Running enterprise-grade Edge applications with EdgeX on Ubuntu Core

EdgeXFoundry Adopter Series

EdgeX Team

Introduction

Agenda

What are you learning today?

- Landscape
- What’s Ubuntu Core
- EdgeX as snaps
- Canonical’s EdgeX Portfolio
- Demo

What are we learning today?

- Q&A + Feedback

Tony Espy

Technical Architect, IoT Field Engineering

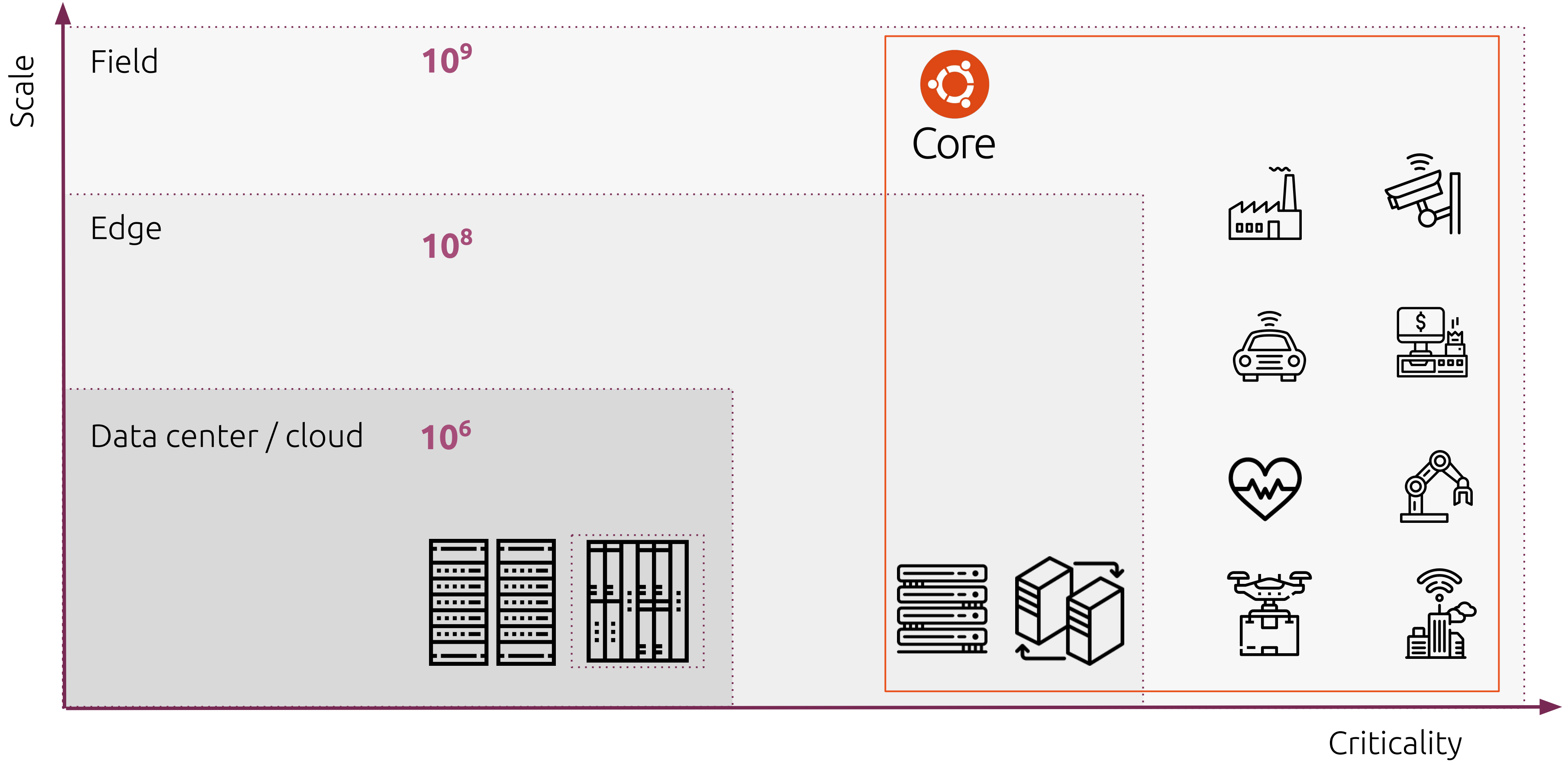
tony.espy@canonical.com

David Beamonte

IoT & Embedded Product Manager

david.beamonte@canonical.com

For the next billions of devices



Scale

Field 10^9

Edge 10^8

