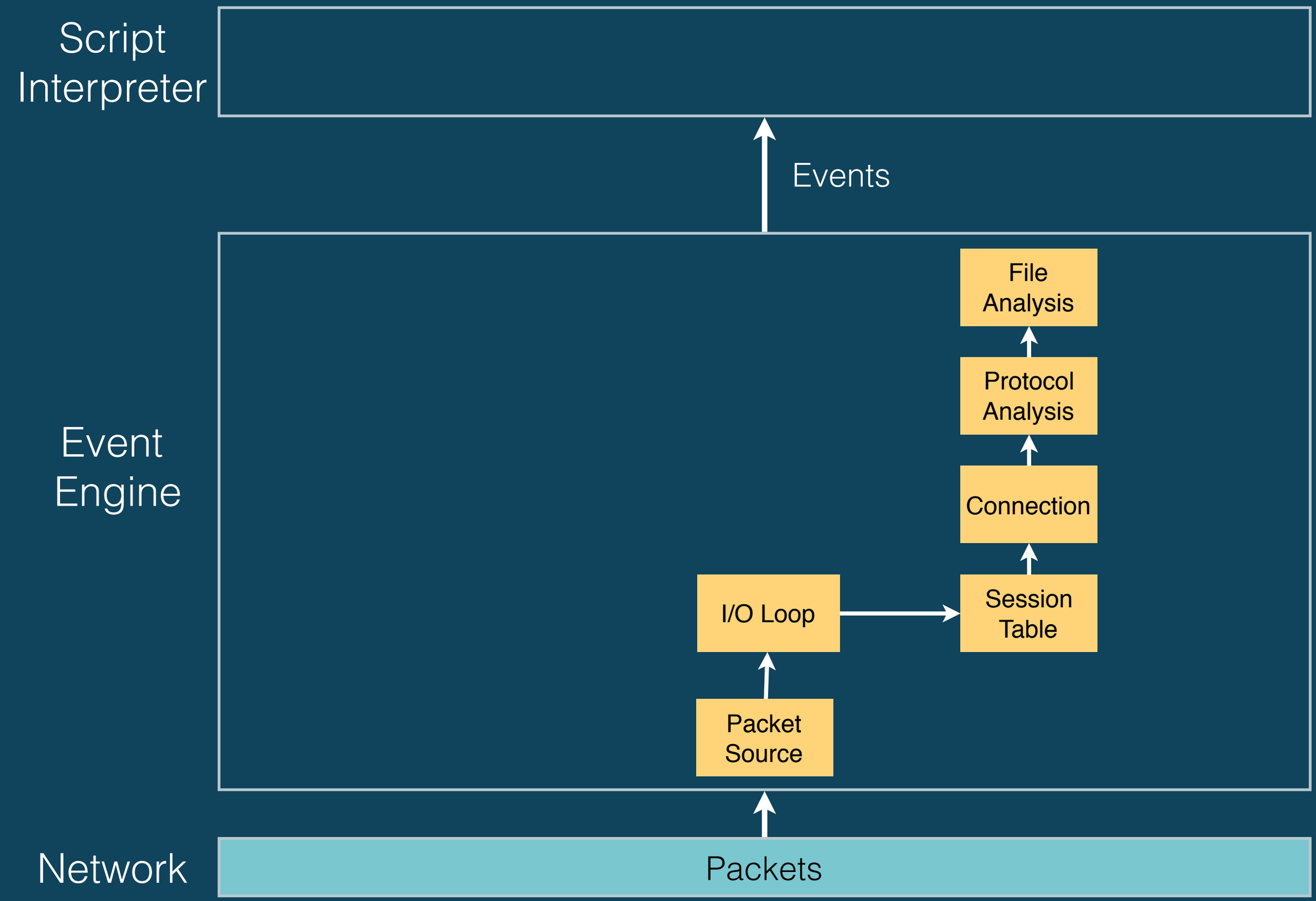
Bro Architecture



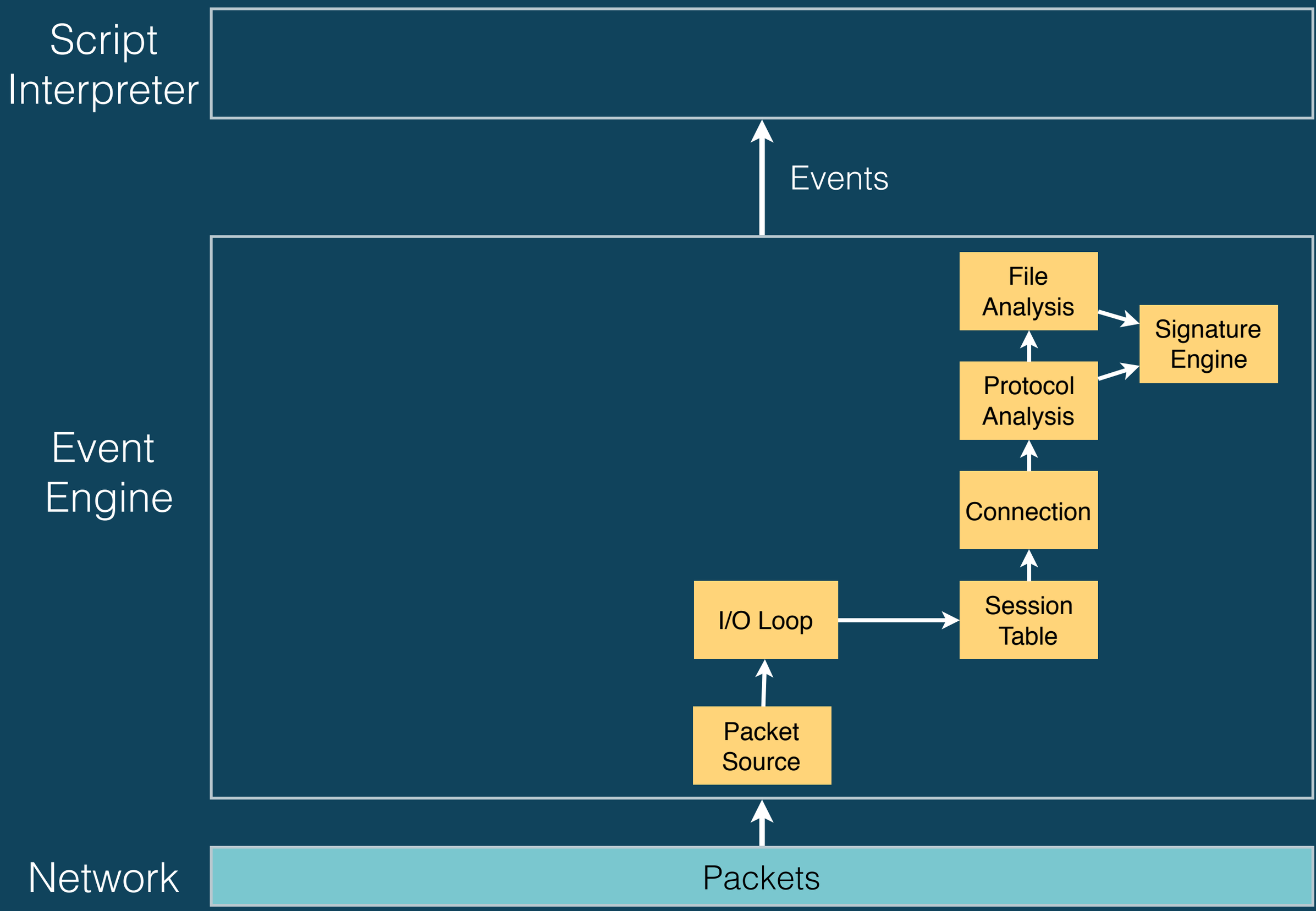
Bro Architecture



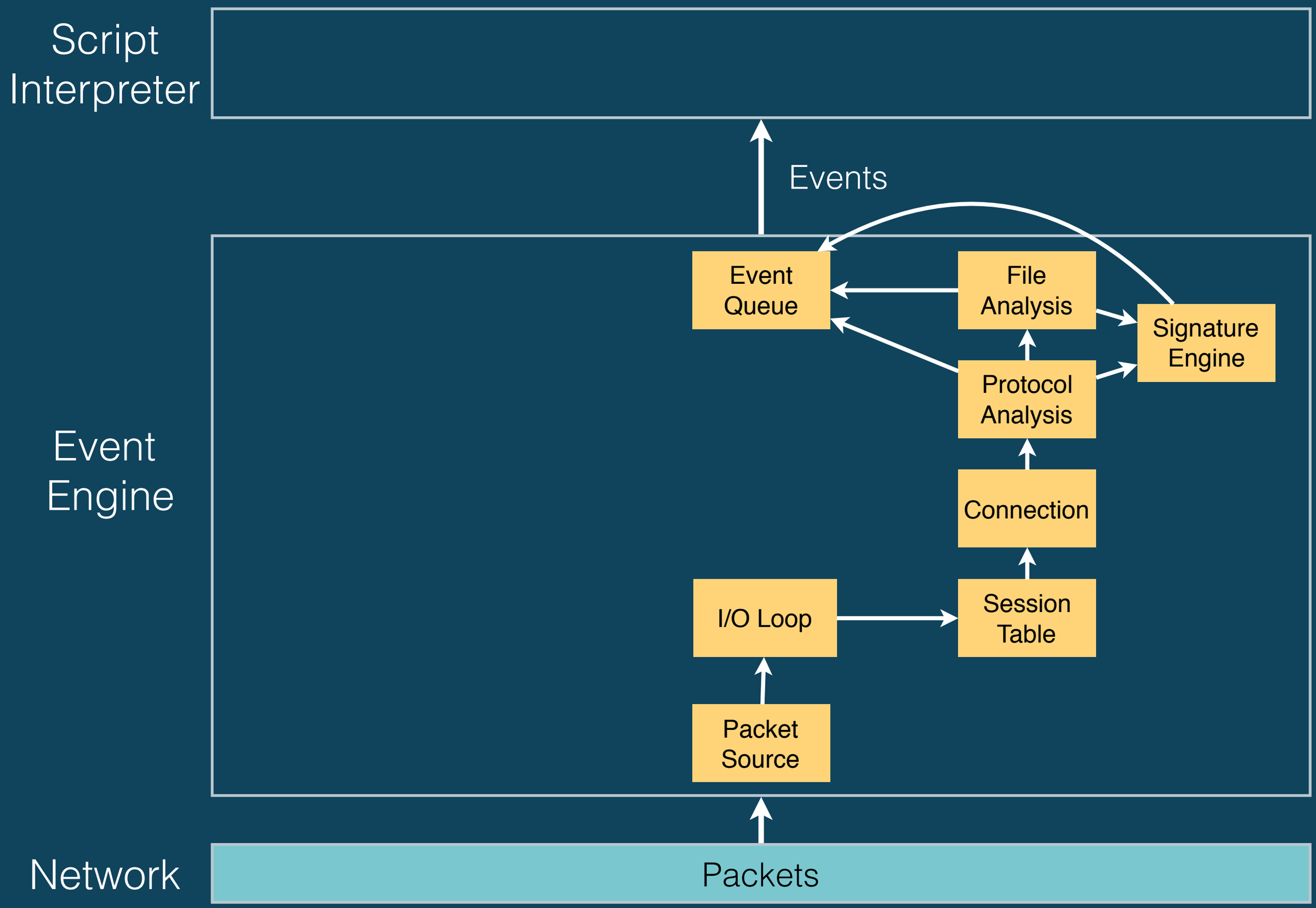
Bro Architecture



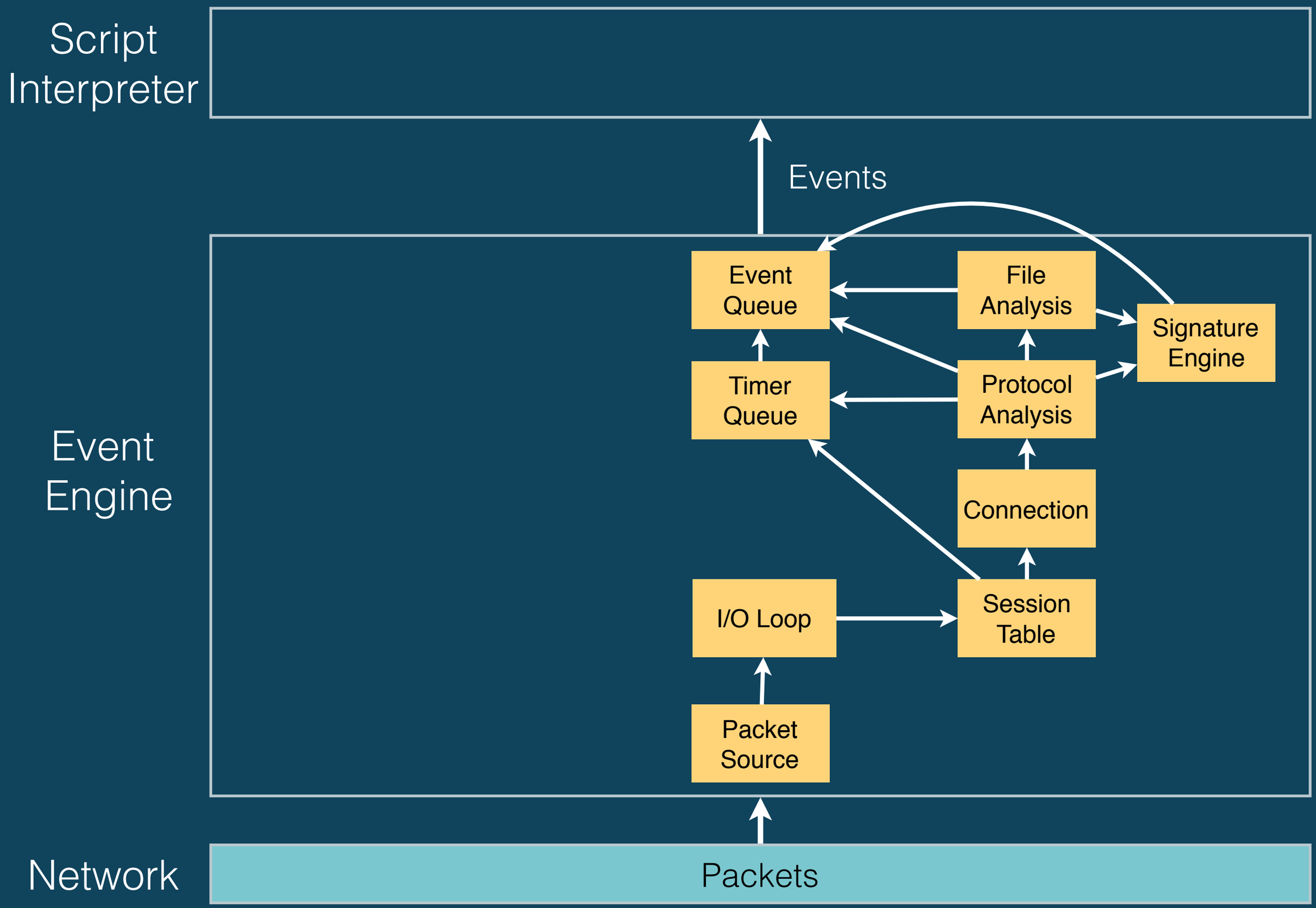
Bro Architecture



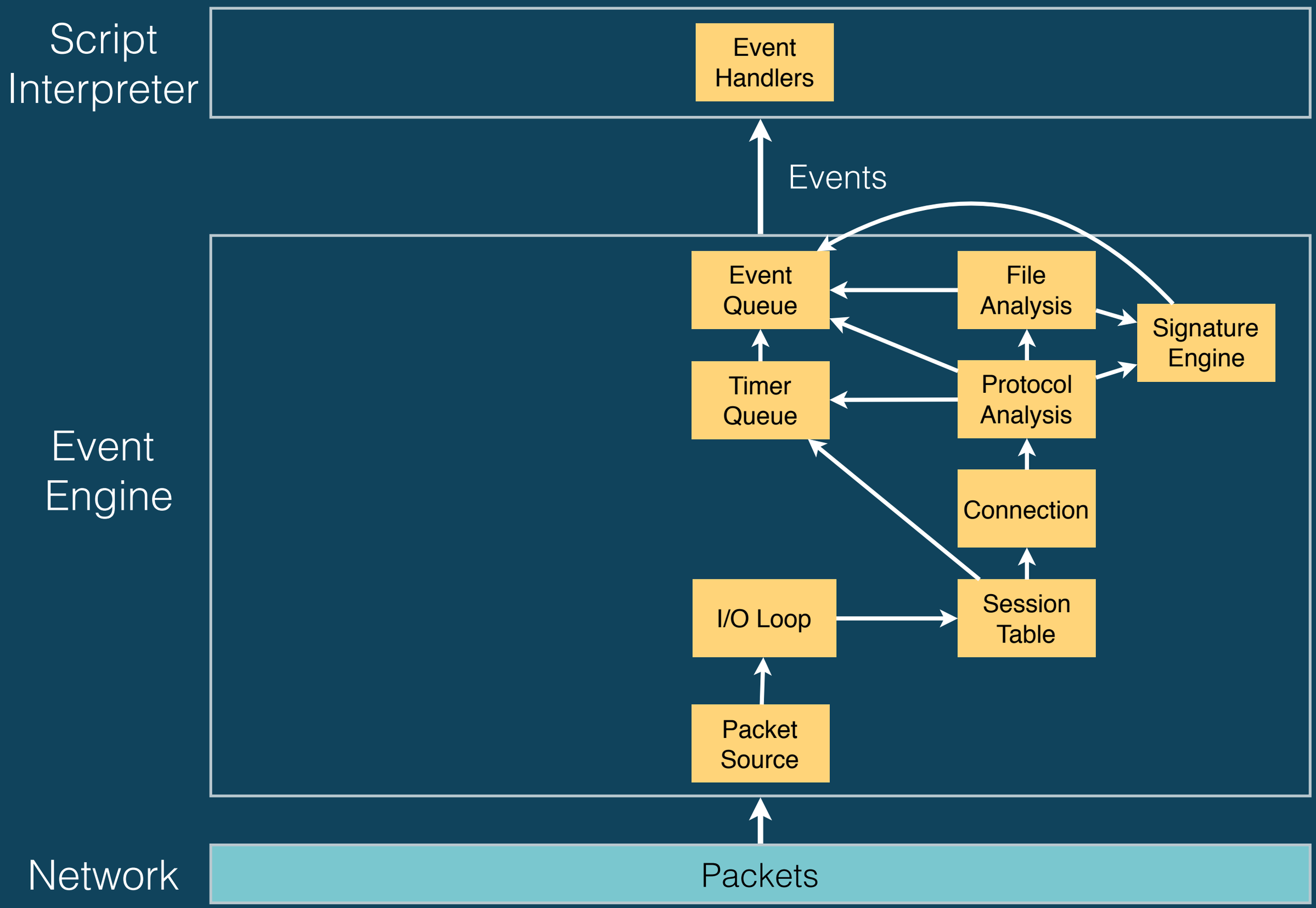
Bro Architecture



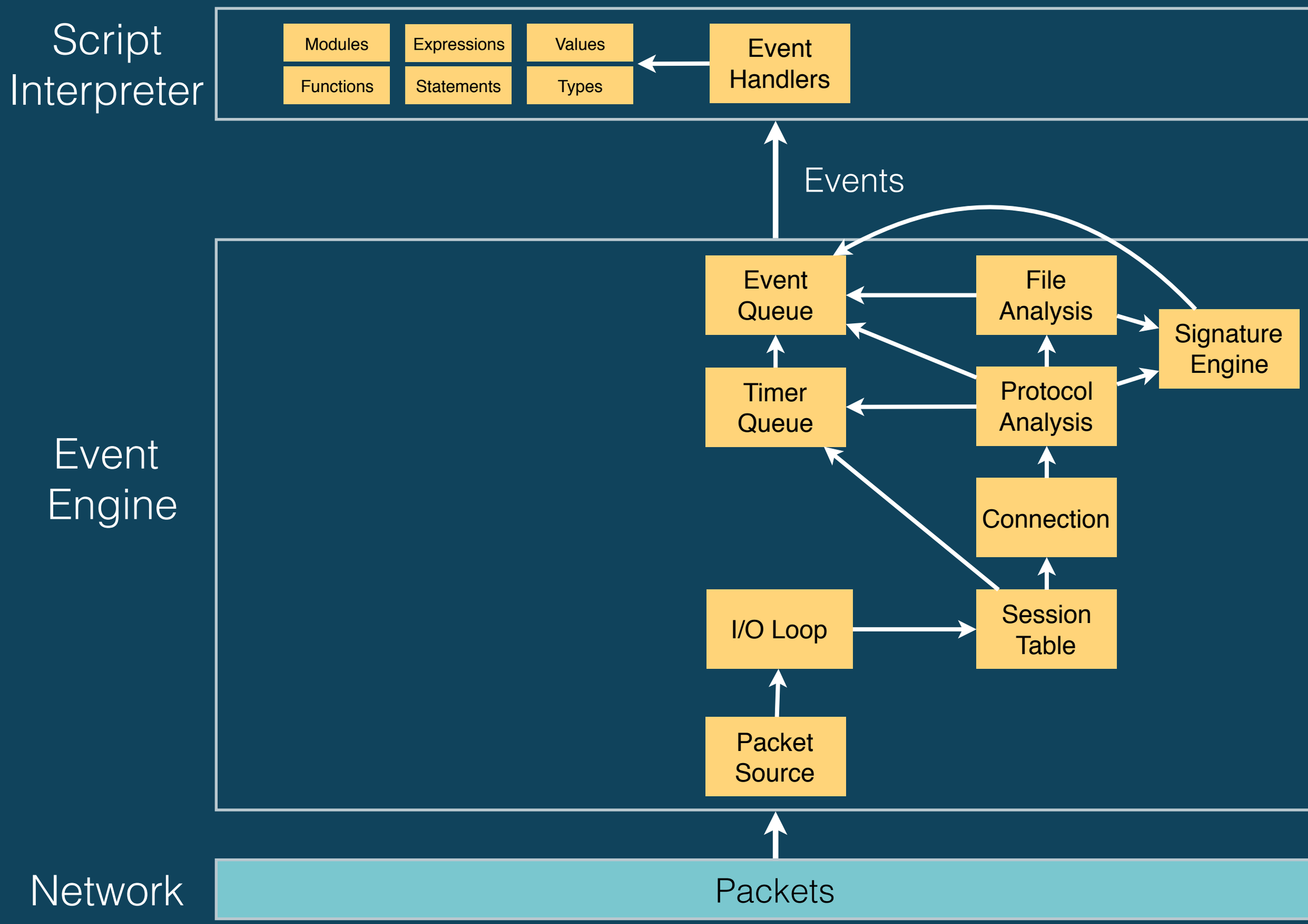
Bro Architecture



