
cisco Live!
TURN
IT
UP
#CiscoLive

The bridge to possible

Getting Started with Model Driven Telemetry

“Towards intent based Operations"

Marisol Palmero, Tech Lead CX CTO
Josh Suhr, Principal Architect in CX PM

BRKEMT-2100

Abstract

- Every time that you install a new element in the network, from the manageability perspective, it seems that those new elements don't talk same "language" as the ones that are already installed and operational, because of the way SNMP, Syslog, NetFlow, IPSLA, NBAR, APIs, ... to communicate with them.
- This brings a big challenge in order to manage and operate networks.
- Model Driven Telemetry is coming up to provide consistency and common approach to manage networks.
- The goal of this session is to show the value of Telemetry, highlighting why is important to standardise disparate data. We´ll explain why YANG has been selected as the modeling language for the purpose of Experience Telemetry and we´d like to share with the audience the ONE Cisco approach across Telemetry.

Marisol Palmero
mpalmero@cisco.com

Josh Suhr
josuhr@cisco.com

Demystifying the implementation of Model Driven Telemetry with YANG

"Differentiated customer and partner value starts with measuring, and thus Telemetry is the central enabling mechanism threading and correlating data-derived insight. Experience Telemetry elevates customer and partner connection up the stack. When Telemetry is ingested in a normalized way across the whole portfolio, and correlated on the end-to-end architecture, from the data center to the campus, from collaboration to security endpoints, that value is compounded."

Carlos Pignataro
Customer Experience CTO, CTO

Agenda

- Out with the old, in with the new: Model-Driven Telemetry
- Operational vs. *Experience* Telemetry
- Real-world example: Experience Telemetry use case
- Key Takeaways

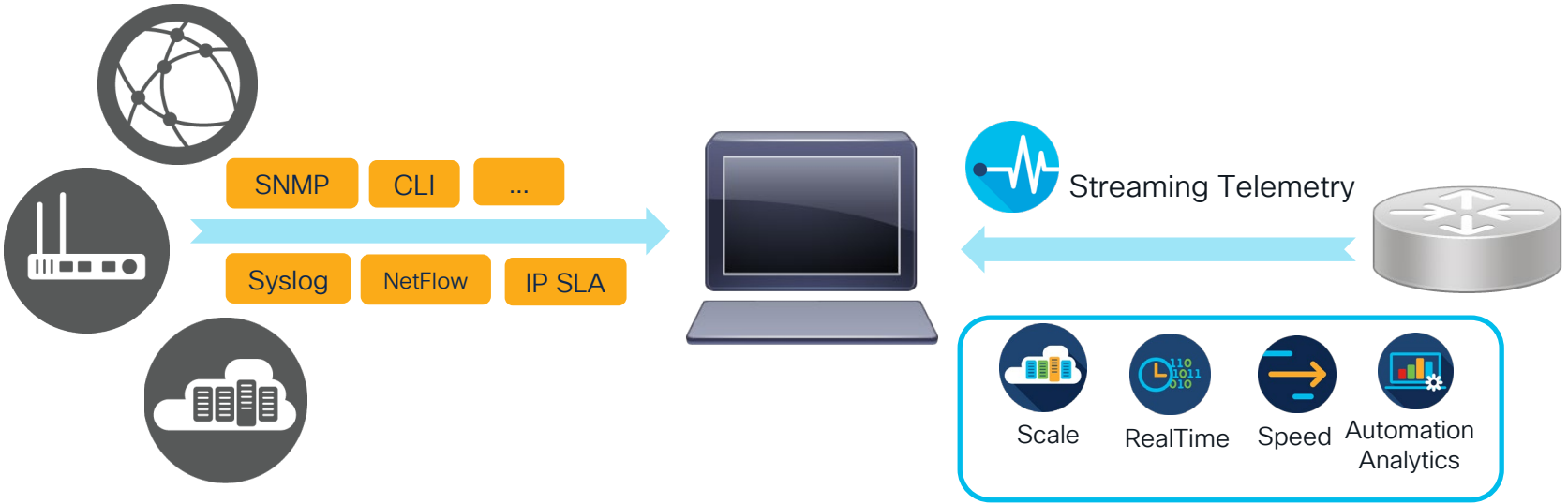
Out with the Old, In with the New: Model-Driven Telemetry

cisco Live!

