University of Glasgow

ScotGrid

OSSEC and Elasticsearch

David Crooks
for Scotgrid Glasgow

HEPSYSMAN June 2015, RAL, 2nd June 2015

GridPP
UK Computing for Particle Physics

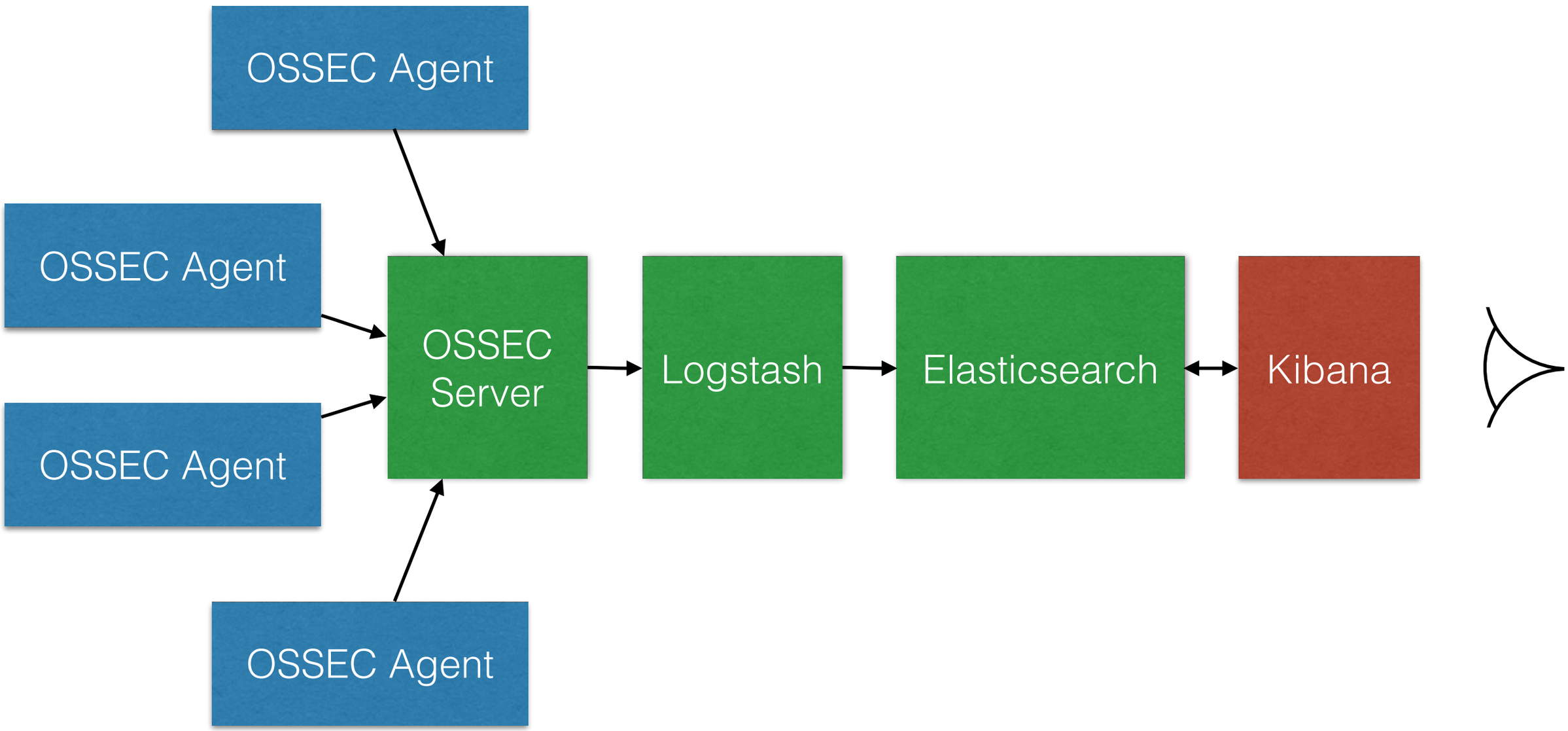
OSSEC

- Host Intrusion Detection System
- Written by Daniel Cid
- About 10 years old
- http://www.ossec.net

