

# IoT at Scale

or, "From Little Things, Big Data Grow"

Christopher Biggs, *Accelerando Consulting*  
@unixbigot *accelerando\_au*

## Who am I?

Christopher Biggs – @unixbigot – @accelerando\_au

- ▷ Brisbane, Australia
- ▷ 20+ years in IT as developer, architect, manager
- ▷ Founder, *Accelerando Consulting*
- ▷ Accelerando is a "full service" consultancy - chips to cloud
- ▷ *IoT, DevOps, Big<sup>†</sup> Data*

<sup>†</sup> Actual bigness may vary

## From little Things, Big Data Grow

### "Internet of Things" (IoT) At Scale

- ▷ **Why Even IoT?** - my personal motivation
- ▷ **Chips to Cloud** - the journey from sensor to senses
- ▷ **The Air Gap** - connecting wireless devices to the Internet
- ▷ **Data Tributaries** - follow the data from droplet to ocean
- ▷ **Making sense of it all** - Managing and visualising IoT data

# Overture: Why Even IoT?

*"Don't ever **invite** a vampire into your house, you silly boy. It renders you powerless."*

-- "Max", *The Lost Boys*

*"The 'S' in IoT stands for 'Security'."*

-- Absolutely Everyone, *The Internet*

## What is IoT For?

It's for not this





People who shouldn't have to do crummy jobs:

Children

People who shouldn't have to do crummy jobs:

People of Colour

People who shouldn't have to do crummy jobs:

Women

People who shouldn't have to do crummy jobs:

Immigrants

People who shouldn't have to do crummy jobs:

Foreigners

People who shouldn't have to do crummy jobs:

Robots

## Wait, what?

"Aren't crummy jobs what robots are *for*?"

*"Do you want a war of extinction? Because that's how you get wars of extinction"*

--Malory Archer, nearly



## Artificial nonintelligence

Doing all the things **no** intelligent being should have to be bothered by.

Give me your tired-of-doing-that,  
your poorly done jobs,  
your harried masses yearning to be stress-free

-- Emma Lazarus, kind of

# IoT's Trinity

# Things

Start with the simplest parts...

# Internet

...connect them together...

Of (i.e. context)

...to form a synergistic whole.

## My Three Laws of IoT

- ▷ Devices must cooperate for the benefit of humans
- ▷ Devices must communicate, and obey instructions
- ▷ Devices must be as simple and reliable as possible



*OK, end of setup*

-- "Ford Fairlane", (not just a car, but a really awful movie)

# Communications

## Crossing the Air Gap

Fast, Cheap, Battery-efficient.

Choose any two.

## PoE (Power-over-Ethernet)

Fast, no-battery, expensive wires.

## EoP (Ethernet over Powerline)

Cheaper wiring, not so fast.

## PoDS (Plain ol' Dumb Serial)

Please, no.

# WiFi

(Wireless Made-up-Acronym)

# Bluetooth

(Don't chew on that Biro)



# ETPH

You know, telephones

# LoRa

Telephones from 1978, but without the wires.

# IPoSUV

## Comms Recap

- ▷ If you have wires, use 'em
- ▷ PoDS is not IoT
- ▷ If you have the power budget for WiFi, great
- ▷ How about a bluetooth mesh?
- ▷ "Low-power" equals "really bloody slow"

# Structuring for Scalability

Insert magic cloud here

Nope

## Edge Computing

aka "Not the cloud but we've spent so long plugging cloud we can't possibly admit that"



## Web API

HTTP is awful on slow links

HTTPs is aw-overflowing

## Message buses

long-lived connection, lightweight messages

# Retrofit

## Websockets

## Build your own

A bit of orchestration, a bit of docker, a bit of time

# Owning an architecture is a big responsibility

Maybe you should start with a puppy.

# Don't build your own

Oh, look, google, amazon and microsoft already did it

Well, they announced it

## Microsoft

Messaging in place, provisioning "coming soon"



# Google

Messaging in place, management in private beta

Device platform (Android Things) supports two vendors only.