Telemetry

What is it?

Telemetry: An *automated* communications *process* by which *measurements* and other data are *collected* at remote or inaccessible points and *transmitted* to receiving equipment *for monitoring*. The word is derived from Greek roots: *tele = remote,* and *metron = measure*. *https://en.wikipedia.org/wiki/Telemetry*



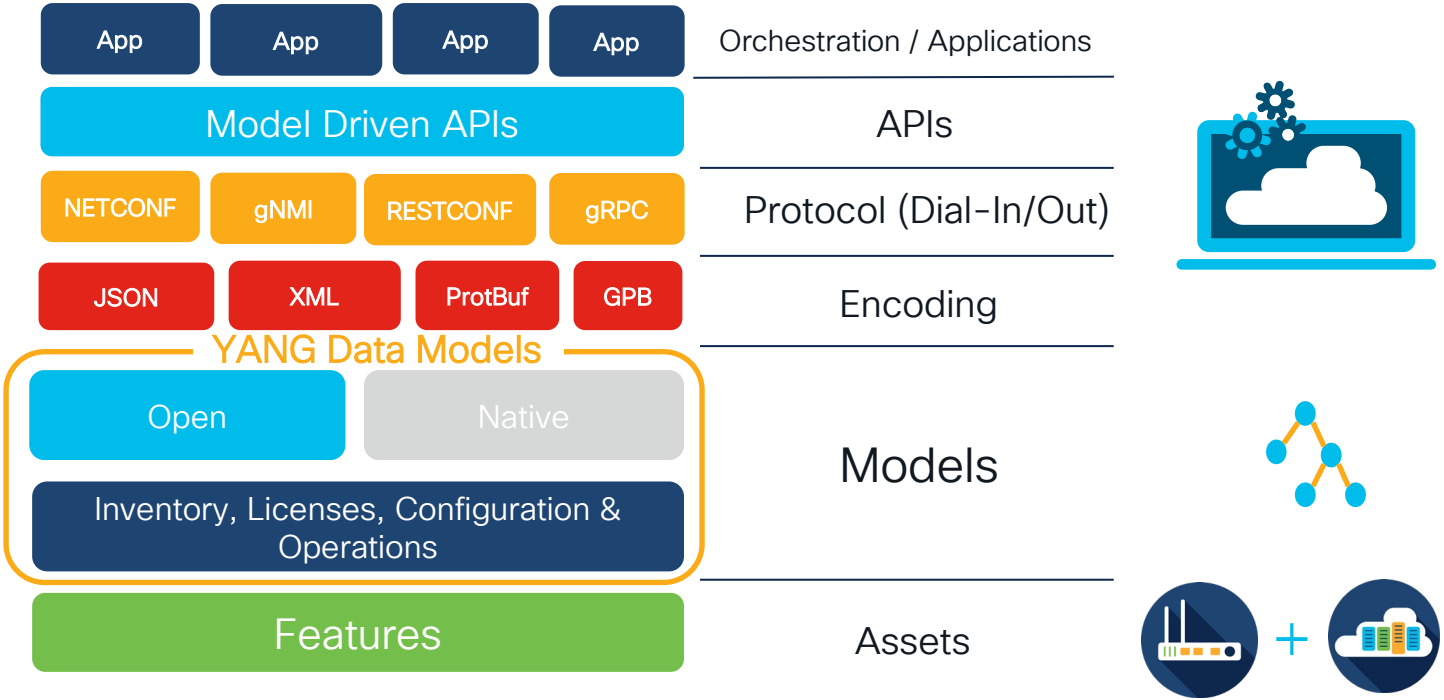
What is new about Telemetry?