OSSEC & Elasticsearch

- Use the ELK stack to visualise the data aggregated by OSSEC.
- Useful blog: http://vichargrave.com
- http://vichargrave.com/ossec-log-management-with-elasticsearch/
- http://vichargrave.com/improved-ossec-log-parsing-with-logstash/

Data flow


OSSEC Agent
OSSEC Agent
OSSEC Agent
OSSEC Agent
OSSEC Server
Logstash
Elasticsearch
Kibana

logstash-syslog.conf

OSSEC can output a data stream in syslog format

```
input {
    #stdin{}
    udp {
        port => LISTENINGPORT
        type => "syslog"
    }
}

filter {
    if [type] == "syslog" {
        grok {
            match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp} %{SYSLOGHOST:syslog_host} %
{DATA:syslog_program}: Alert Level: %{NONNEGINT:Alert_Level}; Rule: %{NONNEGINT:Rule} - %{DATA:Description}; Location: %
{DATA:Location}; (user: %{USER:User};%{SPACE})?(srcip: %{IP:Src_IP};%{SPACE})?(user: %{USER:User};%{SPACE})?(dstip: %
{IP:Dst_IP};%{SPACE})?(src_port: %{NONNEGINT:Src_Port};%{SPACE})?(dst_port: %{NONNEGINT:Dst_Port};%{SPACE})?%
{GREEDYDATA:Details}" }
            add_field => [ "ossec_server", "%{host}" ]
        }
        mutate {
            remove_field => [ "message","syslog_timestamp", "syslog_program", "syslog_host", "syslog_message",
"syslog_pid", "@version", "type", "host" ]
        }
    }
}

output {
      #   stdout {
      #     codec => rubydebug
      #   }
    elasticsearch_http {
        host => "ESHOST"
    }
}
```