Data center / cloud 10^6

Core

Criticality

The IoT&Edge need a new breed of embedded systems

Security

Connectivity

Availability

Longevity

Manageability

Maintainability

Reliability

Serviceability

Just focus on your Application

Time To Market

Operations & Maintenance

Out-of-the-box Security

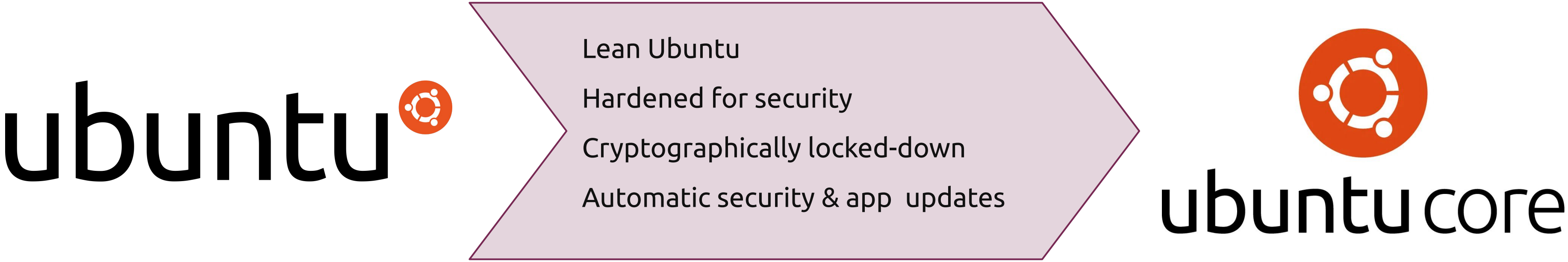
10 YEARS

Support

Ubuntu Core

Core is...

ubuntu primed for secure smart things

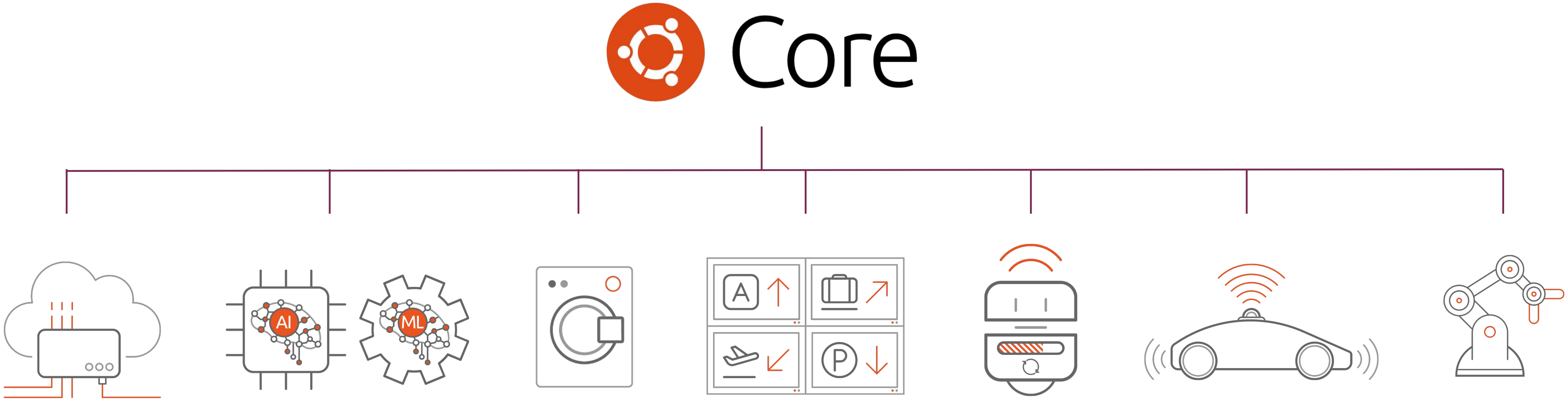


- Lean Ubuntu
- Hardened for security
- Cryptographically locked-down
- Automatic security & app updates