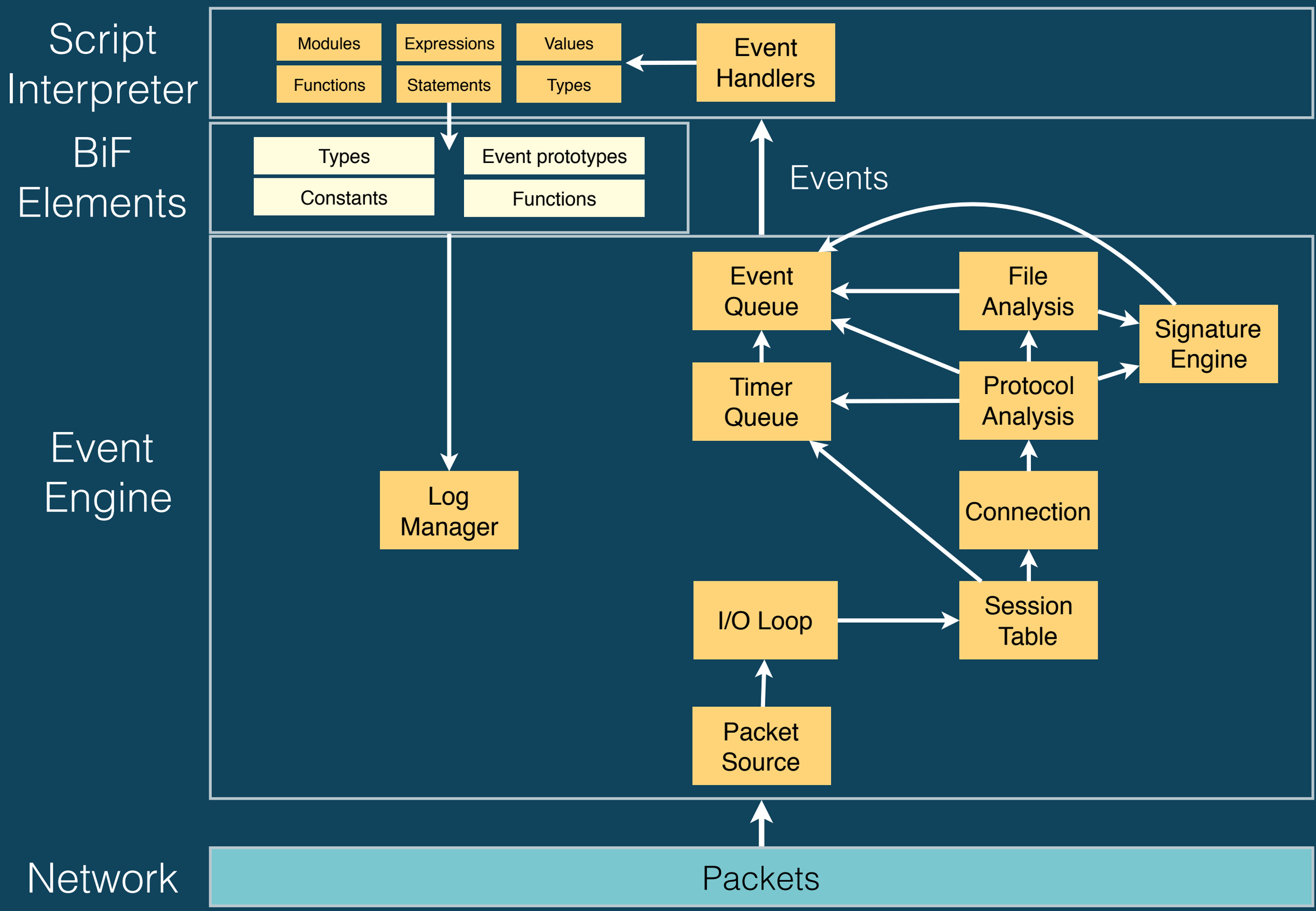
Bro Architecture



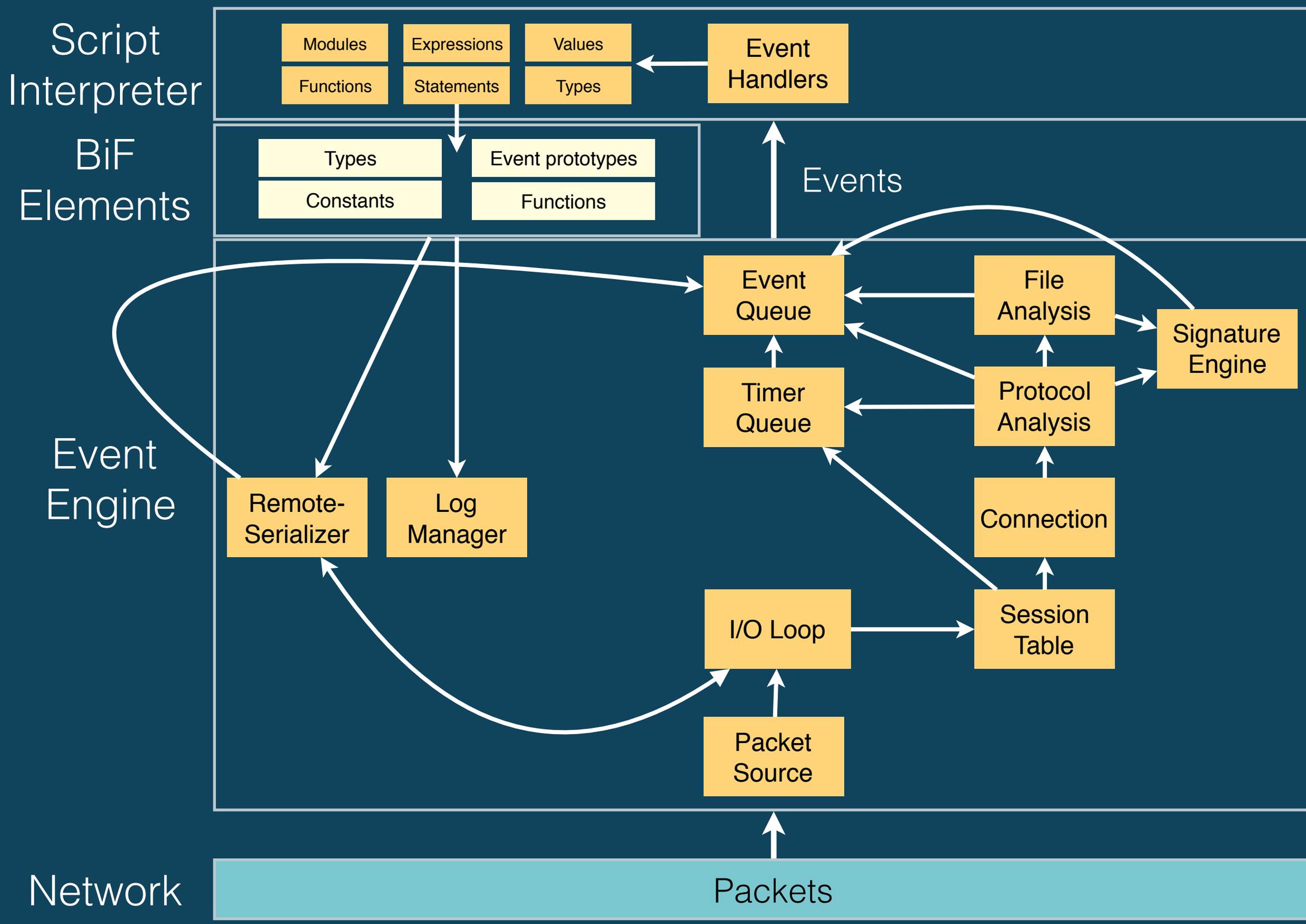
Bro Architecture



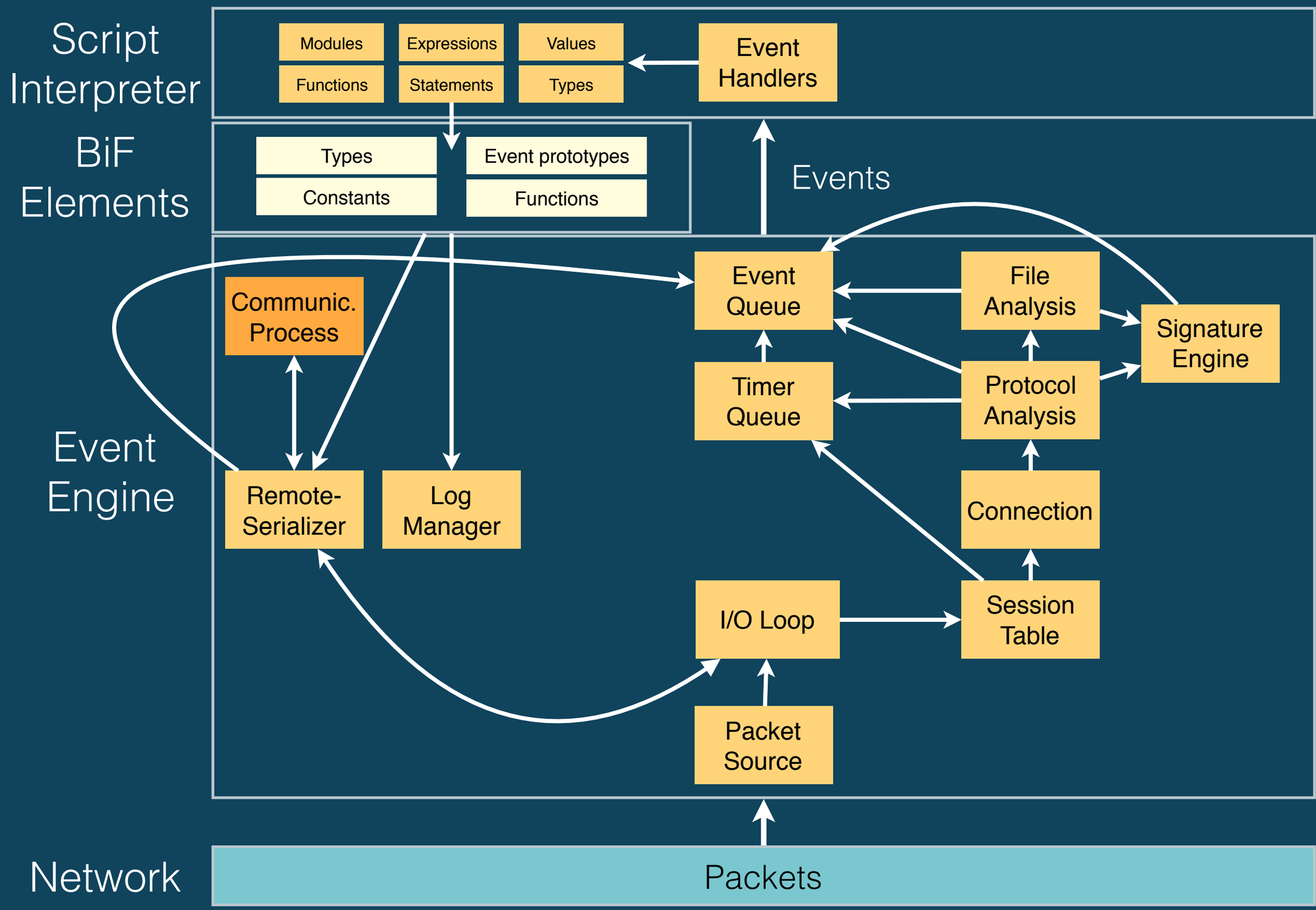
Bro Architecture



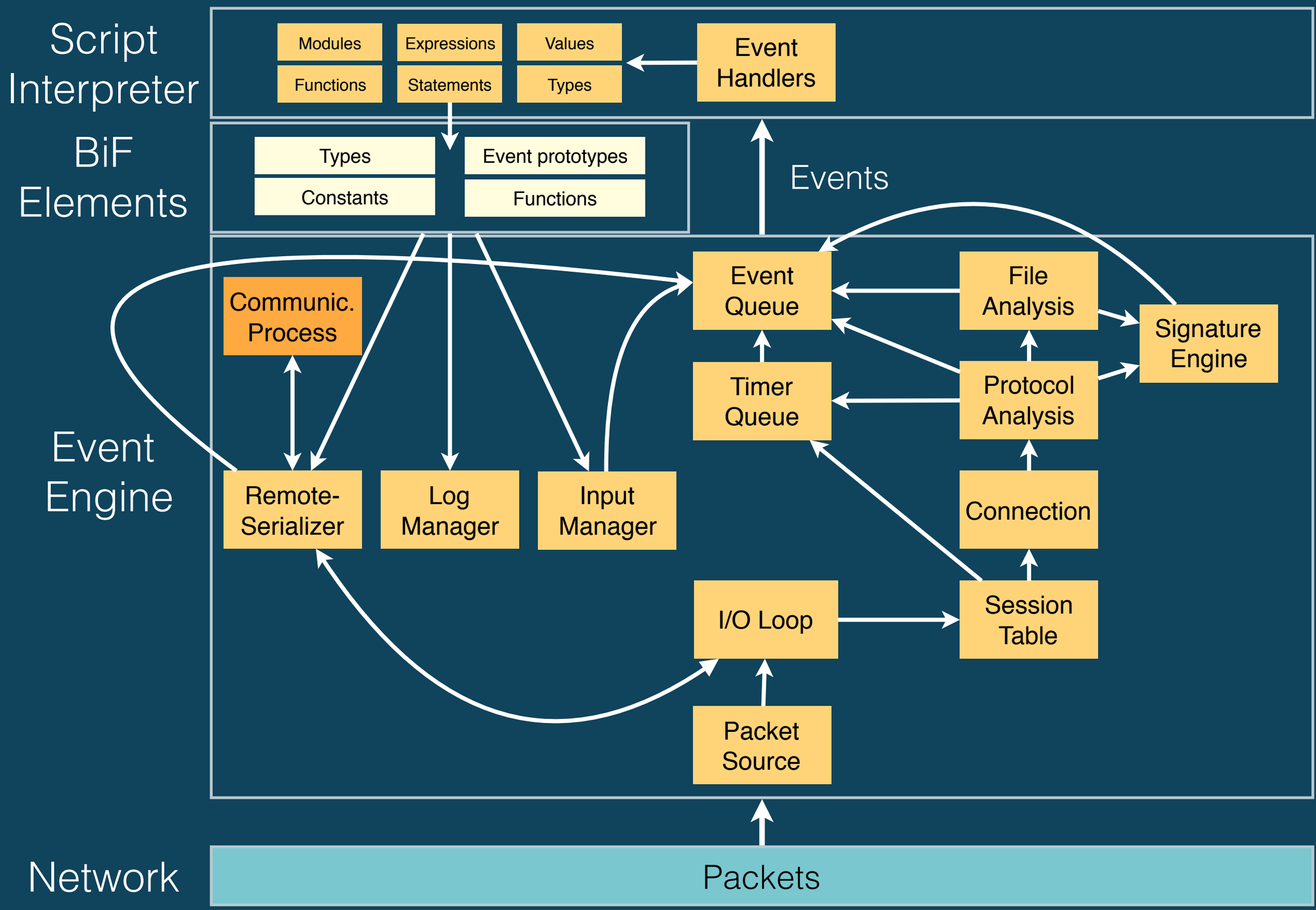
Bro Architecture



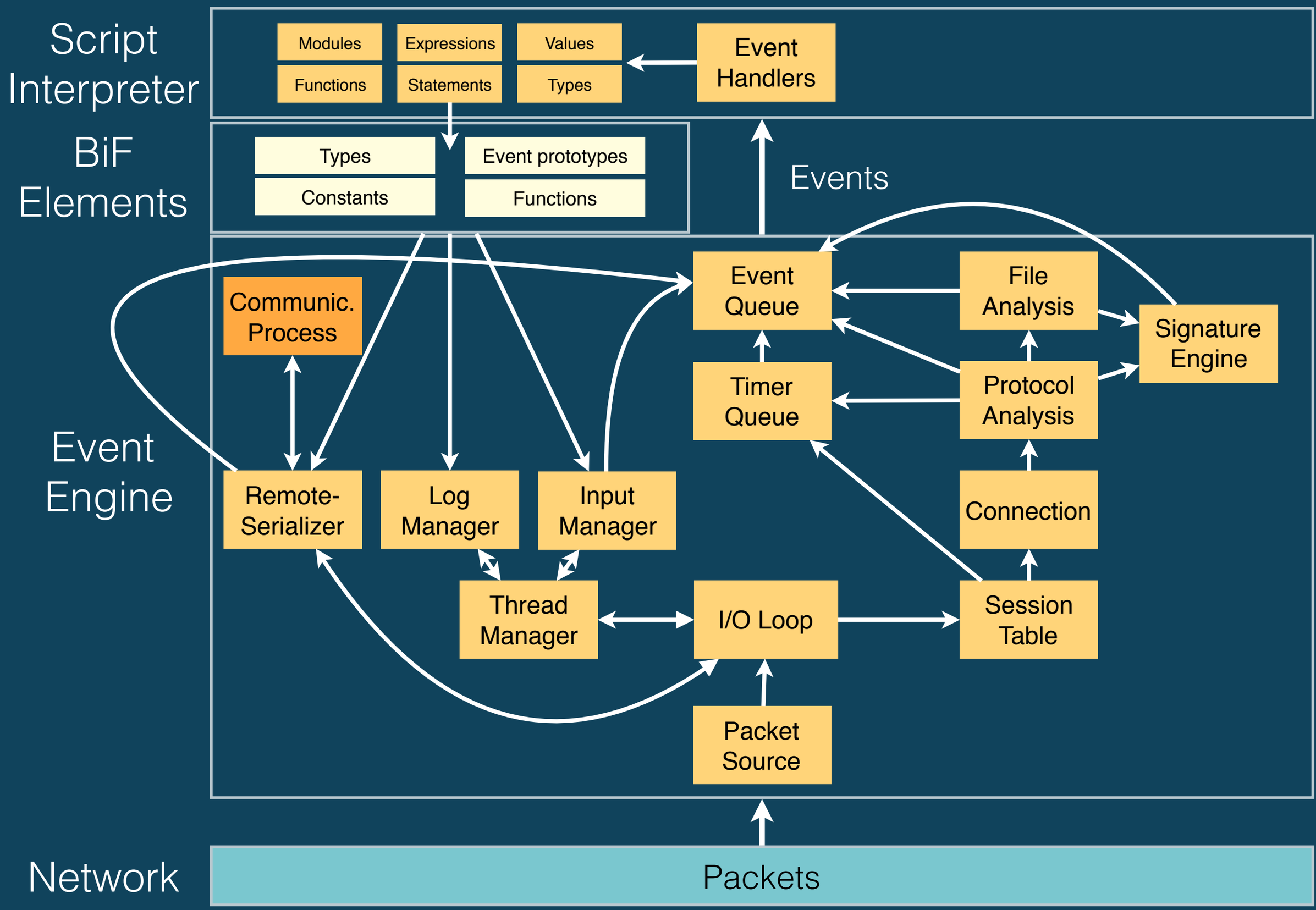
Bro Architecture



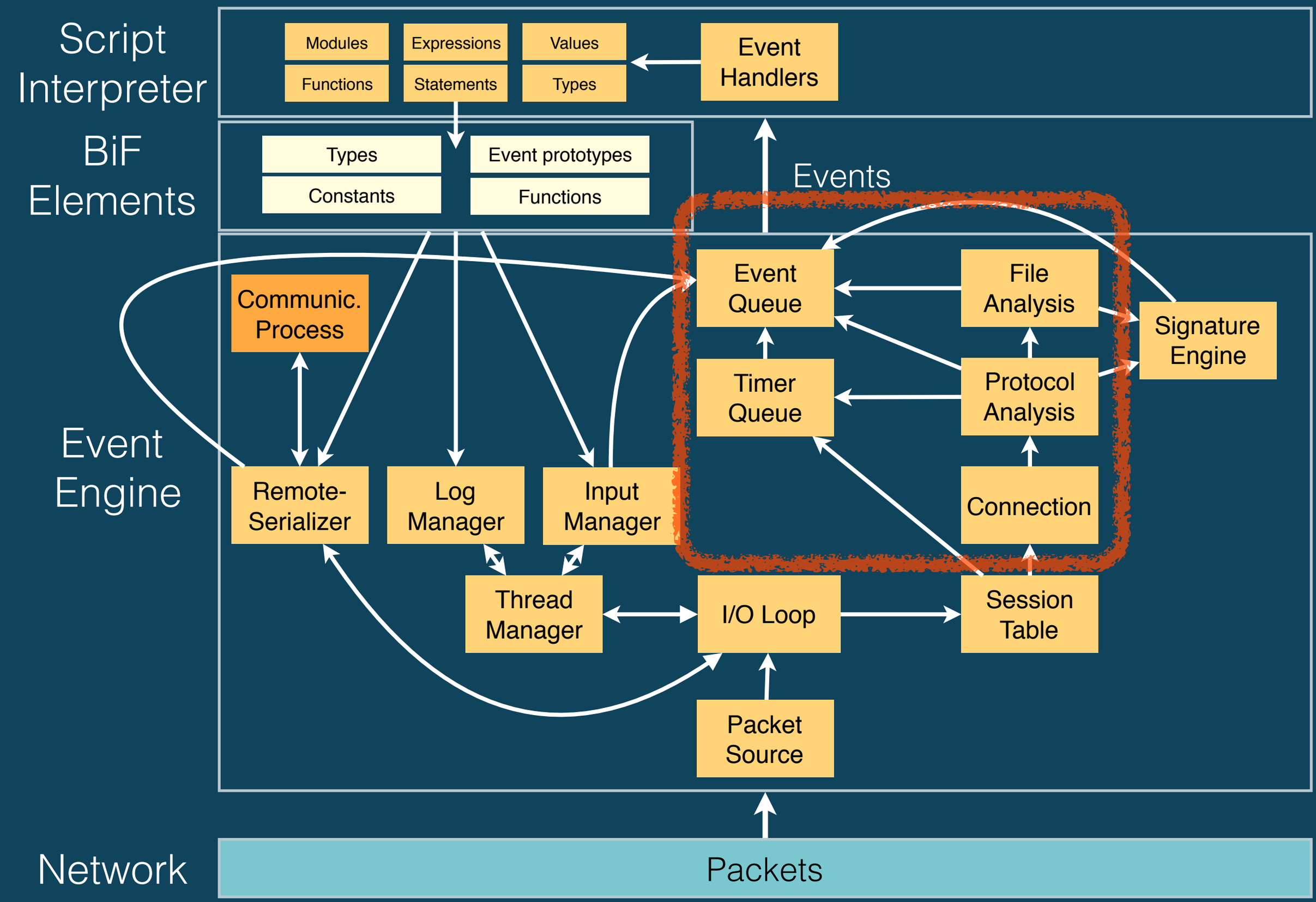
Bro Architecture



Bro Architecture