Model Driven Telemetry (MDT)

- Provides a mechanism to stream data from a model-driven telemetry-capable device to a destination:
  - Push vs Pull
  - Subscriber-based
  - Combines periodic subscriptions with Event-driven
  - Scalability
  - Diversification of manageability

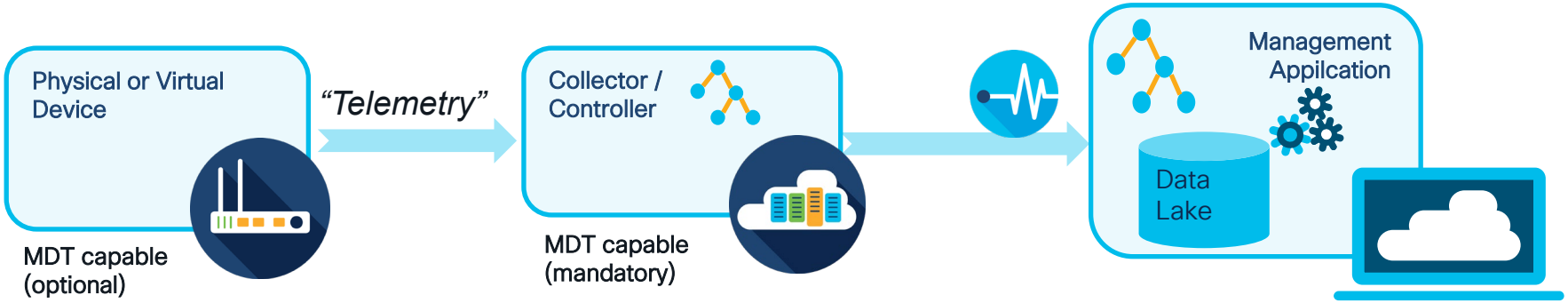
MDT can ensure that those vast quantities of data are truly usable, when they are well structured

Data Model Driven Telemetry with YANG



YANG Data Models are the foundation of automation Structured, Well-Defined, Programmable Network API

Our Telemetry Story



By modeling data with YANG, MDT can ensure that those vast quantities of data are truly usable

Operational vs. Experience Telemetry

How to Classify Telemetry Data?

It depends on who you talk to...

Operational Telemetry

In IT, we often think of telemetry as the mechanism by which we learn about how our networks (or other devices) are *operating* – interface utilization, up/down status, etc.

How to Classify Telemetry Data?

It depends on who you talk to...

Operational Telemetry

In IT, we often think of telemetry as the mechanism by which we learn about how our networks (or other devices) are *operating* – interface utilization, up/down status, etc.

Experience Telemetry

However – telemetry can provide us with valuable data about anything!

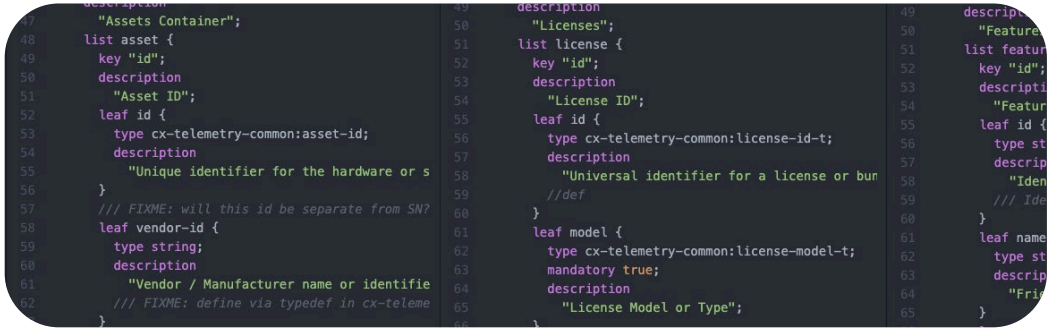
Metrics relevant to our business can be just as helpful as metrics about our operations. Business developers and executives talk about business telemetry; at Cisco we refer to this as **Experience Telemetry**.