One Core to rule them all

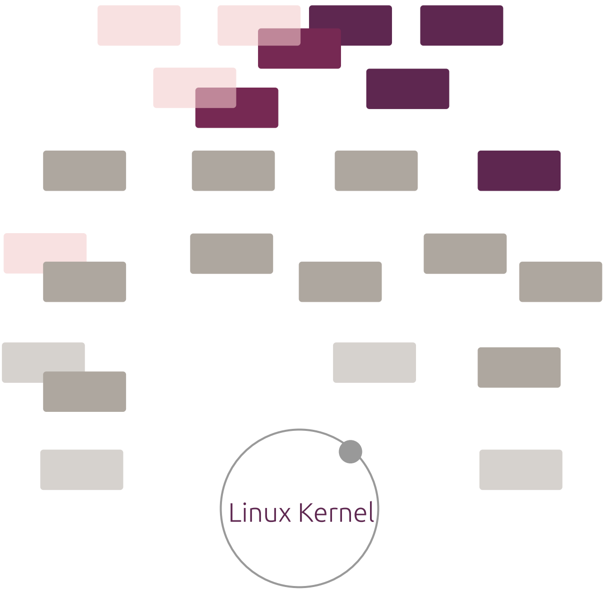
OUR GOAL

Best-in-class and reference solution for all Edge/IoT verticals

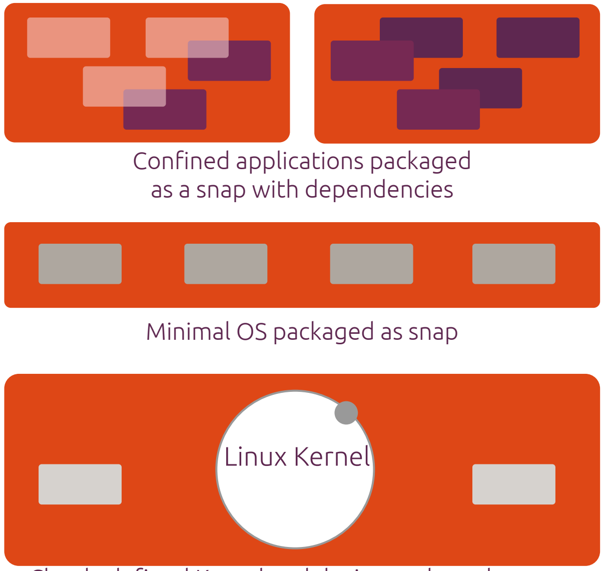


A new OS paradigm

Classic



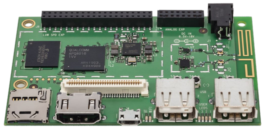
Ubuntu Core



Confined applications packaged
as a snap with dependencies

Minimal OS packaged as snap

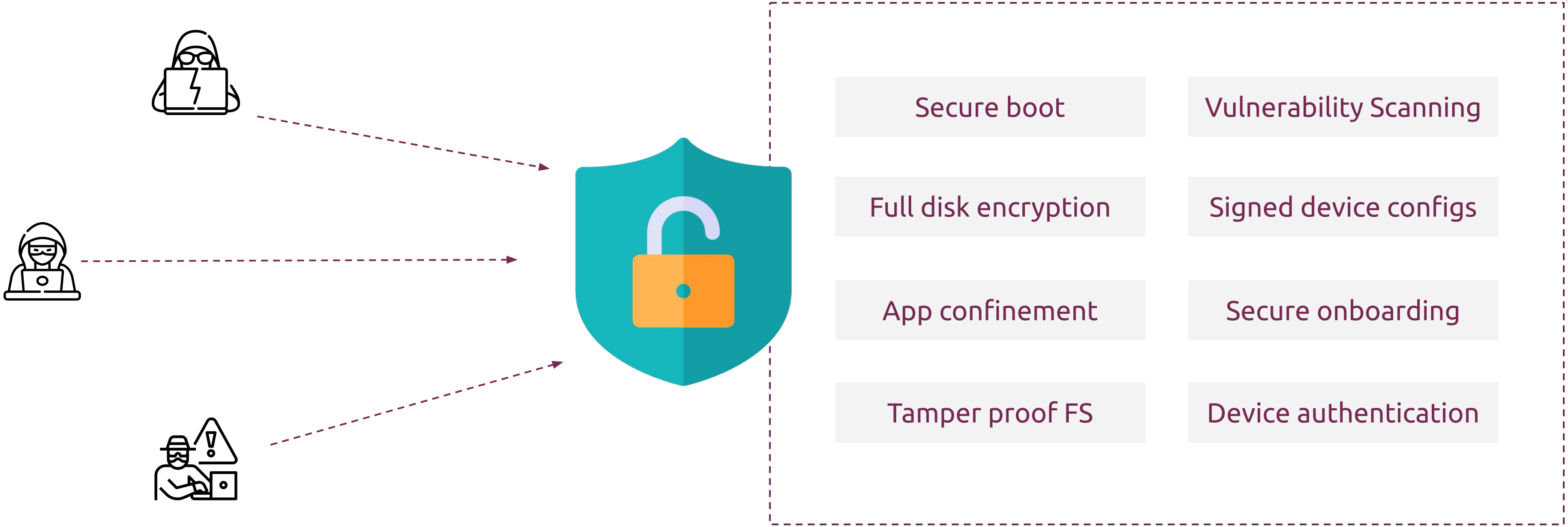
Clearly defined Kernel and device packaged as snap



