Splunk Connect for Syslog: Extending the Platform

Easily onboard *all* of your syslog data!

Mark Bonsack
Ryan Faircloth

Principal Architect | Splunk
Principal Product Manager | Splunk

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or plans of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results may differ materially. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only, and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionalities described or to include any such feature or functionality in a future release.

Mark Bonsack
Ryan Faircloth

Principal Architect | Splunk
Principal Product Manager | Splunk

splunk> .conf20

You Are In the Right Place If:

You want to customize Splunk Connect for Syslog for supported sources

- Configure “Unique Ports” for devices sending on non-standard ports (env_var files)
- Configure Hostname and CIDR block (context-driven) filters

You want to expand the platform for custom data sources

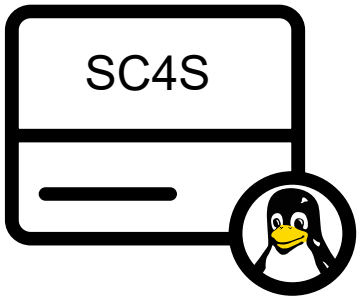
- Add new “log paths” and “filters” for data sources not covered Out of the Box
- Learn Syslog-ng parsing basics

You have inherited an existing (broken?) syslog implementation

- And want to ”back-port” custom syslog-ng or rsyslog filters into an SC4S deployment

And you have:

- Linux admin skills
- A passing familiarity with syslog-ng configuration
- Rudimentary knowledge of templating (we will cover this)



Agenda

1) **SC4S Platform Overview**
   Packaging and Deployment

2) **Simple Customizations**
   Environment Variables and Templating

3) **Context-driven Customizations**
   Context-driven Filters and Splunk Metadata Overrides

4) **Filters and Log Paths**
   Add a new device to SC4S

5) **Hands-on Keyboard!**
   Let's add a new device to SC4S!

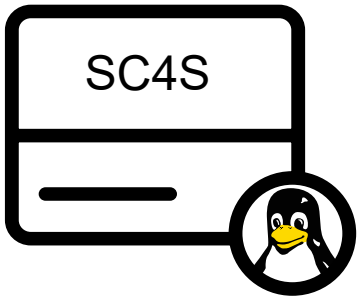
6) **Architectural Considerations**
   Wrap-up/Takeaways

The SC4S Platform: Managed syslog-ng

Thoughtful design yields a consistent, repeatable, and scalable experience

Packaging

- Container allows super-simple runtime and administration
- All dependencies are taken care of
- Allows Splunk to guarantee the experience independent of underlying distro, versions, etc.