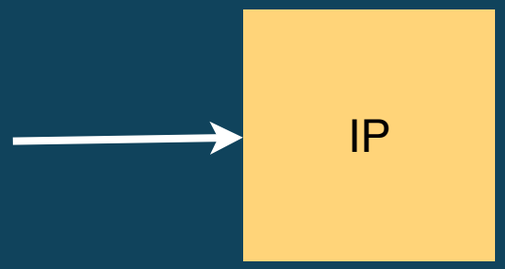


Bro Architecture



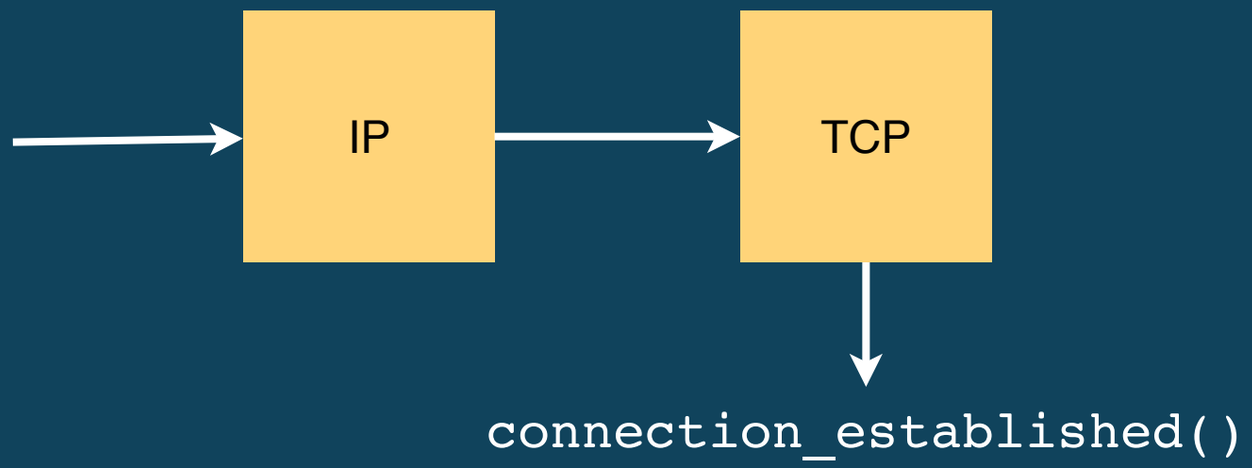
Protocol & File Analysis

Example: SSL Session



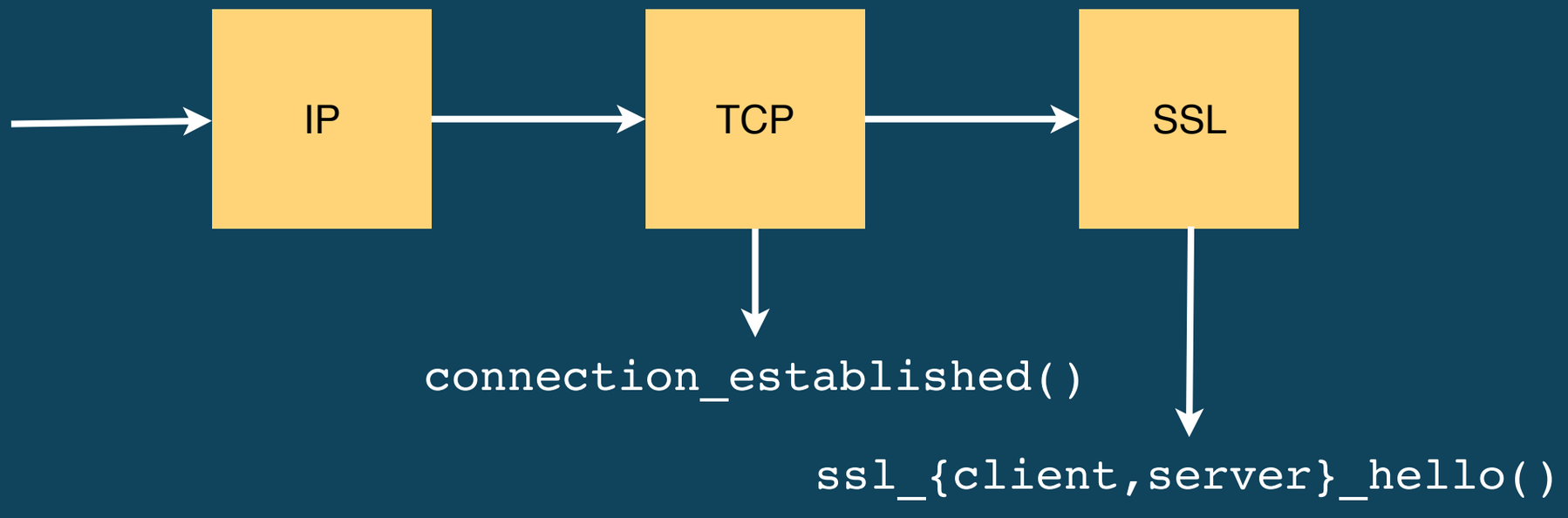
Protocol & File Analysis

Example: SSL Session



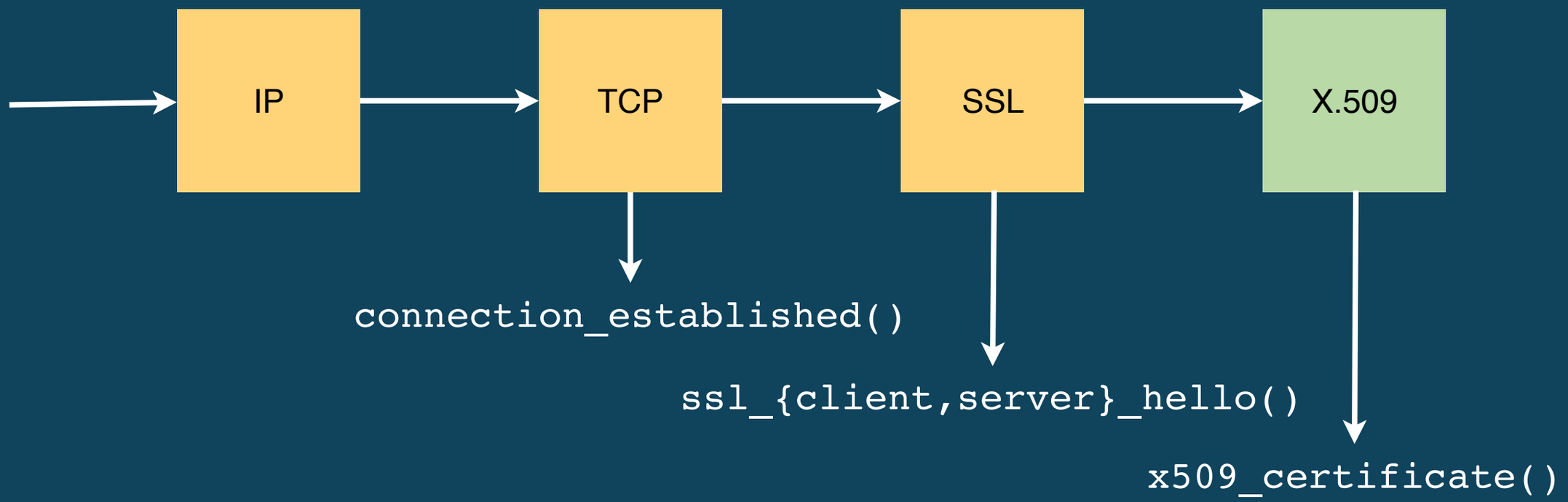
Protocol & File Analysis

Example: SSL Session



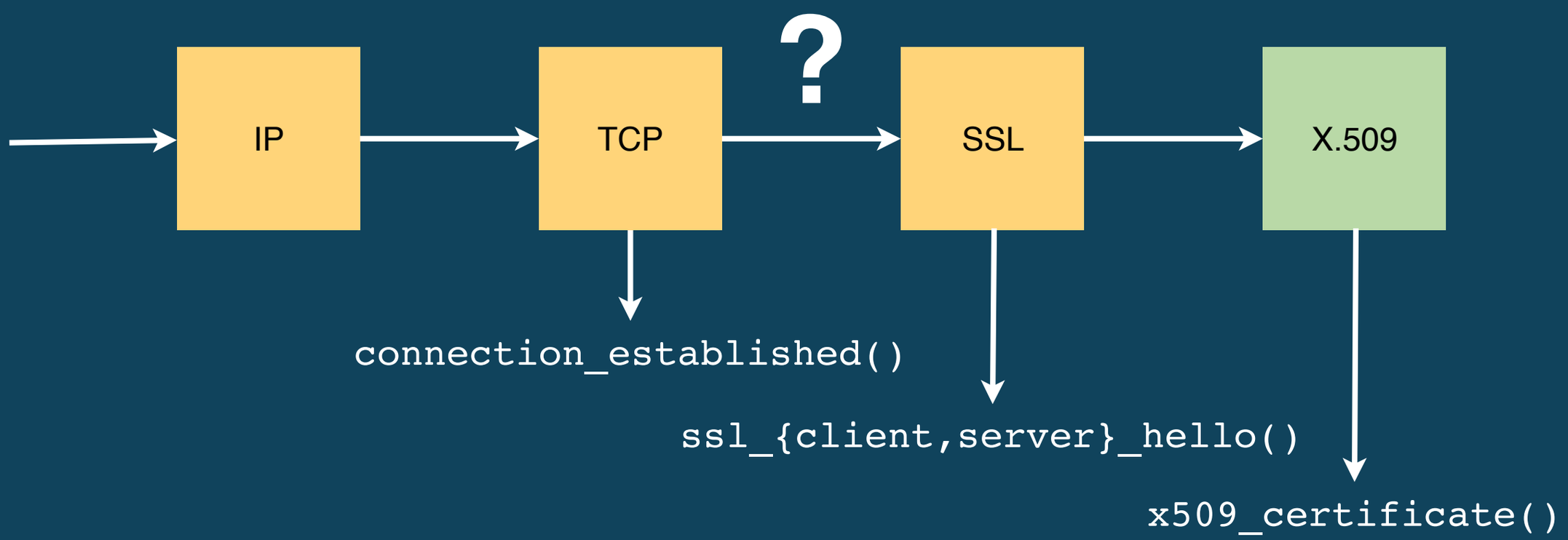