Legend:

Application A ▨ Application B ▨ OS package ▨ Shared library ▨ Device driver ▨

Ultra-secure out of the box



- Secure boot
- Full disk encryption
- App confinement
- Tamper proof FS
- Vulnerability Scanning
- Signed device configs
- Secure onboarding
- Device authentication

Low touch maintenance

To minimise operational downtime

Self-healing

Reliable systems that automatically roll-back in case of a failed update

Always backed up

Devices and application configurations are always backed up for easy recovery

Remotely accessible

Natively IP addressable and remotely accessible for operations

Long-term security support

+10 years

Security patches, bug fixes and software upgrades delivered OTA

Deploying applications

Snap Store

- Open public store
- Default for Devices

Brand Store

- Selected list of snaps for groups of devices
- Snap Store as a service
- Full control over store contents
- Review queue for any 3rd party snaps
- Select periodicity of updates

Hosted by Canonical
Easy management (CLI or web)
Get published in minutes!

Tech description

Snaps - technical overview

- hybrid of a traditional Linux package and a container
- digitally signed
- updates are transactional
    - and support binary delta downloads
- run on any Linux distro with snapd installed**
- unlike typical OCI containers, can easily provide multiple services

** full confinement requires Ubuntu (e.g. Core, Desktop, Server, …)

EdgeX Foundry snaps

- Canonical publishes the EdgeX snaps in the Snap Store
  - snaps are based on upstream releases
  - no changes are made to core functionality, however...
  - we maintain GitHub mirrors which give us patch capability
- The snap edgexfoundry provides the base set of EdgeX services
- Standalone snaps exist for app-service-configurable, and a growing number of device services

EdgeX Foundry snaps

- edgexfoundry
- edgex-cli
- edgex-app-service-configurable
- edgex-device-camera
- edgex-device-grove**
- edgex-device-modbus
- edgex-device-mqtt
- edgex-device-rest
- edgex-device-llrp-rfid**

edgexfoundry snap

- Provides:
  - the minimum set of services required to run EdgeX
  - runtime dependencies (e.g. Consul, Redis, …)
  - rules engine (Kuiper)
  - device-virtual
- Some services are disabled by default (e.g. support-*, SMA)
  - but can be individually enabled using snap cli or REST API
- EdgeX Security is enabled by default
- Consul is enabled by default and cannot be disabled

edgexfoundry snap cont.

- The snap store provides channels indicating risk and tracks for multi-version support
- The snaps in the latest/beta channel are all based on Hanoi:

  ```
  $ sudo snap install edgexfoundry --beta
  ```

- Ireland snaps (in-progress), can be installed from the edge channel:

  ```
  $ sudo snap install edgexfoundry --edge
  ```

- After Ireland releases, snaps will be available from the 2.0 track:

  ```
  $ sudo snap install edgexfoundry --channel=2.0/stable
  ```

Application & device service snaps

- Application and device service snaps:
  - are installed with services disabled
    - allows for external config (e.g. devices, profile, vault tokens)
  - snap content interfaces
    - allow files to be shared between snaps
  - support an autostart config option
    - if external config is provided, set to "true"

Snap configuration environment overrides

- EdgeX configuration can be overridden via per-service env vars
    - i.e. as done in docker-compose files
- The snap configure hook now provides env override support
- Setting snap .env config options adds env vars to service wrappers
- Snap config can be supplied via
    - the snap cli (e.g. snap set)**
    - another snap via snapd's REST API (aka snapd-control)**
    - the gadget snap (part of every Ubuntu Core image)

** services must be manually restarted

Snap configuration example - Core-Data

- Set Core Data's port to 2112:

```
$ sudo snap set edgexfoundry env.core-data.service.port=2112
```

- this creates a service.env file for core-data w/in the snap and sets:

```
SERVICE_PORT=2112
```