Templating

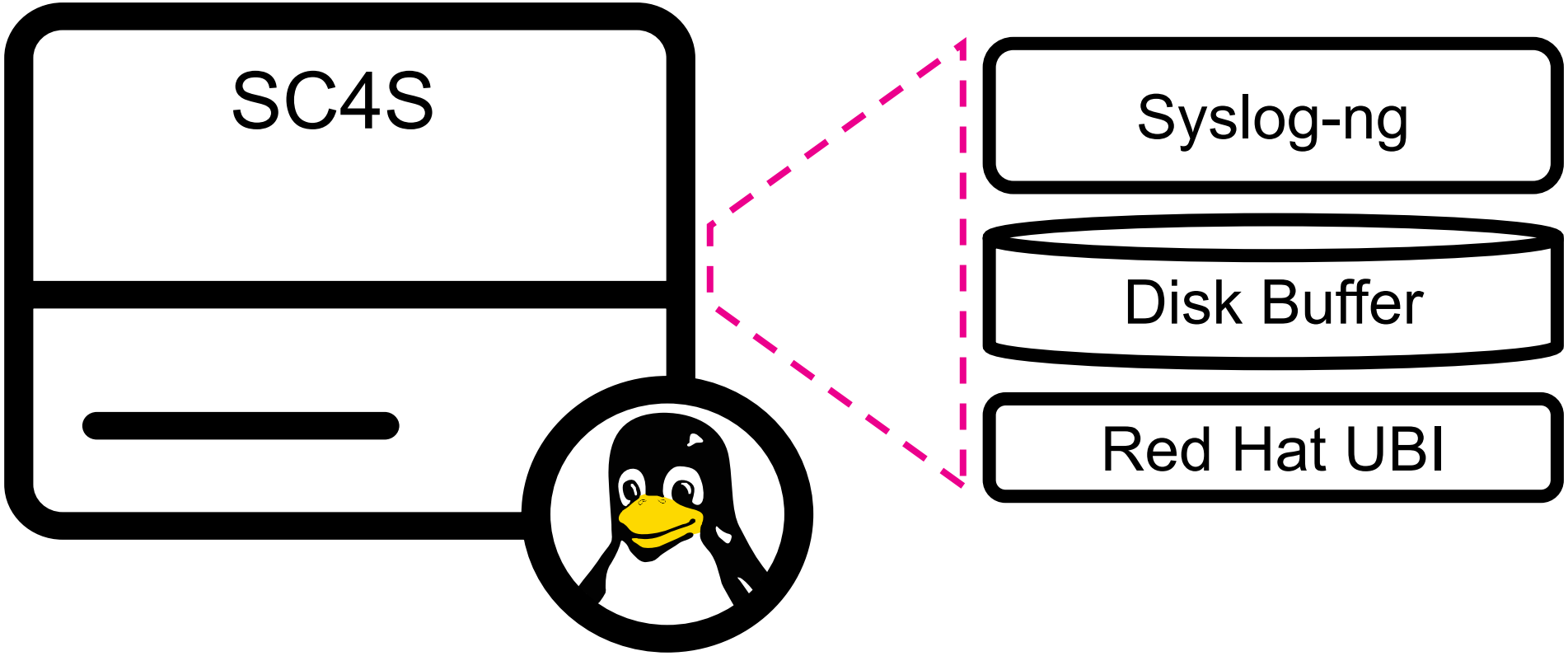
- Use of “go templates” (gomplate) to create syslog-ng config at runtime
- Fills a significant gap in the syslog-ng configuration “language”
- Allows underlying syslog-ng config to be abstracted; ”programming” hidden

Log Path Processing/Splunk Metadata Assignment

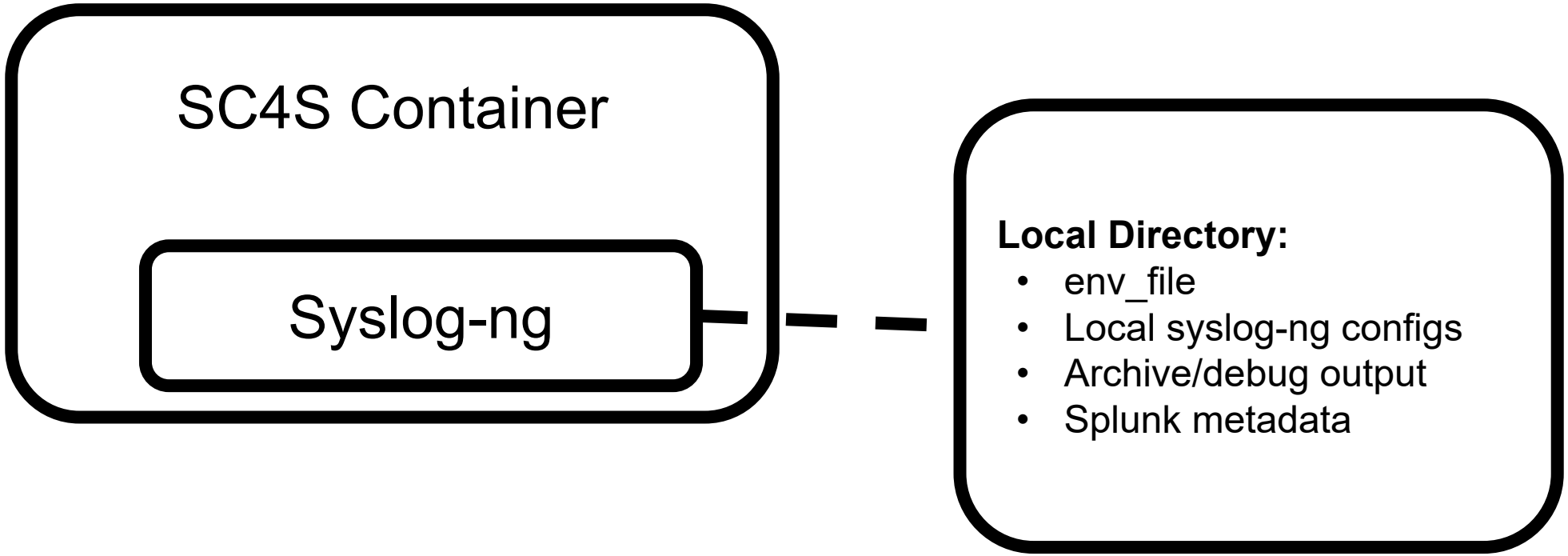
- The guts of the “magic”
- Deep understanding the event contents and crafting the proper Splunk metadata
- Handed to Splunk on a “silver platter”; no props/transforms needed at ingest.