logstash-syslog.conf

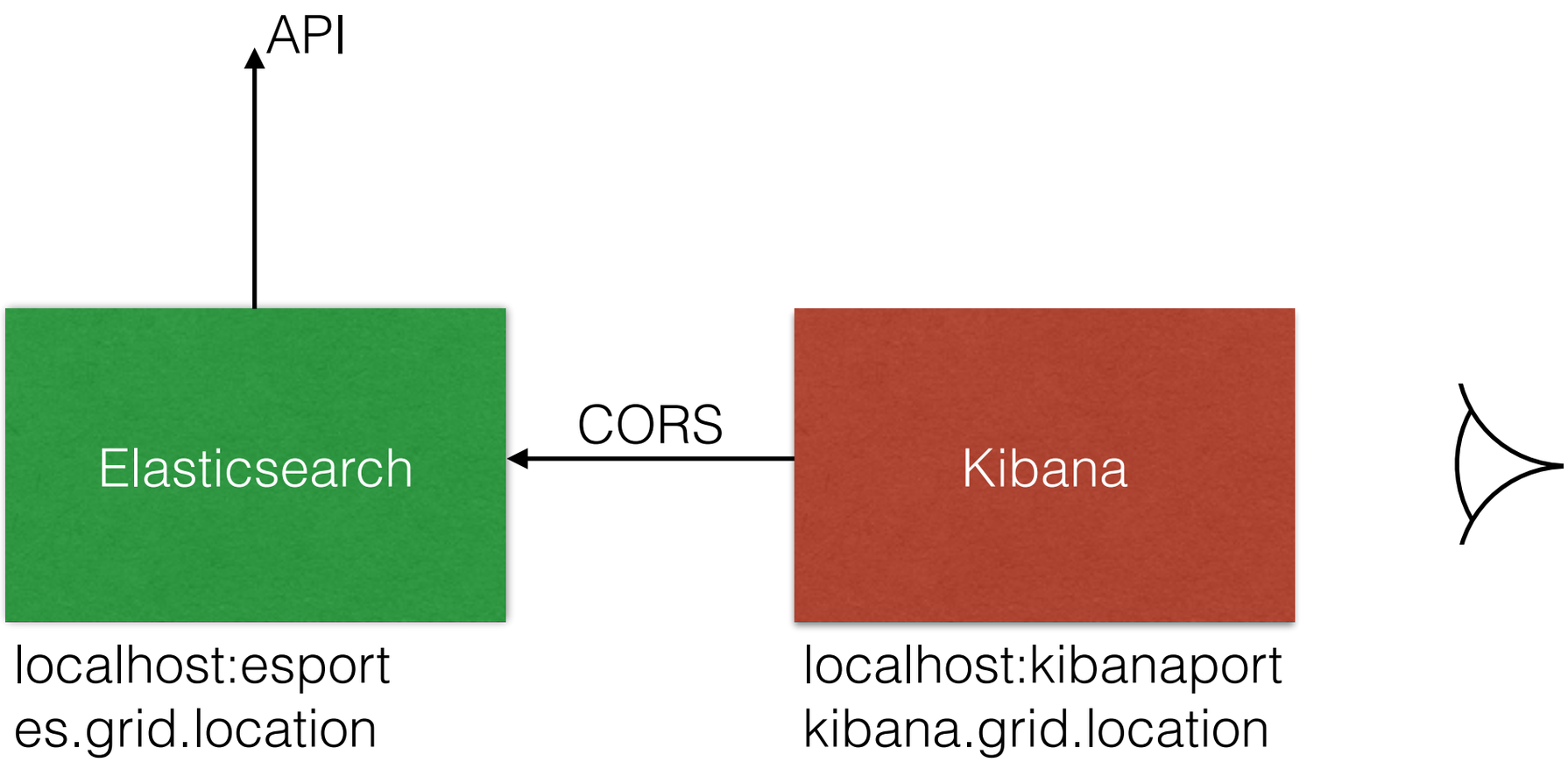
OSSEC can output a data stream in syslog format

```
input {
    #stdin{}
    udp {
        port => LISTENINGPORT
        type => "syslog"
    }
}

filter {
    if [type] == "syslog" {
        grok {
            match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp} %{SYSLOGHOST:syslog_host} %
{DATA:syslog_program}: Alert Level: %{NONNEGINT:Alert_Level}; Rule: %{NONNEGINT:Rule} - %{DATA:Description}; Location: %
{DATA:Location}; (user: %{USER:User};%{SPACE})?(srcip: %{IP:Src_IP};%{SPACE})?(user: %{USER:User};%{SPACE})?(dstip: %
{IP:Dst_IP};%{SPACE})?(src_port: %{NONNEGINT:Src_Port};%{SPACE})?(dst_port: %{NONNEGINT:Dst_Port};%{SPACE})?%
{GREEDYDATA:Details}" }
            add_field => [ "ossec_server", "%{host}" ]
        }
        mutate {
            remove_field => [ "message","syslog_timestamp", "syslog_program", "syslog_host", "syslog_message",
"syslog_pid", "@version", "type", "host" ]
        }
    }
}

output {
      #   stdout {
      #     codec => rubydebug
      #   }
    elasticsearch_http {
        host => "ESHOST"
    }
}
```