Protocol & File Analysis

Example: SSL Session

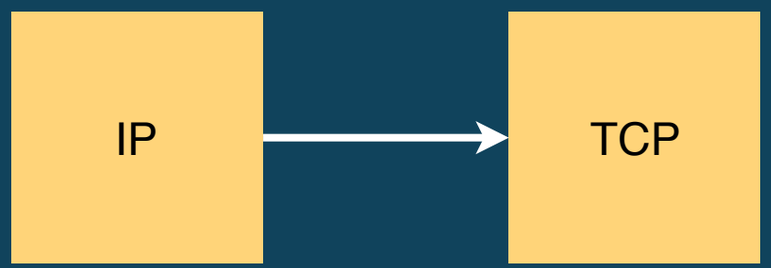


Protocol & File Analysis

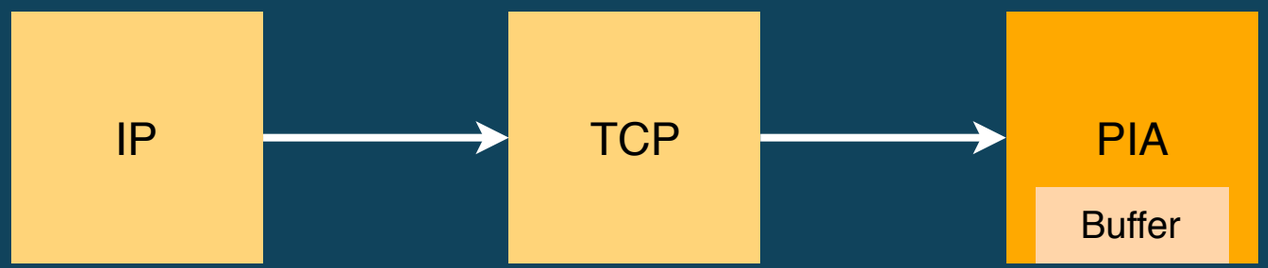
Example: SSL Session



Dynamic Protocol Detection

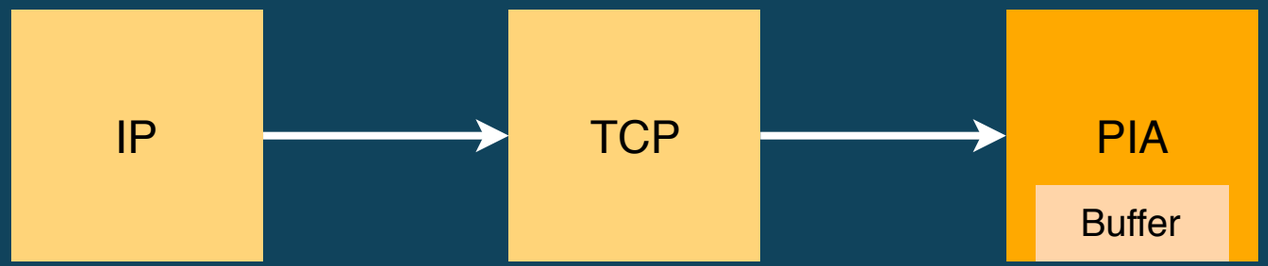


Dynamic Protocol Detection



Dynamic Protocol Detection

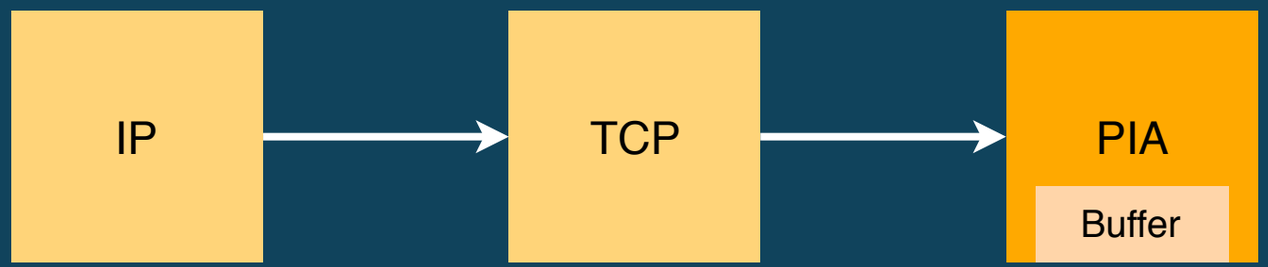
```