- .env files are sourced by wrapper scripts prior to launching services

Snap configuration - API Gateway

- API Gateway is used to auth external client access to EdgeX services
- It must be configured first, before it can be used
- Snap config options include:
  - Gateway user (with public cert/key)
  - TLS configuration
- These options allow an Ubuntu Core EdgeX image to be created with a pre-configured, ready-to-use API Gateway
- The security-config command is still available for advanced use cases

Snap configuration - API Gateway

- This slide shows how the API Gateway can be configured for JWT (vs. having to use the security-config command):

```
$ sudo snap set edgexfoundry \
    env.security-proxy.user=user01,USER_ID,ES256
```

```
$ sudo snap set edgexfoundry \
    env.security-proxy.public-key= |-
     -----BEGIN PUBLIC KEY-----
     MFkwEwYHKoZIzj0CAQYIKoZIz...
     .
     .
     -----END PUBLIC KEY-----
```

Constructing an Ubuntu Core EdgeX image

- Create an Ubuntu Core model assertion listing snaps
  - a signed JSON document
- Provide configuration via snap config and/or content interfaces
  - via a configuration snap (using snapd's REST API)
  - via a gadget snap
- Build an image using ubuntu-image tool
- For more details see:

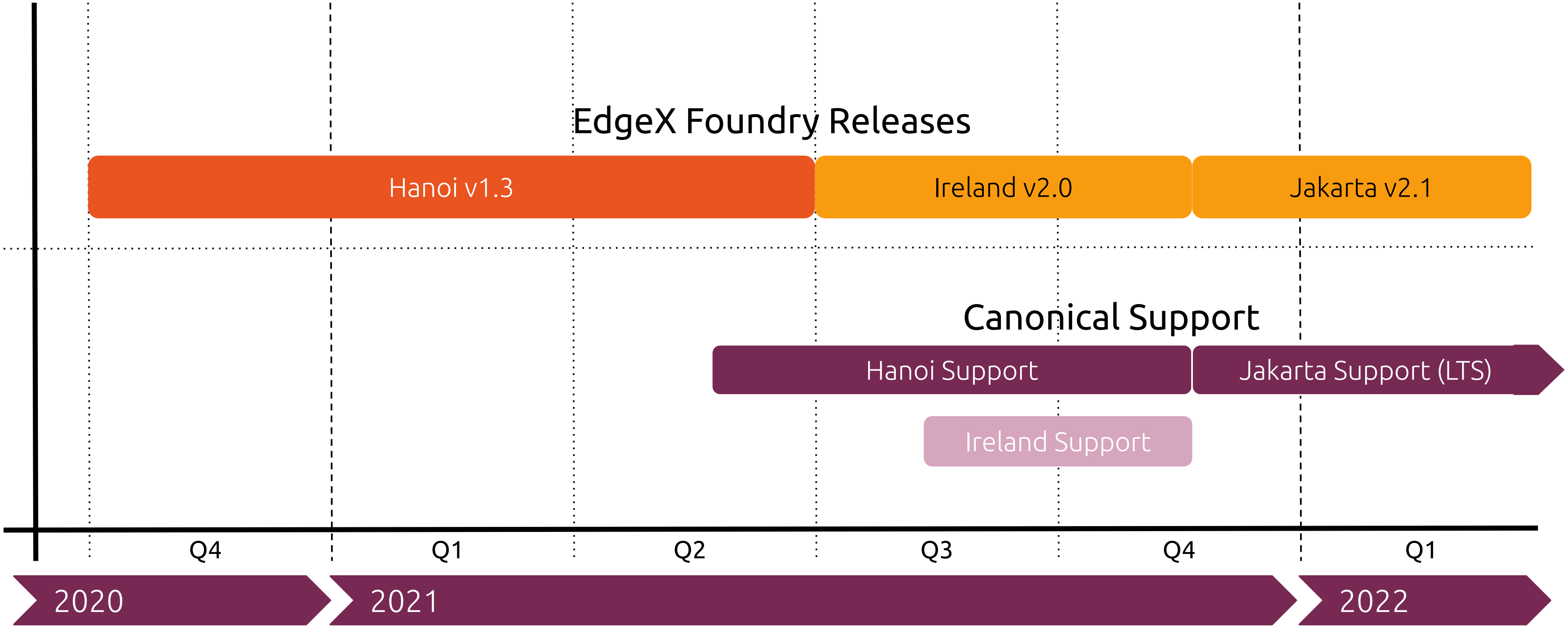
  https://ubuntu.com/core/docs/image-building

Snap documentation (Hanoi)