Securing Elasticsearch

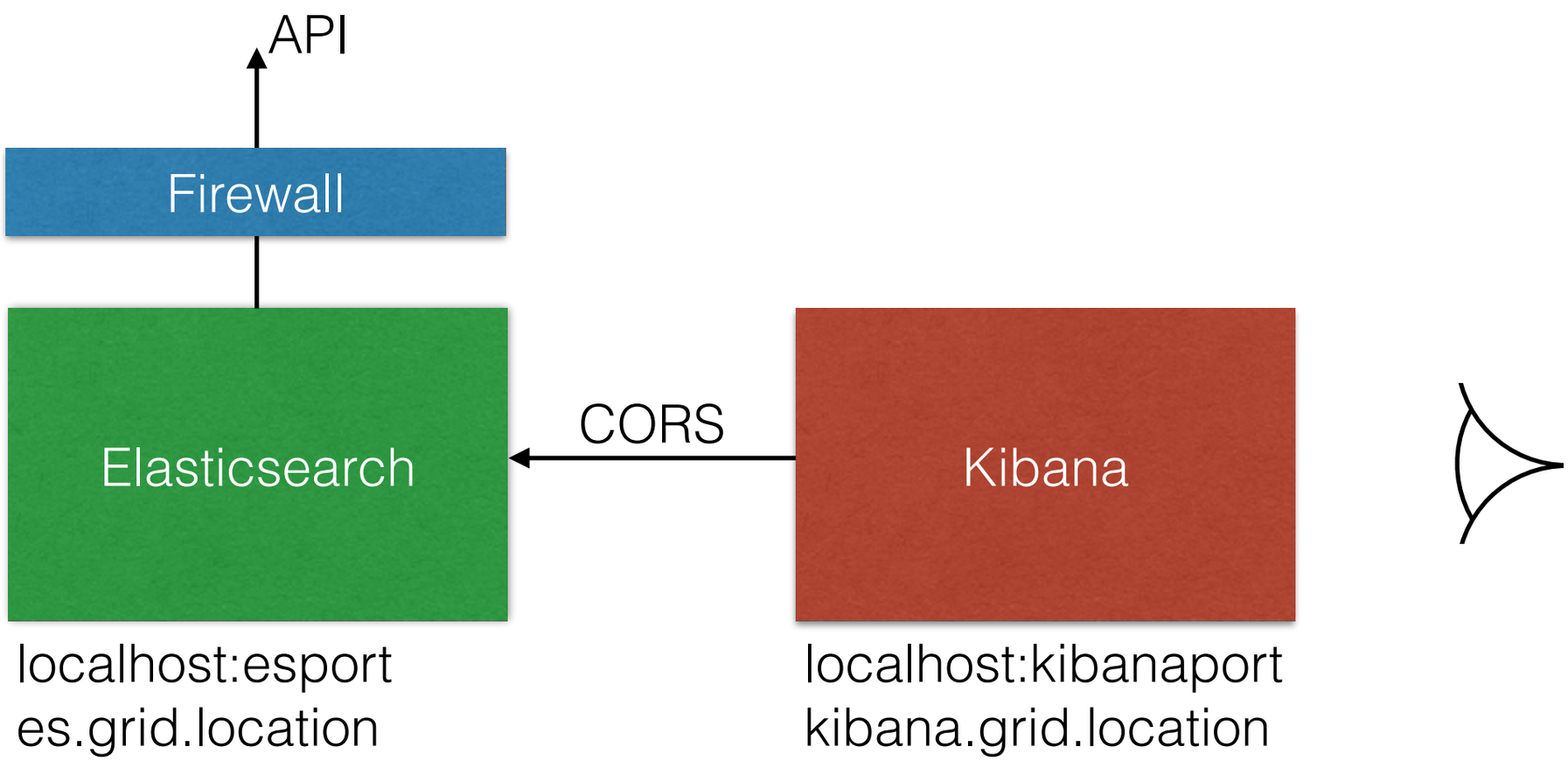
- Elasticsearch provides a very flexible interface which can lead to exposure of data
- Does not provide security layer as part of initial product
- Take appropriate steps to secure installation
- (Elastic has released a subscription-based security product, “shield”: https://www.elastic.co/products/shield)