How to Classify Telemetry Data?

Operational Telemetry

Experience Telemetry

Best practices that apply to Operational Telemetry also apply to Business Telemetry – such as the importance of **standardizing** and **modeling** your data.



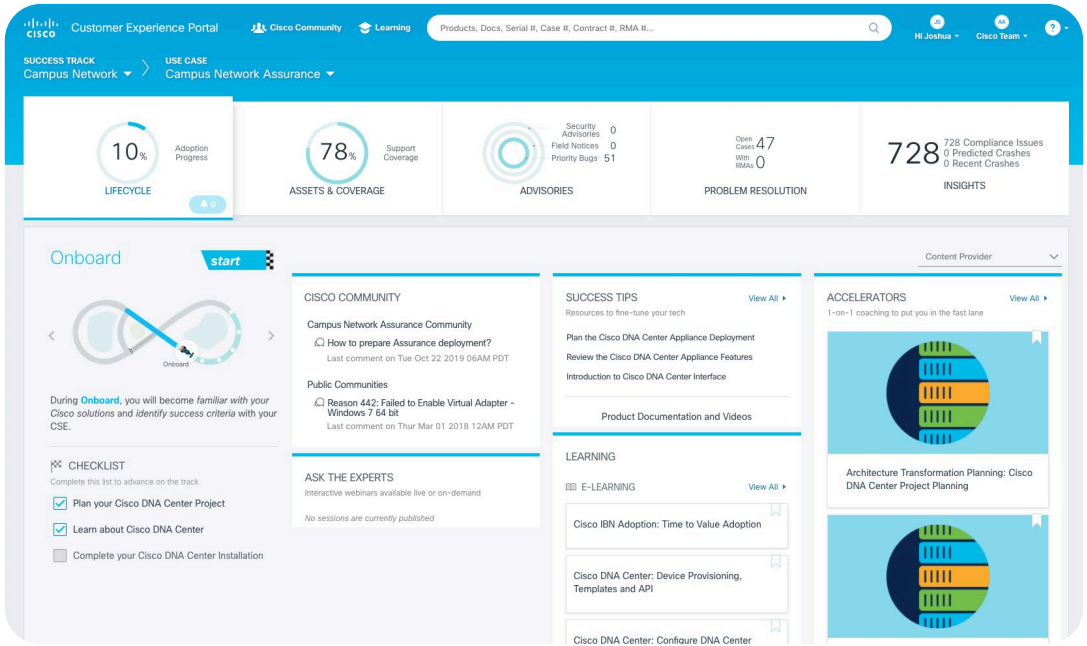
What does Experience Telemetry look like?

- Report the business value of a solution, device, or other asset
  - Asset Identification – a unique product, feature, user, …
  - Associated entities / dependencies
  - License – one time purchase, subscription
  - State – (de-)activated
  - Usage - Usage information for the entity, utilization, performance, …

Experience telemetry often uses Operational telemetry objects!