Turnkey Packaging: The SC4S Container

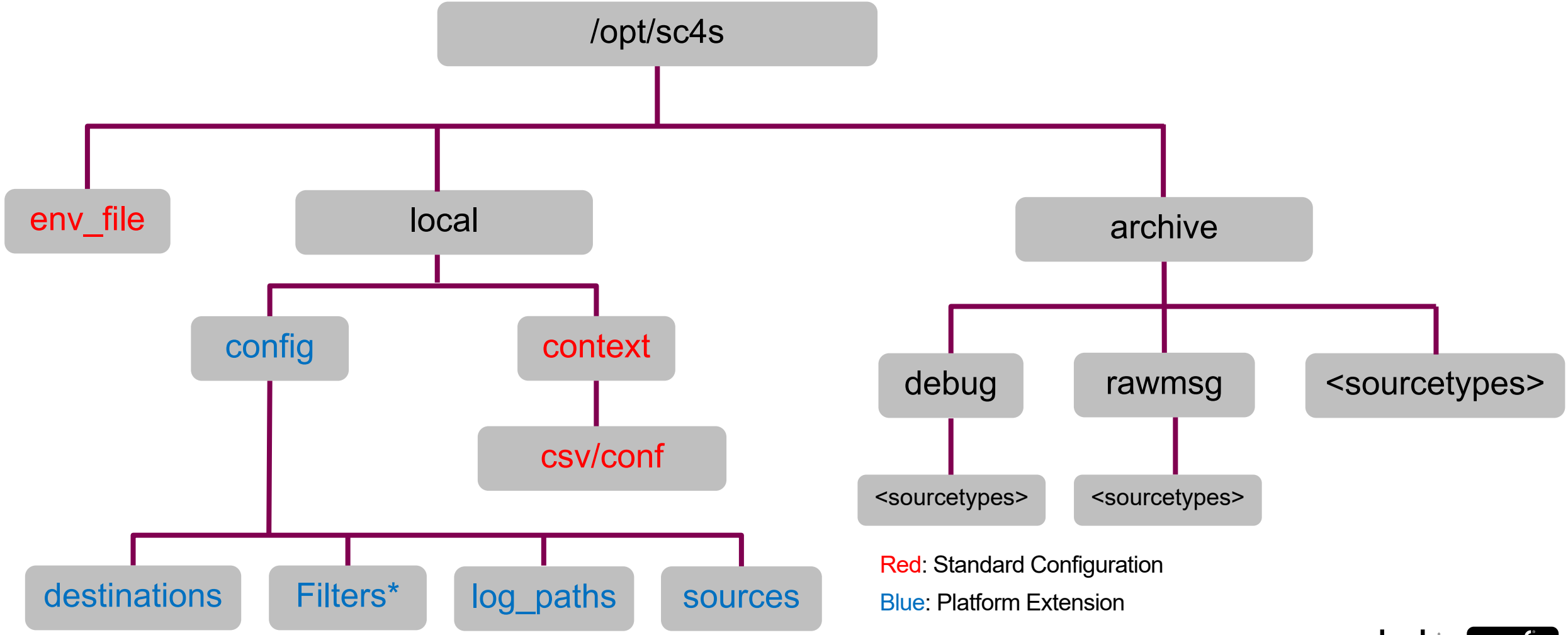
Turn-key syslog appliance



SC4S: Configuration



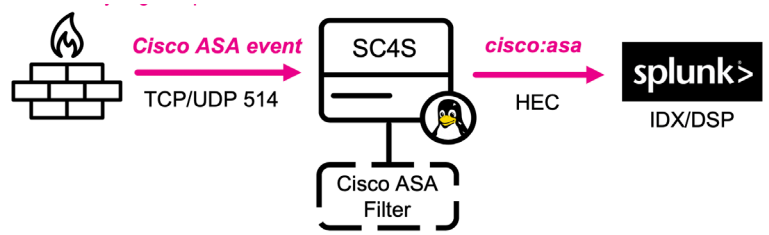
SC4S Configuration: Local Directory



* Filter directory is optional; recent update consolidates filters directly into log paths