Analyzer::register_for_port(Analyzer::SSL, 443/tcp);
```



Dynamic Protocol Detection

```
signature dpd_ssl_server {
  ip-proto == tcp
  payload /^(\x16\x03[\x00\x01\x02\x03[…].*/
  tcp-state responder
  enable "ssl"
}
```

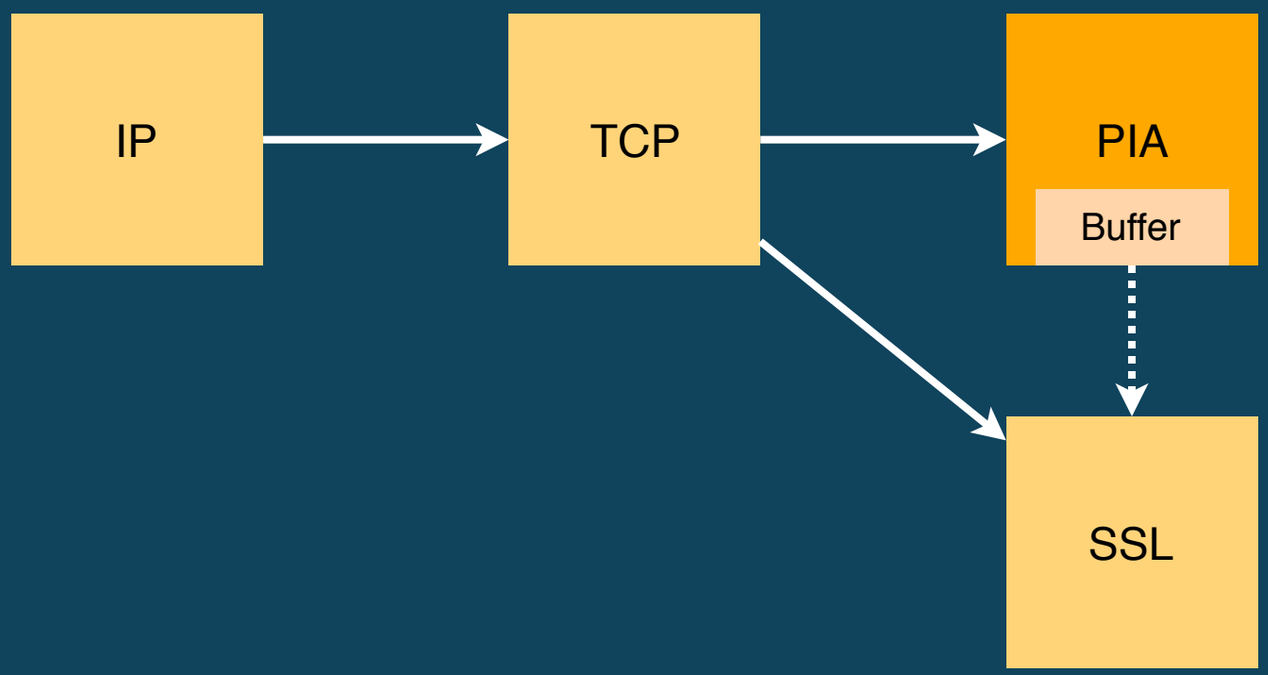
```
Analyzer::register_for_port(Analyzer::SSL, 443/tcp);
```



Dynamic Protocol Detection

```
signature dpd_ssl_server {