Securing Elasticsearch



CORS:Cross Origin Resource Sharing

Securing Elasticsearch



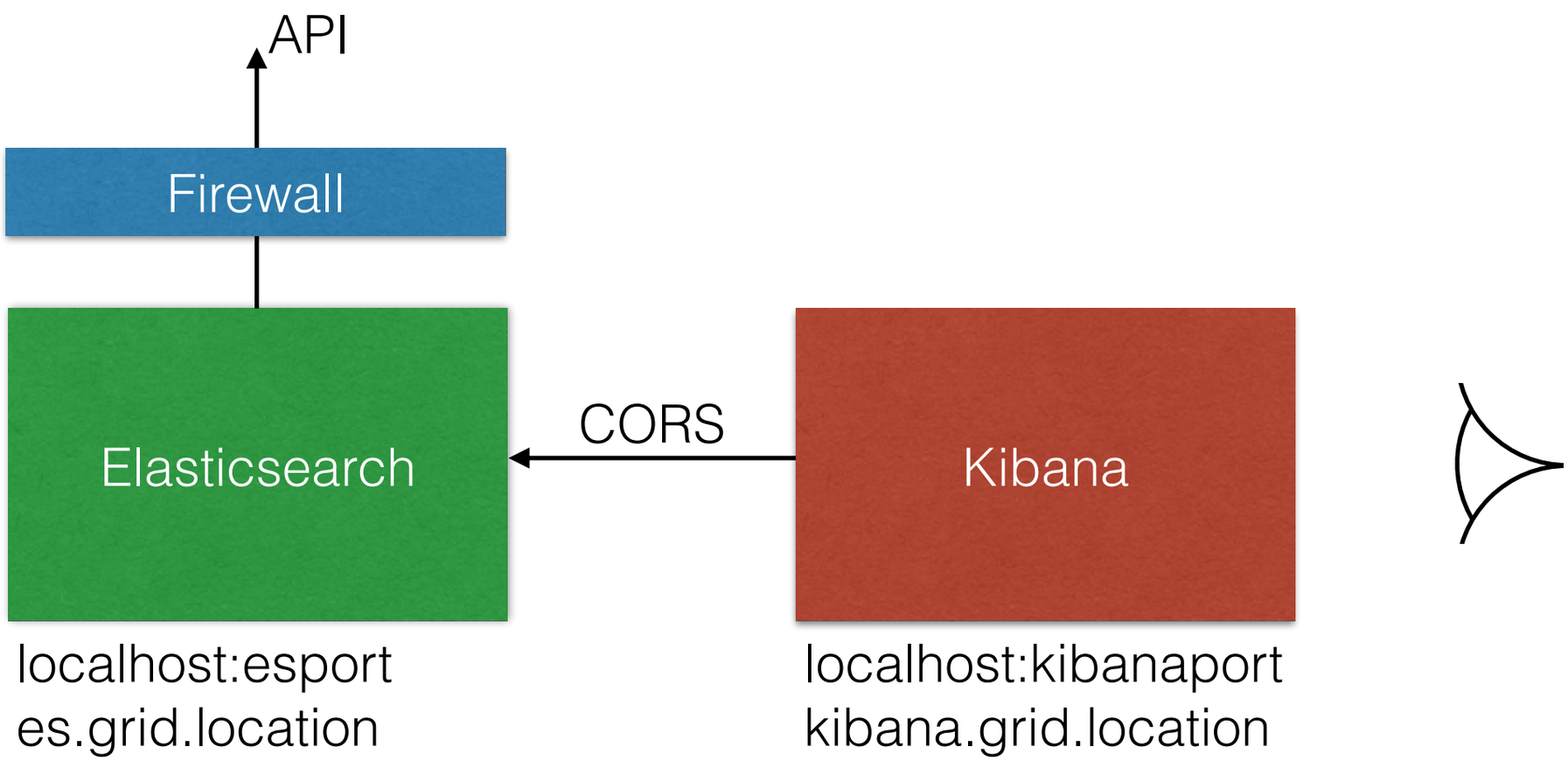
Secure Elasticsearch service from external access: Firewall