Telemetry Use Cases

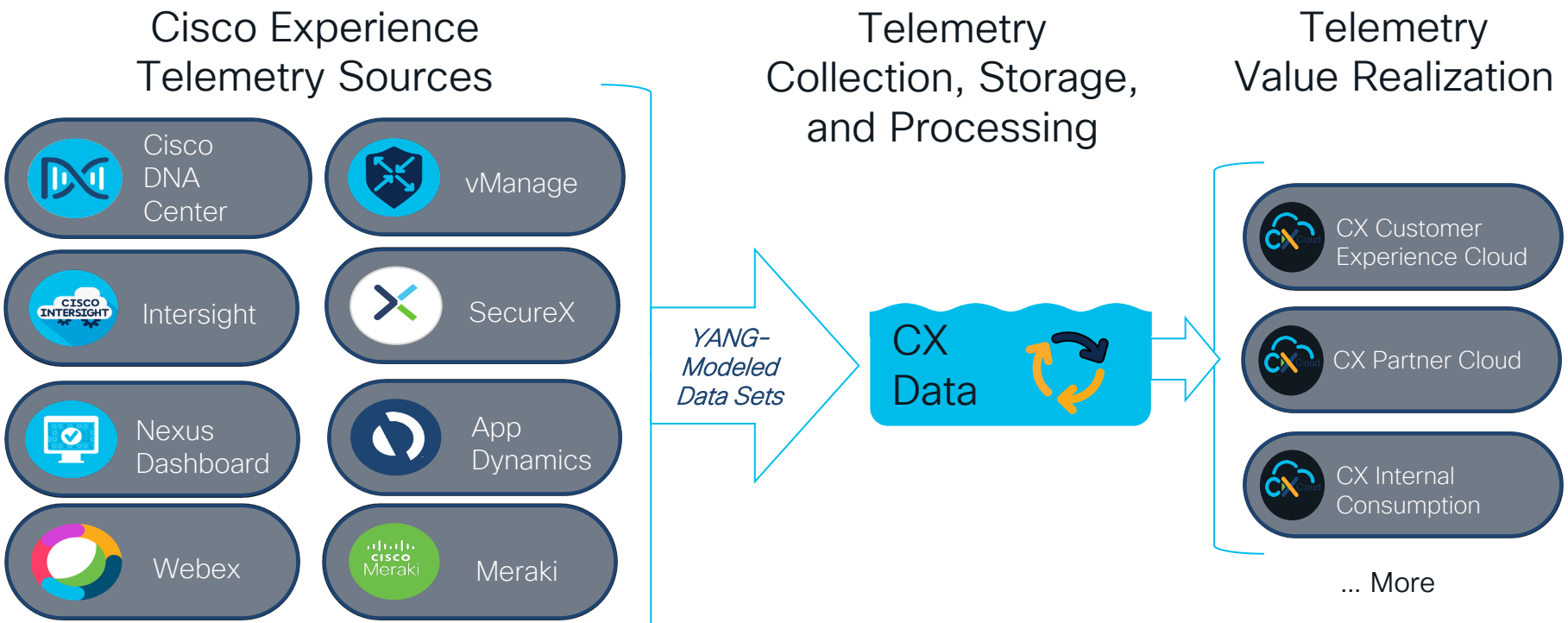
Making it Real: CX Cloud



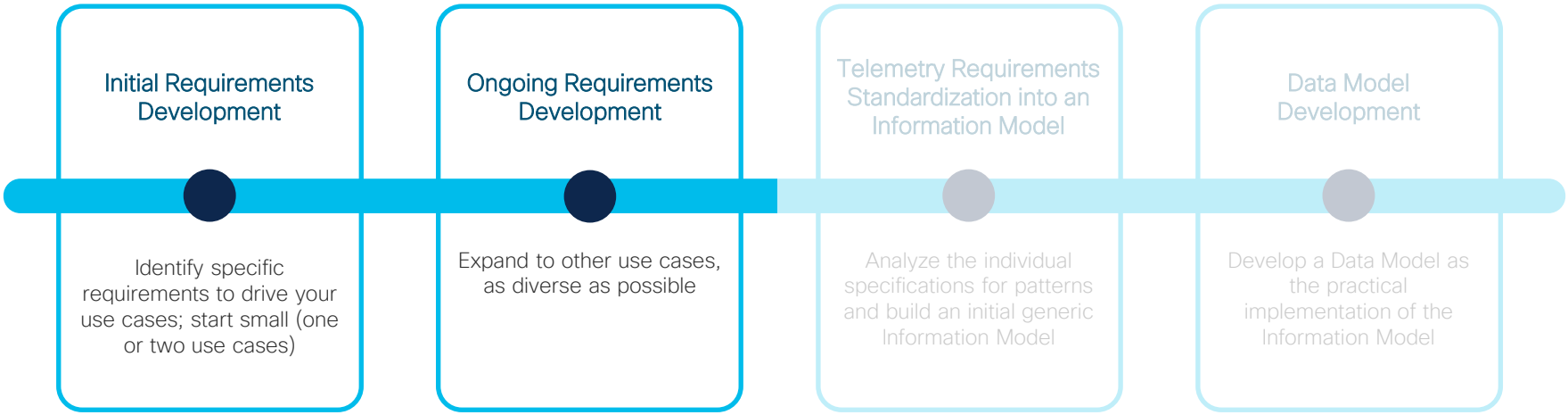
CX Cloud aggregates information across multiple solutions, product families, etc. to provide collaborative intelligence.