  ip-proto == tcp
  payload /^(\x16\x03[\x00\x01\x02\x03[…].*/
  tcp-state responder
  enable "ssl"
}
```

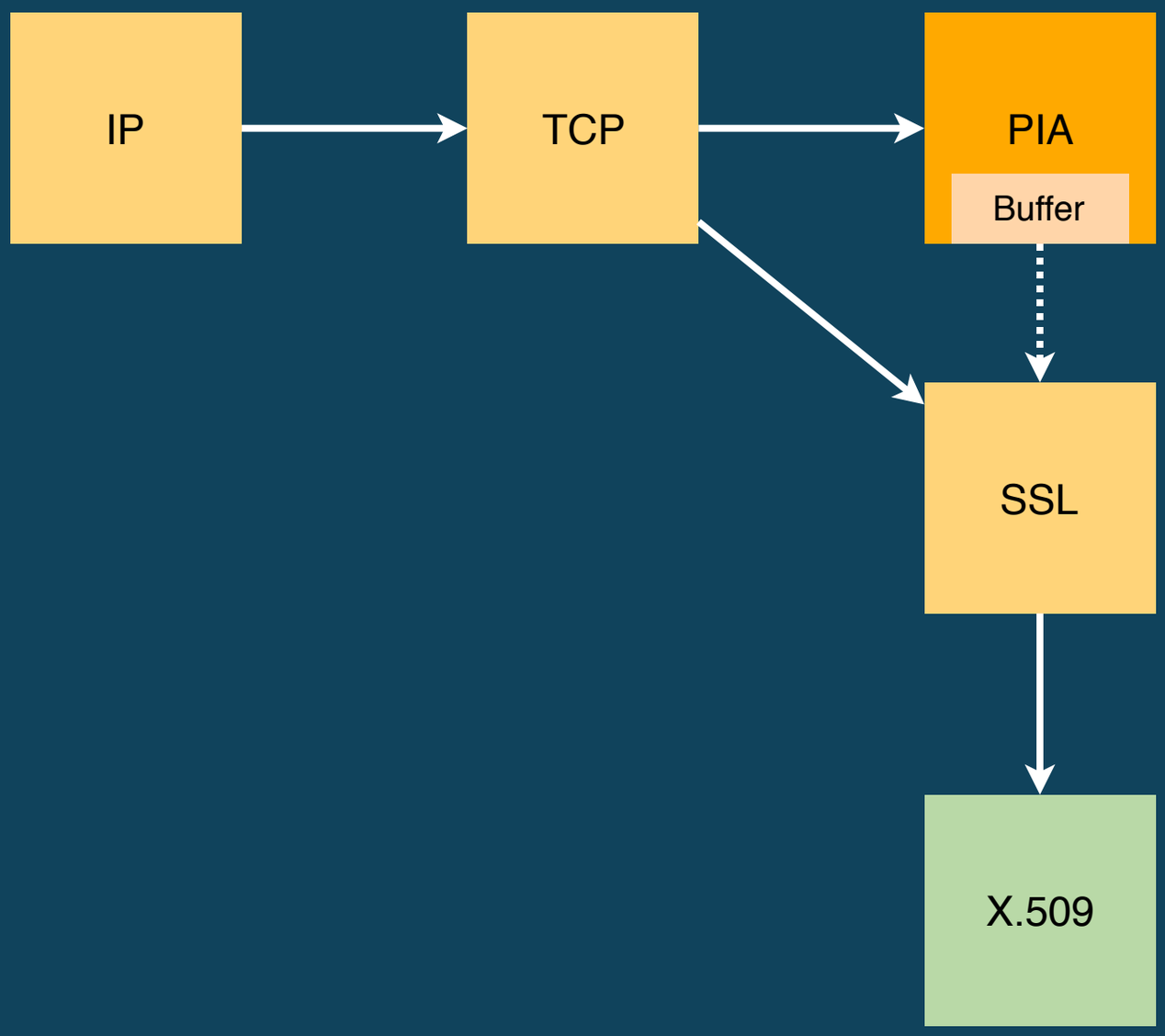
```
Analyzer::register_for_port(Analyzer::SSL, 443/tcp);
```



Dynamic Protocol Detection

```
signature dpd_ssl_server {
  ip-proto == tcp
  payload /^(\x16\x03[\x00\x01\x02\x03[…].*/
  tcp-state responder
  enable "ssl"
}
```