## Amazon

Messaging, device management function-provisioning in place.

No news on device provisioning.

So, tell me again about rolling my own?

(I'll tell you on Thursday)

## Give me a hint?

Saltstack. Mosquitto. Docker.

## Leveraging saltstack

- ▷ ZeroMQ message bus transport
- ▷ Tree-structured hierarchy
- ▷ Top level master (or multiple)
- ▷ Intermediate "syndicate masters" (optional)
- ▷ Leaf-node "minions"

## Playing nice with MQTT

- ▷ Message broker federation (one- or two-way)
- ▷ Lots of implementations
- ▷ Mosquitto in Docker

## What do you mean you want it reliable?

- ▷ MQTT QoS options (but not end-to-end)
- ▷ Apache Kafka
- ▷ Amazon Kinesis

## Case study 1 - MQTT and Kinesis

- ▷ Sensors transmit to on-site MQTT broker
- ▷ Cloud MQTT broker federates on-site brokers
- ▷ Cloud ingester writes to Elasticsearch
- ▷ Cloud ingester forwards certain events to kinesis stream
- ▷ AWS Lambda functions handle kinesis event
- ▷ Problems: no recovery of missed messages



## Case study 2 - Saltstack and Elasticsearch (fleetvalid.com)

- ▷ Sensors connect to on-site saltstack gateway
- ▷ (Potential store-and-forward for disconnected sensor stations)
- ▷ Saltstack gateway connects to cloud 'master'
- ▷ Event 'engine' on master relays events to Elasticsearch data store

## Case study 3 - Meshes and AWS IoT (evacmate.com.au)

- ▷ Sensors create a mesh network over bluetooth
- ▷ One or more gateway nodes have 3G uplinks
- ▷ Data streamed to cloud over websocket
- ▷ Gateway nodes managed with AWS IoT

# Making sense of data

## Layers of data

- ▷ Device health
- ▷ Population health
- ▷ Application data

Put it all in a data bucket

But don't look at it

Except when you do

## Artificial nonintelligence

Machine learning for "this is fine"



## Machine Learning in Elastic Search

"this is **not** fine for 10am on Tuesday"

# Managing your things

## Big data bassackwards

## Addressing the fleet

Saltstack - rich syntax for addressing sets of devices

## Addressing the fleet

MQTT comms - bidirectional messaging but little structure

## Addressing the fleet

Broadcasts - IP or otherwise

## Reactive architectures

Rule based response to events.

## Coda

- ▷ IoT (for me) is about relieving Future Shock
- ▷ IoT's trinity: Hardware, Comms, Data
- ▷ Crossing the Air Gap
- ▷ Flowing to the sea
- ▷ Case studies
- ▷ Making sense of it all
- ▷ Knowing when to run



## Homework

- ▷ Make your systems amenable to cooperation
- ▷ Try out Amazon GreenGrass - lambda functions on your Raspberry Pi
- ▷ Keep an eye on beta offerings from Google and Microsoft.
- ▷ Use a message fabric to instrument your distributed systems
- ▷ Try out Elastic or Google ML for anomaly detection
- ▷ What cool things can you do with LoRa and 300bits per second?



## Resources, Questions

Related talks - <http://christopher.biggs.id.au/#talks>

### Me - Christopher Biggs

- ▷ Twitter: @unixbigot
- ▷ Email: [christopher@biggs.id.au](mailto:christopher@biggs.id.au)
- ▷ Slides, and getting my advice: <http://christopher.biggs.id.au/>
- ▷ Accelerando Consulting - IoT, DevOps, Big Data - <https://accelerando.com.au/>

## Links

- ▷ [Open source Bluetooth Mesh](#) on current hardware
- ▷ Future Standard [Bluetooth mesh networking](#)
- ▷ [Public access LoRaWAN](#) in Sydney
- ▷ The '[mosquitto](#)' MQTT broker:
- ▷ Example of [how I use SaltStack to provision MQTT](#)
- ▷ [Asynchronous Microservices](#)