SC4S: The Magic

Message Processing/Splunk Metadata Assignment



All syslog-ng config files follow the same basic scheme

Global Options	High-level configuration, include files
Log Paths (Source, Dest, Filter)	Parsing, enhancement, Spunk metadata
Sources	Network (by far most common)
Filters	Match incoming events
Message Parsing	Timestamp and Metadata
Destinations	Send to Splunk via HEC by default

SC4S: env_file

SC4S high-level configuration

Splunk URL/TOKEN

```
SPLUNK_HEC_URL=https://splunk.foo.com:8088/services/collector/event
SPLUNK_HEC_TOKEN=b123456a-dcfe-1234-abcd-a03184455e76
```

Unique Listening (Source) Ports

```
SC4S_LISTEN_JUNIPER_NETSCREEN_TCP_PORT=5000
SC4S_LISTEN_CISCO_ASA_TCP_PORT=5001
```

Debug and Alternate Destination Switches

```
SC4S_SOURCE_STORE_RAWMSG=yes
SC4S_DEST_GLOBAL_ALTERNATES=d_hec_debug,d_archive
```

Kernel Parameters

```
SC4S_SOURCE_UDP_SO_RCVBUFF=33554432
SC4S_SOURCE_LISTEN_UDP_SOCKETS=32
```

SC4S: Templating

Translate environment variables to running config

Templating process (gomplate) runs at container startup

Translate environment variables:

```
SC4S_SOURCE_UDP_SO_RCVBUFF=33554432
```

From this (in the template):

```
so-rcvbuf({{getenv "SC4S_SOURCE_UDP_SO_RCVBUFF" "1703936"}})
```

To this (in the final config):

```
so-rcvbuf(33554432)
```

Full conditionals and other programming constructs:

```
   {{- if or (conv.ToBool (getenv "SC4S_ARCHIVE_GLOBAL" "no")) (conv.ToBool
(getenv "SC4S_ARCHIVE_CISCO_ASA" "no")) }}
    destination(d_archive);
   {{- end}}
```

SC4S: Context Files

The key to metadata assignment for all events

Splunk Metadata

```
/opt/sc4s/local/context/splunk_metadata.csv
```

Vendor/Product Context (Filters)

```
/opt/sc4s/local/context/vendor_product_by_source.conf
/opt/sc4s/local/context/vendor_product_by_source.csv
```

Compliance Overrides (Sub-filters)

```
/opt/sc4s/local/context/compliance_meta_by_source.conf
/opt/sc4s/local/context/compliance_meta_by_source.csv
```

Reverse DNS (Fix broken hostnames)

```
/opt/sc4s/local/context/host.csv
```

SC4S: Log Paths – Parsing and Metadata

Creating the Magic

Source

- Network: default “soup” port – 514 or unique port(s) set by environment variable(s)
- Any locally-configured file or system source
- Acts as a filter

Log Filter

- Primary parser to determine major sourcetype
- Can utilize different criteria – source IP/hostname, regex in the message or header

Log Path Processing/Splunk Metadata Assignment

- The guts of the “magic”
- Deep understanding the event contents and crafting the proper Splunk metadata
- Data Handed to Splunk on a “silver platter”; no props/transforms needed at ingest.