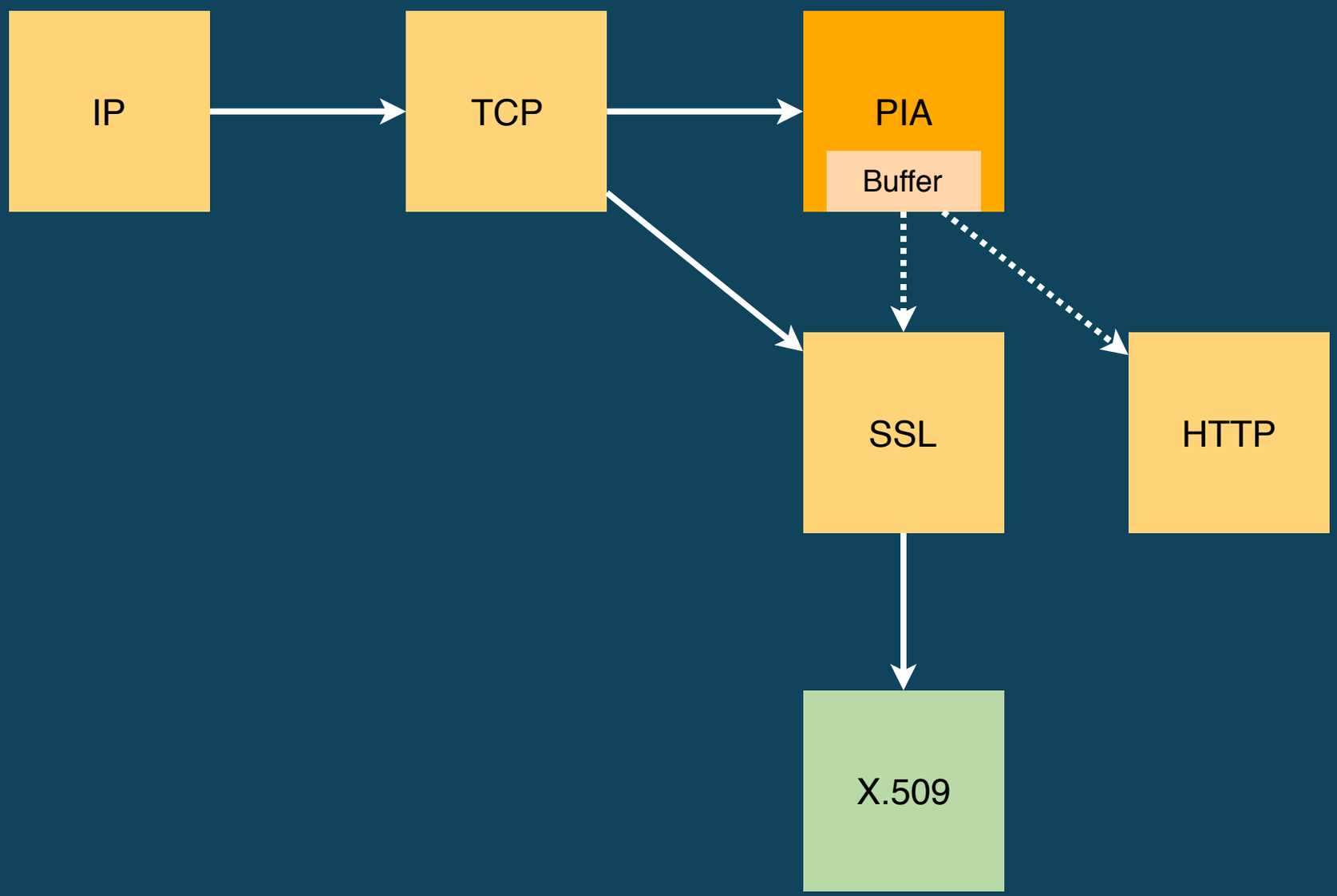
```
Analyzer::register_for_port(Analyzer::SSL, 443/tcp);
```



Dynamic Protocol Detection

```
signature dpd_ssl_server {
  ip-proto == tcp
  payload /^(\x16\x03[\x00\x01\x02\x03[…].*/
  tcp-state responder
  enable "ssl"
}
```

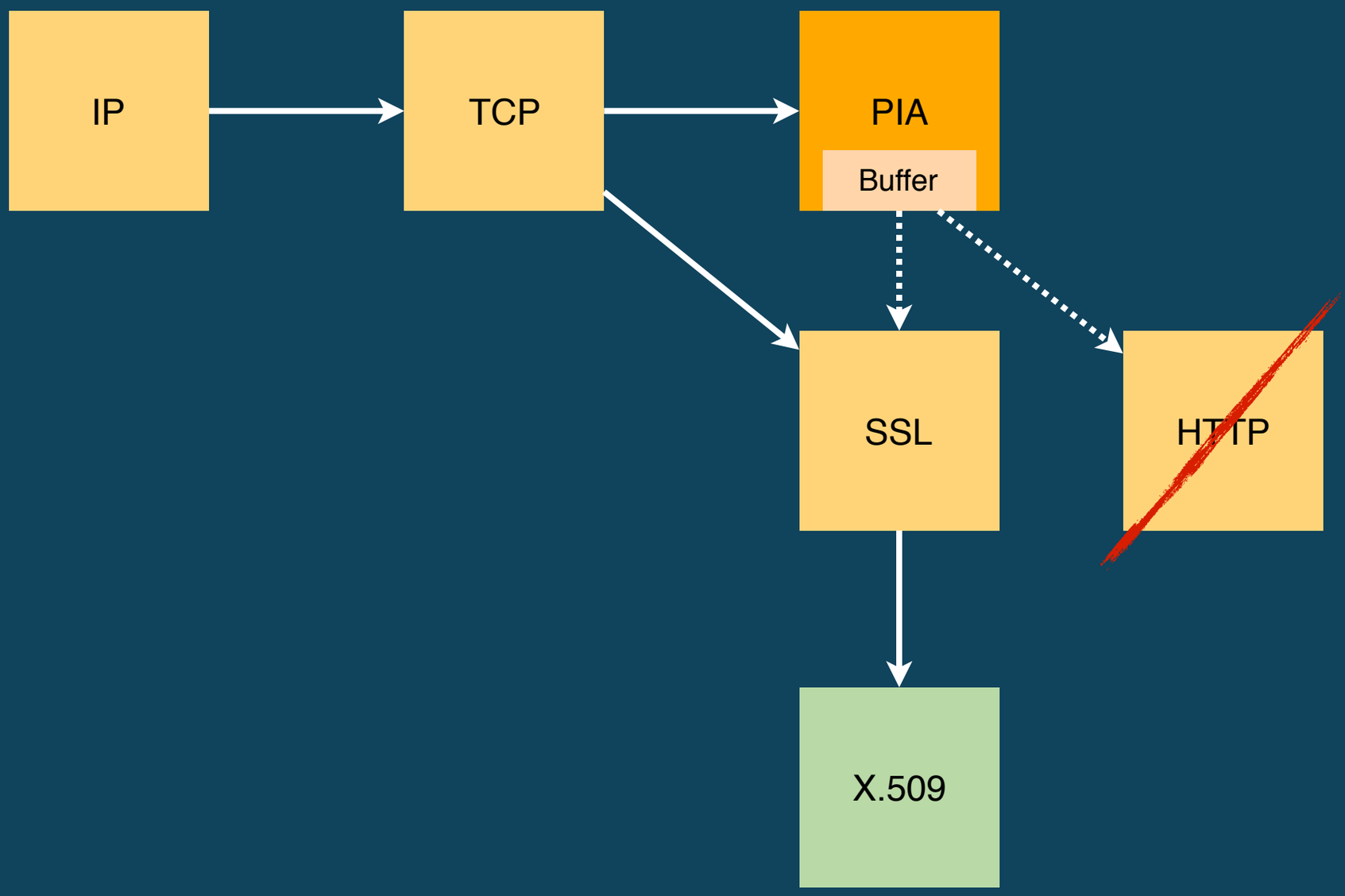
```
Analyzer::register_for_port(Analyzer::SSL, 443/tcp);
```



Dynamic Protocol Detection

```
signature dpd_ssl_server {
  ip-proto == tcp
  payload /^(\x16\x03[\x00\x01\x02\x03[…].*/
  tcp-state responder
  enable "ssl"
}
```

```
Analyzer::register_for_port(Analyzer::SSL, 443/tcp);
```



Protocol Analyzer API

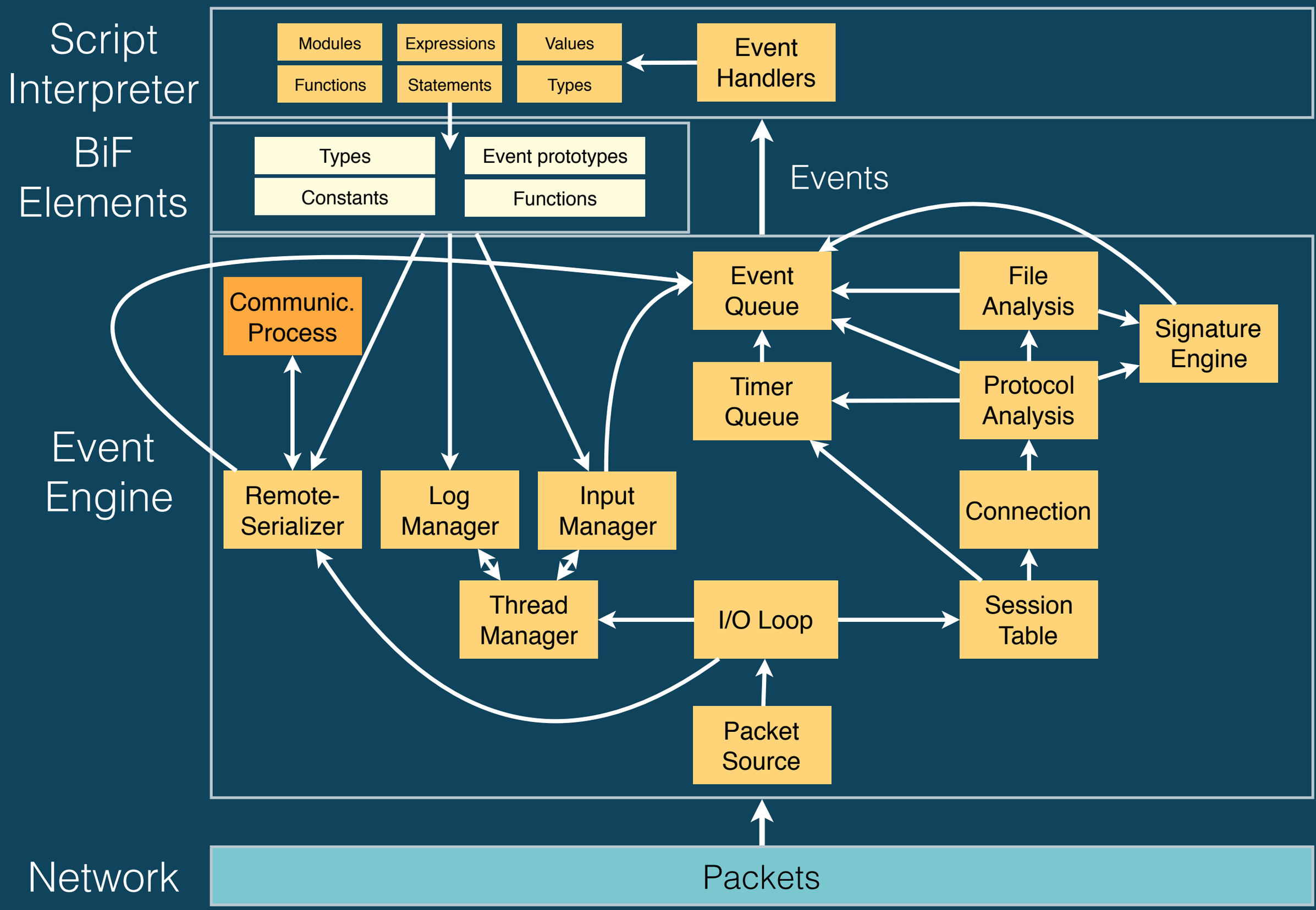
```
class Analyzer {
 virtual void Init();
 virtual void Done();
 virtual void DeliverPacket(int len, const u_char* data, bool orig,
                            bool orig, uint64 seq, const IP_Hdr* ip,
                            int caplen);
 virtual void DeliverStream(int len, const u_char* data, bool orig);
 virtual void Undelivered(uint64 seq, int len, bool orig);
 virtual void EndOfData(bool is_orig);
 virtual void FlipRoles();
 }
```

```
class TCP_ApplicationAnalyzer : public Analyzer {
 virtual void EndpointEOF(bool is_orig);
 virtual void ConnectionFinished(int half_finished);
 virtual void ConnectionReset();
 };
```

