Wrangling Your IOT Data Into Splunk

Damien Dallimore

IOT Dreamcatcher , Splunk

.conf2016

splunk>

Disclaimer

During the course of this presentation, we may make forward looking statements regarding future events or the expected performance 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 could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC. 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, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. 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 functionality described or to include any such feature or functionality in a future release.

It all starts with **Getting The Data In**

Getting the data in should not be hard

And you should be able to transform the data for the destination

So I’ve tried to ease the pain

KINESIS	Amazon Kinesis Modular Input
AMQP	AMQP Messaging Modular Input
MERAKI	Cisco Meraki Presence Modular Input
COAP	COAP Modular Input
CMD	Command Modular Input
LOG	Java Logging Appenders
JMS	JMS Messaging Modular Input
AGENT	JVM Instrumentation Agent
KAFKA	Kafka Messaging Modular Input
JMX	Monitoring of Java Virtual Machines with JMX
MQTT	MQTT Modular Input
PDI	Protocol Data Inputs
PubNub	Pubnub Modular Alert
PubNub	Pubnub Modular Input
REST	REST API Modular Input
SEND file	Scheduled Export of Indexed Data (SEND) to File
SNMP	SNMP Modular Input
TESLA	Tesla Vehicle Modular Input
	Twilio SMS Alerting

Common design approach

Simple and intuitive to install and configure

Multi platform support

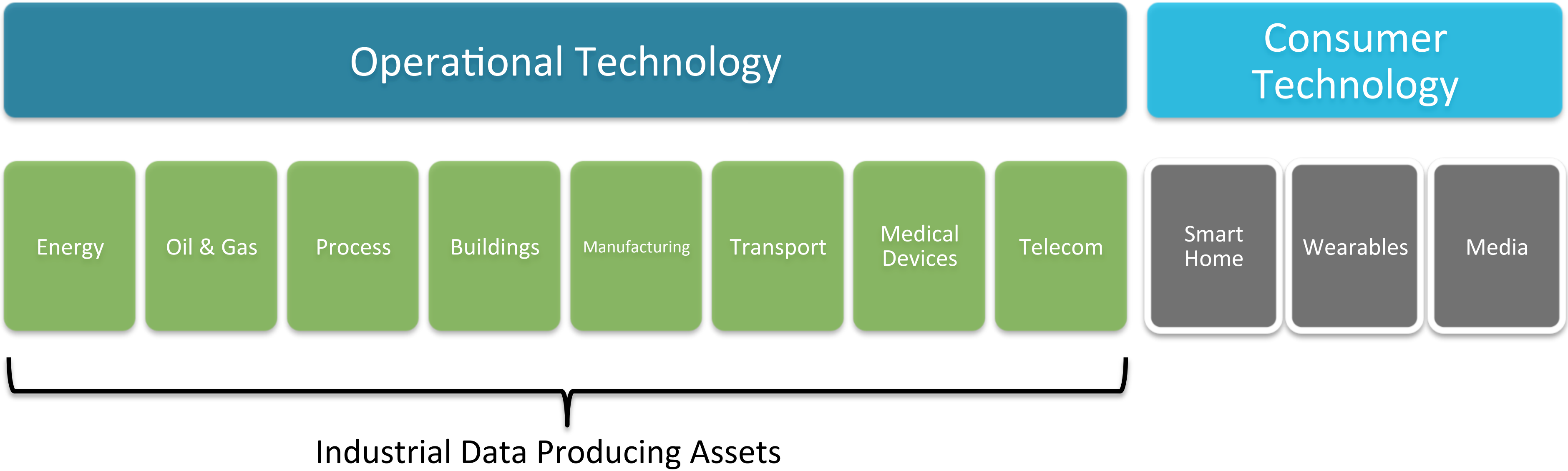
Extensible via plugging in your own own custom handlers to pre process received data before indexing

Can scale vertically and horizontally

Open and community supported/collaborated

What is this IOT data, is it just these things ?

The IOT data landscape is much, much vaster



Operational Technology

Consumer Technology

Energy

Oil & Gas

Process

Buildings

Manufacturing

Transport

Medical Devices

Telecom

Smart Home

Wearables

Media

Industrial Data Producing Assets

Let’s wrangle this data into Splunk

IOT Data accessible via IOT Protocols

CoAP (Constrained Application Protocol)

MQTT (formerly MQ Telemetry Transport)

HTTP APIs (RESTful or otherwise)

IOT Data accessible via Messaging APIs

AMQP (Advanced Message Queuing Protocol)

JMS (Java Message Service)

Apache Kafka

Amazon Kinesis

IOT Data accessible via “Other Means”

Command Modular Input (capture output from commands)

Cisco Meraki Modular Input (wireless access points & devices)

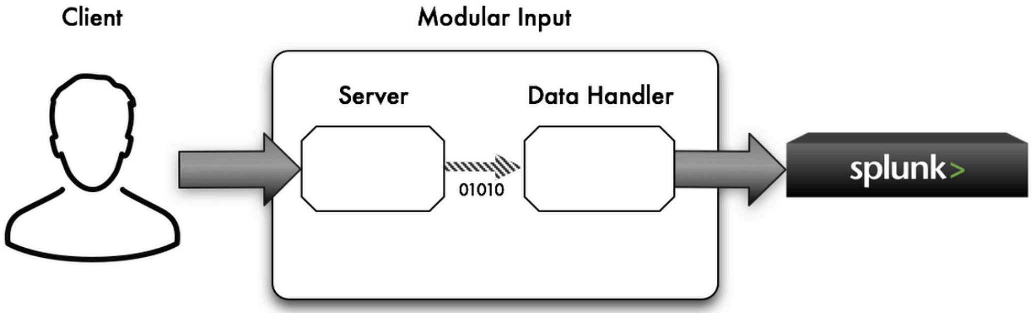
Splunk Stream (capture packets off the wire)
Splunk HTTP Event Collector (push events to Splunk from code, logging libraries , AWS Lambdas)
Integrations with 3rd party aggregators/integrators/gateways (Kepware , Pubnub, Octoblu)

PDI Protocol Data Inputs

Augments Splunk’s native Data Input options(TCP/UDP/HEC) to receive **text or binary data** via many different protocols

Plugin your own custom data handlers to pre process data before it gets indexed in Splunk

Freely available Add-On hosted at Splunkbase



Key features

Receive any type of data , **text or binary**

Many protocols supported

- HTTP POST/PUT ,HTTP File upload, TCP, UDP, SockJS, Websockets, SSL/TLS support

Dynamically plugin your own custom data handlers that can do whatever you code them to do

Write data handlers in numerous different languages

- Java , Python, JavaScript, Groovy, Scala, PHP, Clojure, Ceylon, CoffeeScript

Designed for high scale

- large volumes of concurrent client connections
- high data volumes
- Optionally uses HEC (HTTP Event Collector) out the back end to send data to Splunk
- non blocking , asynchronous , event driven architecture internally
- scales over all your available CPU cores

Why use this App

Decode Binary data

- Proprietary protocols , compressed data, encrypted data, binary files etc..

Custom pre-processing / pre-computation of data

Integrate other data processing or CEP (complex event processing) frameworks

- Storm , Spark , Siddhi, Esper

Support large numbers of concurrent client connections

Support large scale data throughput

Tap into traditionally “harder to get at” data for Splunk

Want to dive deeper on anything ?

Lets Jam !

Talk to me now

Ad Hoc chalk talks

Tweet at me : @damiendallimore

Email me : ddallimore@splunk.com

Visit the IOT booths at anytime

Attend other IOT sessions

THANK YOU

.conf2016

splunk>