- edgexfoundry Snap Store page:
  https://snapcraft.io/edgexfoundry
- edgexfoundry
  https://github.com/canonical/edgex-go/blob/hanoi-launchpad/snap/README.md
- edgex-app-service-configurable
  https://github.com/canonical/edgex-app-service-configurable/blob/hanoi-lp/snap/README.md
- edgex-device-camera
  https://github.com/canonical/edgex-device-camera-go/blob/hanoi-lp/snap/README.md
- edgex-device-modbus
  https://github.com/canonical/edgex-device-modbus-go/blob/hanoi-lp/snap/README.md
- edgex-device-mqtt
  https://github.com/canonical/edgex-device-mqtt-go/blob/hanoi-lp/snap/README.md
- edgex-device-rest
  https://github.com/canonical/edgex-device-rest-go/blob/hanoi-lp/snap/README.md
- edgex-snap-hooks
  https://github.com/canonical/edgex-snap-hooks/blob/v1.0.0/README.md

Products & Roadmap

Canonical's Support Roadmap



Our products

Product acceleration

Boost TTM

Long-term engagement

Security + Support

ODM Program

Device Enablement

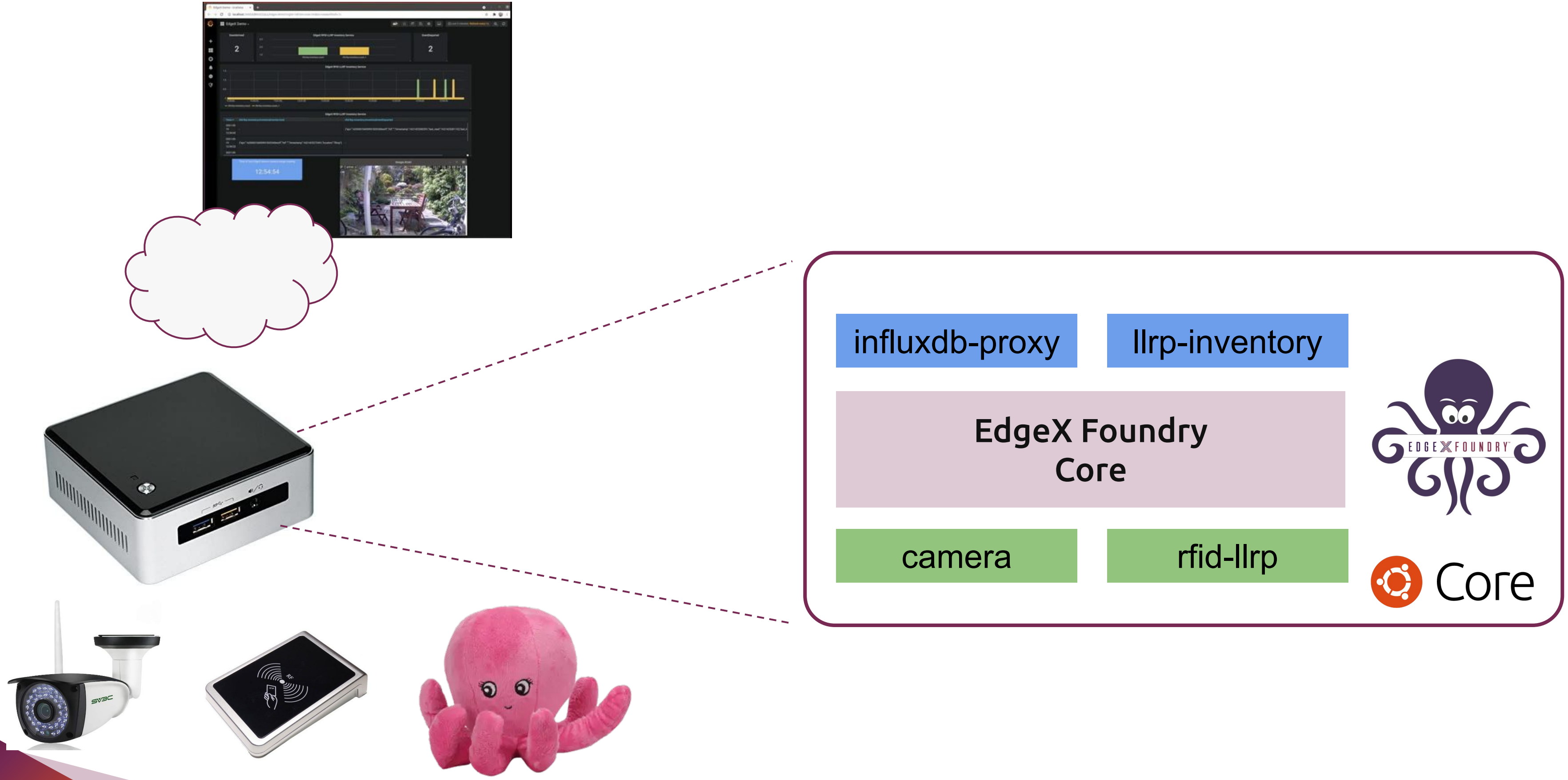
Smart-Start

OTA updates (Security + Apps)