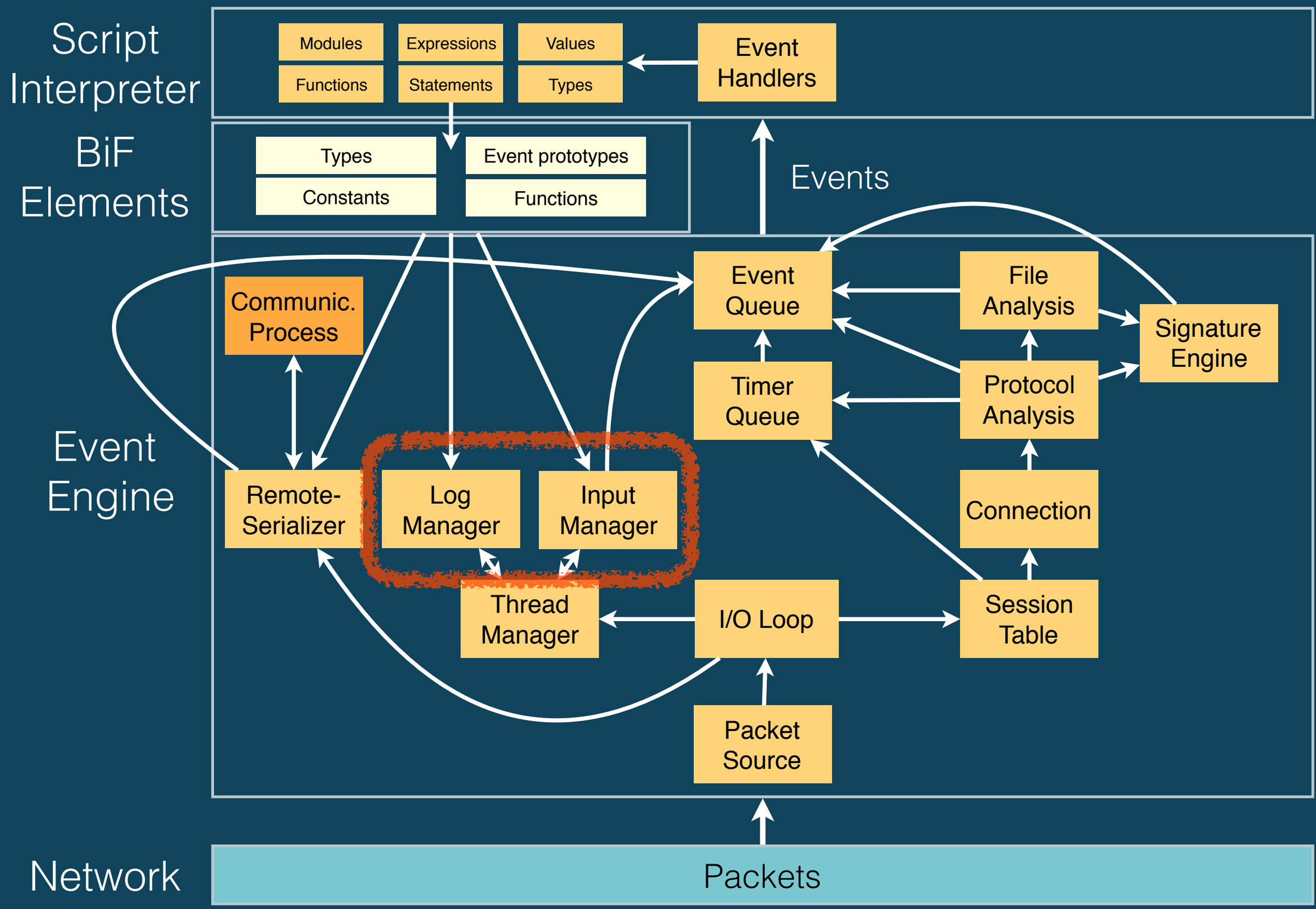
File Analyzer API

```
class Analyzer {
  virtual void Init();
  virtual void Done();
  virtual bool DeliverChunk(const u_char* data,
                            uint64 len, uint64 offset);
virtual bool DeliverStream(const u_char* data, uint64 len);
virtual bool EndOfFile();
virtual bool Undelivered(uint64 offset, uint64 len);
};
```

Bro Architecture



Bro Architecture



Writers & Readers

Log Writers

ASCII

SQLite

Input Readers

ASCII

Binary

Raw file

SQLite

Log Writer API

```
class WriterBackend {
 virtual bool DoInit(const WriterInfo& info, int num_fields,
 virtual bool DoWrite(int num_fields, const Field* const* fields,
                      threading::Value** vals);
 virtual bool DoSetBuf(bool enabled);
 virtual bool DoFlush(double network_time);
 virtual bool DoRotate(const char* rotated_path, double open,
                       double close, bool terminating);
 virtual bool DoFinish(double network_time);
 virtual bool DoHeartbeat(double network_time, double current_time);
 };
```

Each writer runs in its own thread.

Input Reader API

```
class ReaderBackend {
 virtual bool DoInit(const ReaderInfo& info, int arg_num_fields,
                     const threading::Field* const* fields);
 virtual void DoClose();
 virtual bool DoUpdate();
 virtual bool DoHeartbeat(double network_time, double current_time);

 // Simple mode.
void SendEvent(const char* name, const int num_vals,
               threading::Value* *vals);
void Put(threading::Value** val);
void Delete(threading::Value** val);
void Clear();
void EndOfData();

// Tracking mode.
void SendEntry(threading::Value** vals);
void EndCurrentSend();
};
```

Each reader runs in its own thread.

Bro Plugins

Build & install Bro components independently

Distribute as a Bro package

Log writers
Input readers
Protocol analyzers

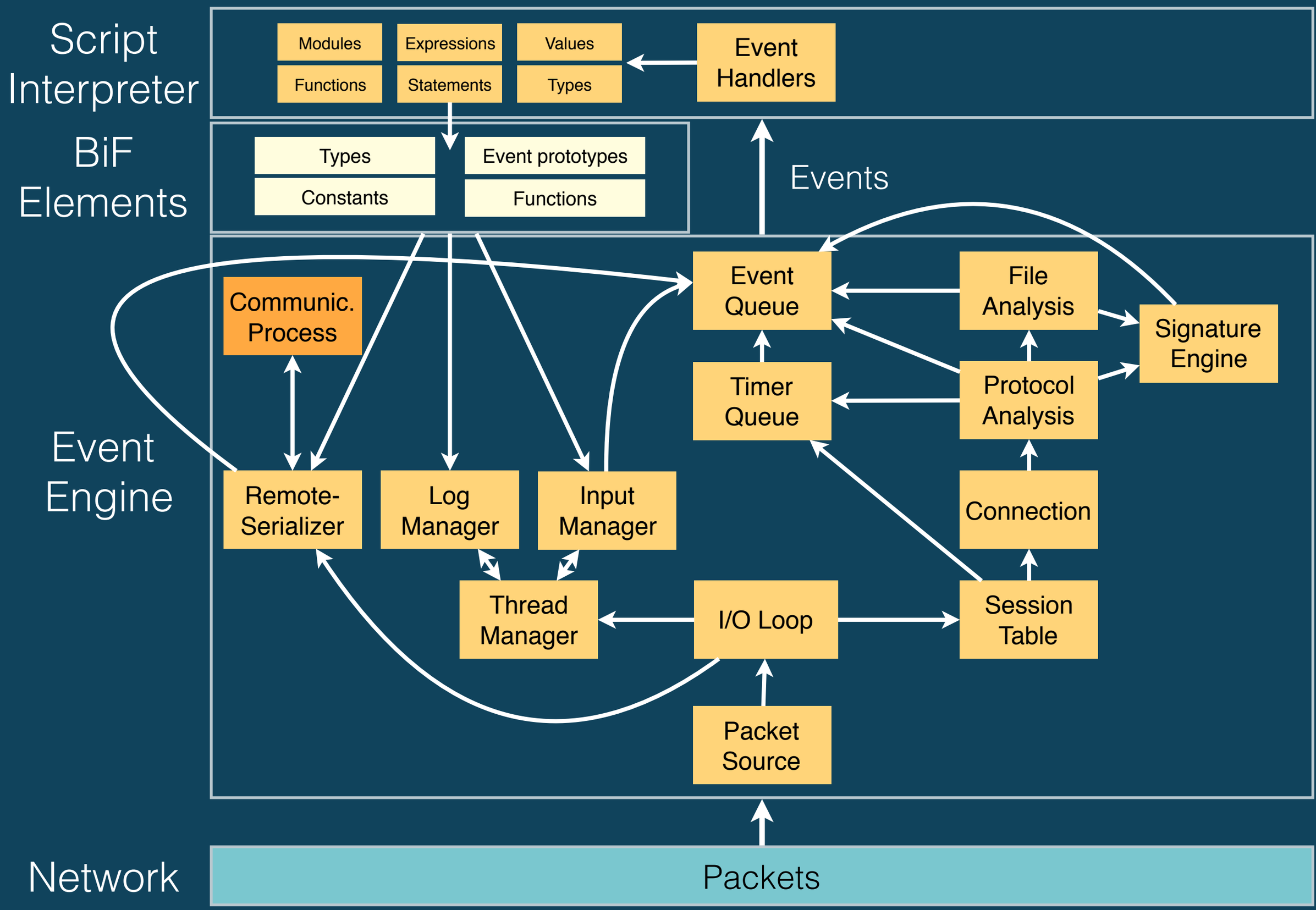
File analyzers
Packet Sources
BiF elements

Bro Scripts

BYOP

```
# ~/bro/aux/bro-aux/plugin-support/init-plugin icsi-plugin ICSI BroMagic
Installing icsi-plugin/CHANGES ...
Installing icsi-plugin/CMakeLists.txt ...
Installing icsi-plugin/configure ...
Installing icsi-plugin/configure.plugin ...
Installing icsi-plugin/scripts/__load__.bro ...
Installing icsi-plugin/scripts/ICSI/BroMagic/__load__.bro ...
Installing icsi-plugin/scripts/init.bro ...
Installing icsi-plugin/src/bromagic.bif ...
Installing icsi-plugin/src/Plugin.h ...
Installing icsi-plugin/src/Plugin.cc …
[…]
# cd icsi-plugin/
# ./configure --brodist=$HOME/bro/master
Build Directory         : build
Bro Source Directory    : /home/robin/bro/master
[…]
# make && make install
[…]
# bro -N
ICSI::BroMagic - <Insert description> (dynamic, version 0.1)
Bro::ARP - ARP Parsing (built-in)
Bro::AsciiReader - ASCII input reader (built-in)
Bro::AsciiWriter - ASCII log writer (built-in)
Bro::AYIYA - AYIYA Analyzer (built-in)
[…]
```

Bro Architecture



Script Interpreter

Events

Event Engine

Questions?

Network

Packets