CORS:Cross Origin Resource Sharing

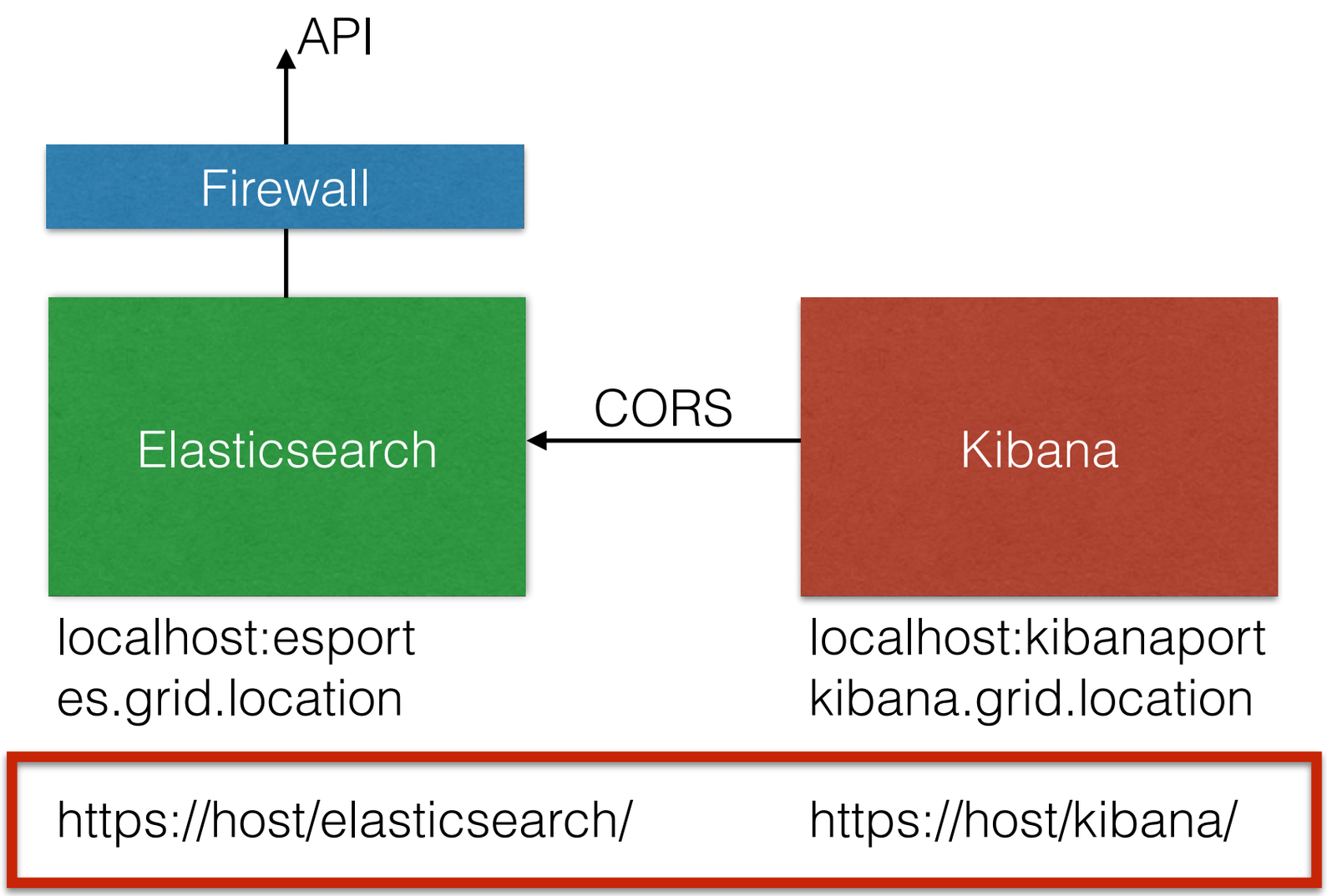
Securing Elasticsearch

- Secure access to Elasticsearch/Kibana
- Use https/authentication by grid cert
  - SSL configuration
  - Reverse proxy for elasticsearch
- Limit by IP range if desired

Securing Elasticsearch



Securing Elasticsearch



Grid additions

- One of the most useful features of OSSEC on installation is its wide range of initial rules
- Grid has its own set of logs and flags - we have followed process of picking out “unknown problems” and flagging them with a downstream rule.