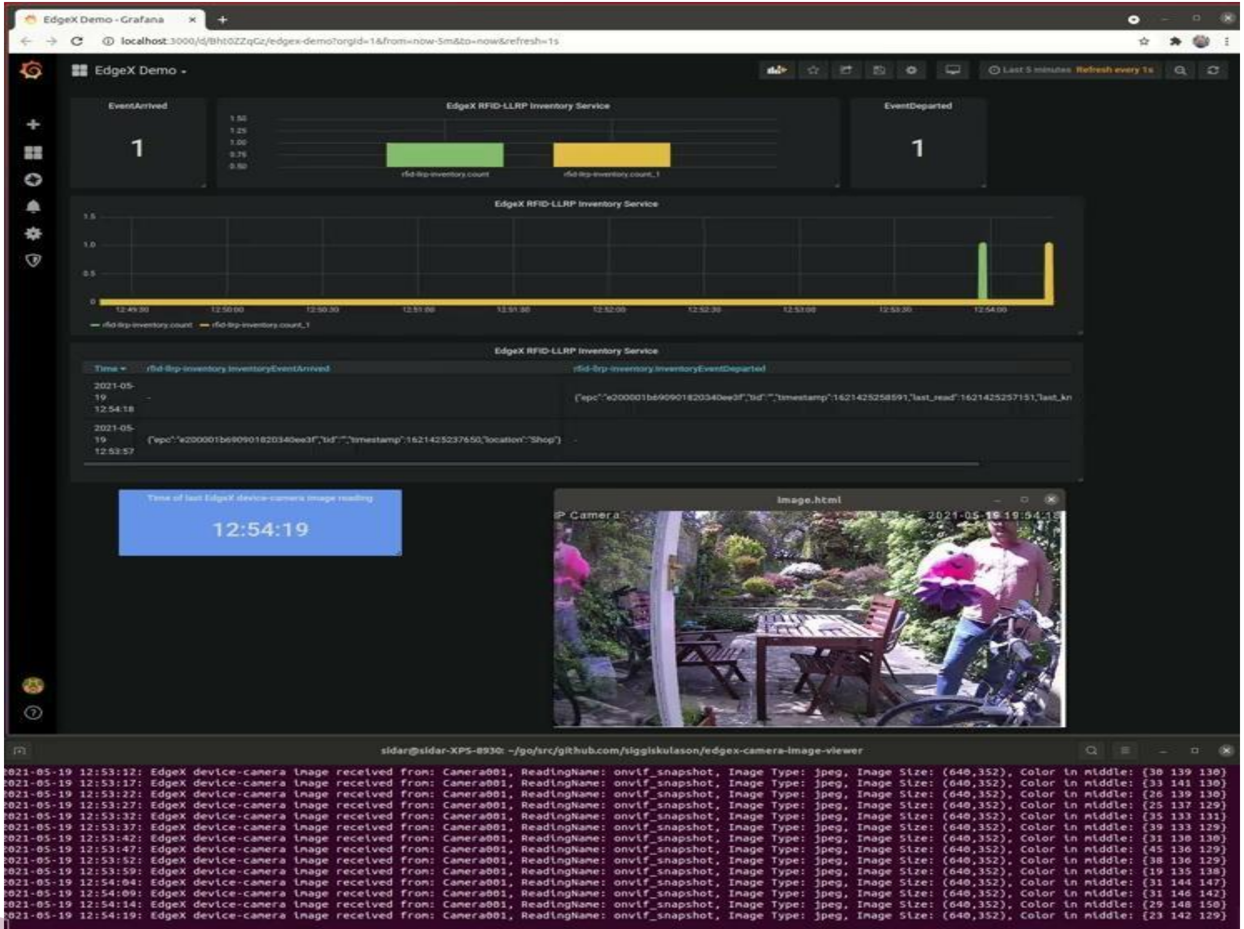
Support

Demo

Demo Description



Demo: RFID detection + Camera snapshots


EdgeX Demo - Grafana
localhost:3000/d/Bht0ZZqGz/edgex-demo?orgId=1&from=now-5m&to=now&refresh=1s
EdgeX Demo
Last 5 minutes Refresh every 1s
EventArrived
1
EdgeX RFID-LLRP Inventory Service
1.50
1.25
1.00
0.75
0.50
rfid-llrp-inventory.count
rfid-llrp-inventory.count_1
EventDeparted
1
EdgeX RFID-LLRP Inventory Service
1.5
1.0
0.5
0
12:49:30
12:50:00
12:50:30
12:51:00
12:51:30
12:52:00
12:52:30
12:53:00
12:53:30
12:54:00
rfid-llrp-inventory.count
rfid-llrp-inventory.count_1
Time of last EdgeX device-camera image reading
12:54:19
image.html
IP Camera
2021-05-19 19:54:18

EdgeX RFID-LLRP Inventory Service

Time	rfid-llrp-inventory.InventoryEventArrived	rfid-llrp-inventory.InventoryEventDeparted
2021-05-19 12:54:18	-	{"epc":"e200001b6909018203400ee3f","tid":"","timestamp":1621425258591,"last_read":1621425257151,"last_kn
2021-05-19 12:53:57	{"epc":"e200001b6909018203400ee3f","tid":"","timestamp":1621425237650,"location":"Shop"}	-

sidar@sidar-XPS-8930: ~/go/src/github.com/siggiskulason/edgex-camera-image-viewer

```
2021-05-19 12:53:12: EdgeX device-camera image received from: Camera001, ReadingName: onvif_snapshot, Image Type: jpeg, Image Size: (640,352), Color in middle: {30 139 130}
2021-05-19 12:53:17: EdgeX device-camera image received from: Camera001, ReadingName: onvif_snapshot, Image Type: jpeg, Image Size: (640,352), Color in middle: {33 141 130}
2021-05-19 12:53:22: EdgeX device-camera image received from: Camera001, ReadingName: onvif_snapshot, Image Type: jpeg, Image Size: (640,352), Color in middle: {26 139 130}
2021-05-19 12:53:27: EdgeX device-camera image received from: Camera001, ReadingName: onvif_snapshot, Image Type: jpeg, Image Size: (640,352), Color in middle: {25 137 129}