Destination

- HEC is the primary destination
- Alternates can be supplied on a per Log Path (sourcetype) basis
- Alternates can include HEC destinations, local file, and external network destinations

Architectural Considerations

The syslog protocol is *old!*

Keep it Simple

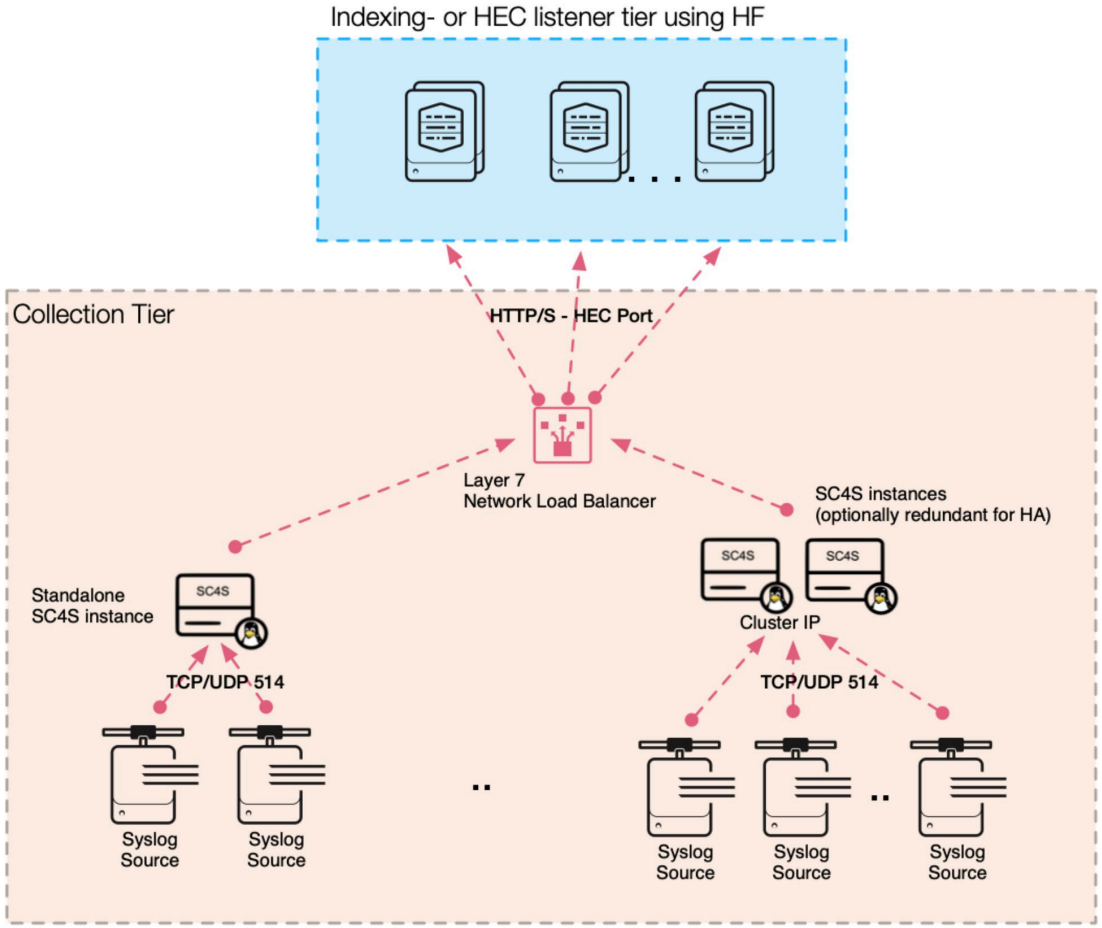
- Syslog can be a “religion”, best approach is to not over-engineer

Distribute collection to the management zones

- Account for limitations of the the syslog protocol
- Minimize scale issues
- Minimize “blast radius” of failures

Don't over optimize for HA

- Syslog is a lossy protocol (think MP3)
- Realize that syslog is, at best, “Mostly Available”



Community Contribution and Resources

Links to key resources

Splunkbase:

- Overview and links to other resources

SC4S Blog Series:

- Blog series covering the lifecycle of SC4S including Extending the Platform (Part 3)

Performant AND Reliable Syslog UDP is best:

- Real-world myth-busting truths about the syslog protocol and architectural best practices

Github Repo:

- Your home for issue tracking, PRs, and other open-source goodness

Documentation:

- Read this! It will eliminate 90% of deployment/configuration issues!

Key Takeaways

Extending the SC4S Platform

Simple: Environment Variables

Context-Driven: Filters and Metadata

Platform Extension: Log Paths for New Devices!

Thank You

Please provide feedback via the

SESSION SURVEY