Example

- One test available is to look for possible rootkit events through hidden files
- Interpretation important
- One node, for example, which showed large set of these triggers (node showed problems elsewhere which caused further investigation)
- Using Kibana it was straightforward to
  - isolate that node
  - pick out the rootcheck warnings
  - *remove* the rootcheck warnings
  - observe that in fact the node was throwing SATA errors - not a rootkit, but a bad hard disk

Example

48h view, whole cluster

~4h view, whole cluster

Example

~4h view, specific node

~24h view, specific node

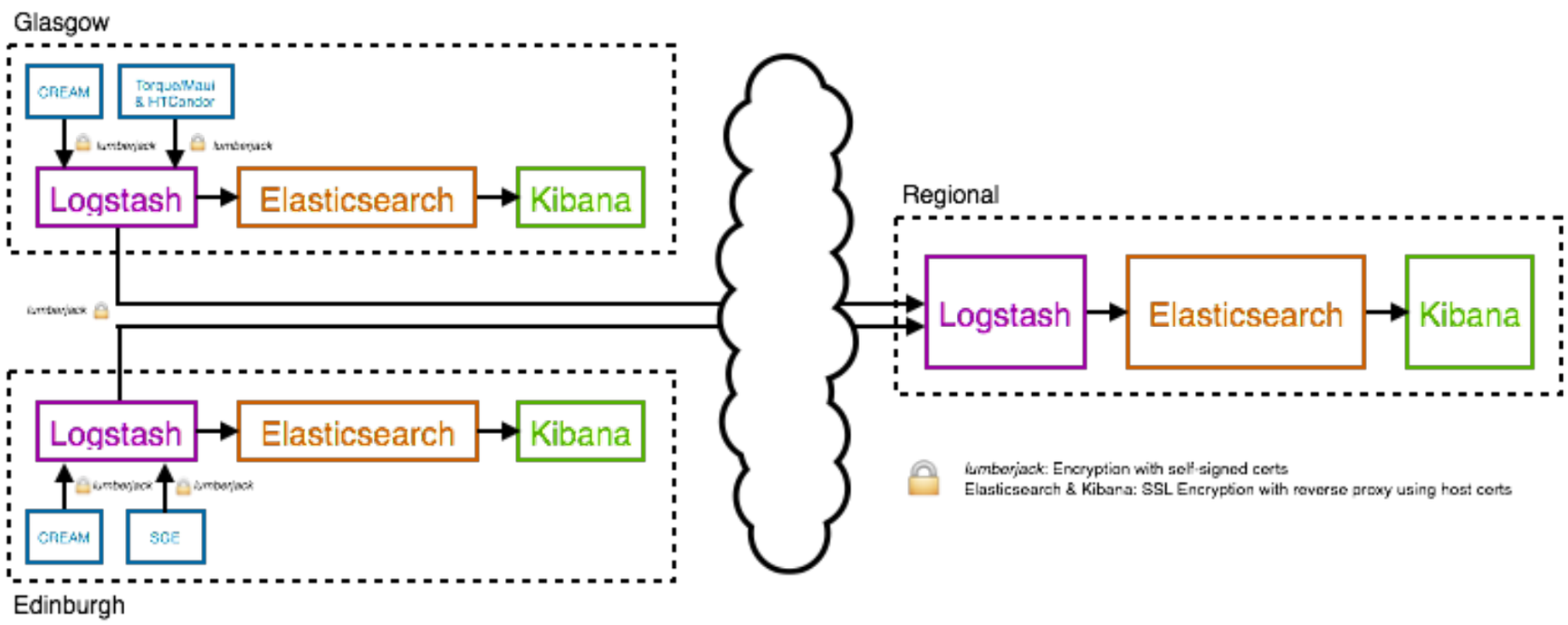
~10min view, specific node

Multisite Analytics

- CHEP work with Andrew Washbrook at ECDF
- Question: Analytics at a Distributed Tier-2?
- Look at process
  - Transport
  - Security considerations

Process

- Use site logstash instances which forward data to regional instance as well as ingesting into site elasticsearch



- Additional information injected at site logstash layer to identify site information at a regional level

Security

- Security, both in terms of encrypted data flow and data protection, is an important consideration
  - Used the same firewall, https and reverse proxy techniques as in OSSEC instance
  - Potentially sensitive user and location data were removed prior to transport off the site domain by stripping or anonymising identified strings
  - Used encrypted data streams with the lumberjack protocol, using a standard TLS infrastructure using self-signed host certificates

Summary

- Two applications for an ELK stack (Andy will talk more next about the multisite example)
- Next steps are to look at scaling