2021-05-19 12:53:32: EdgeX device-camera image received from: Camera001, ReadingName: onvif_snapshot, Image Type: jpeg, Image Size: (640,352), Color in middle: {35 133 131}
2021-05-19 12:53:37: EdgeX device-camera image received from: Camera001, ReadingName: onvif_snapshot, Image Type: jpeg, Image Size: (640,352), Color in middle: {39 133 129}
2021-05-19 12:53:42: EdgeX device-camera image received from: Camera001, ReadingName: onvif_snapshot, Image Type: jpeg, Image Size: (640,352), Color in middle: {31 130 130}
2021-05-19 12:53:47: EdgeX device-camera image received from: Camera001, ReadingName: onvif_snapshot, Image Type: jpeg, Image Size: (640,352), Color in middle: {45 136 129}
2021-05-19 12:53:52: EdgeX device-camera image received from: Camera001, ReadingName: onvif_snapshot, Image Type: jpeg, Image Size: (640,352), Color in middle: {38 136 129}
2021-05-19 12:53:59: EdgeX device-camera image received from: Camera001, ReadingName: onvif_snapshot, Image Type: jpeg, Image Size: (640,352), Color in middle: {19 135 138}
2021-05-19 12:54:04: EdgeX device-camera image received from: Camera001, ReadingName: onvif_snapshot, Image Type: jpeg, Image Size: (640,352), Color in middle: {31 144 147}
2021-05-19 12:54:09: EdgeX device-camera image received from: Camera001, ReadingName: onvif_snapshot, Image Type: jpeg, Image Size: (640,352), Color in middle: {31 146 142}
2021-05-19 12:54:14: EdgeX device-camera image received from: Camera001, ReadingName: onvif_snapshot, Image Type: jpeg, Image Size: (640,352), Color in middle: {29 148 150}
2021-05-19 12:54:19: EdgeX device-camera image received from: Camera001, ReadingName: onvif_snapshot, Image Type: jpeg, Image Size: (640,352), Color in middle: {23 142 129}
```

Wrap-up

EdgeX + Ubuntu Core

Focus on our application ⇒ Accelerate TTM

Reduce Operations & Maintenance Costs

Security, reliability and LTS

Thank you. Questions?