It heavily leverages both Operational and Experience telemetry.

CX Cloud Telemetry Data Flow



Our Modeling Journey



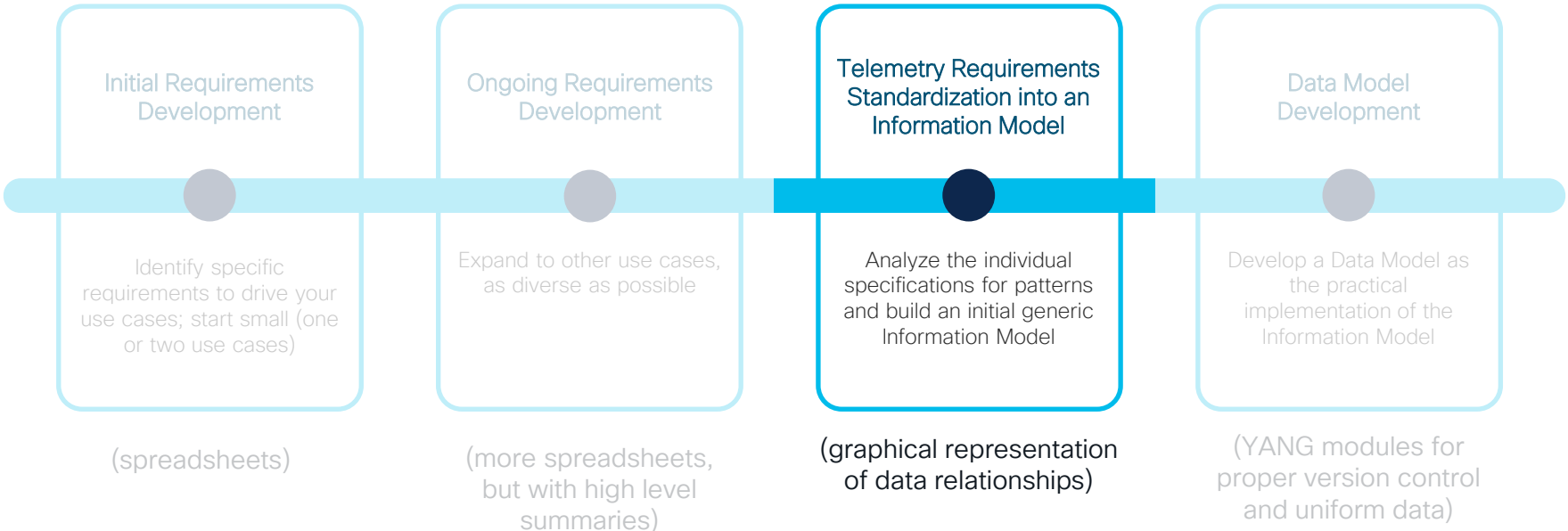
Modeling Across Varying Datasets

	Data Source Schema (Serialization)	Dataset “licensePurchased”	Reference
 Cisco DNA Center	AVRO	licenseItem "maxUsageCount” ”usageCountRemaining”	Sample AVRO file
 Webex Meetings	Json	list-licenses “totalUnits” “consumedUnits”	https://developer.webex.com/docs/api/v1/licenses/list-licenses
Meraki	Json	licenses “totalDurationInDays” “durationInDays”	https://developer.cisco.com/meraki/api/#!get-organization-licenses

Modeling Across Varying Datasets

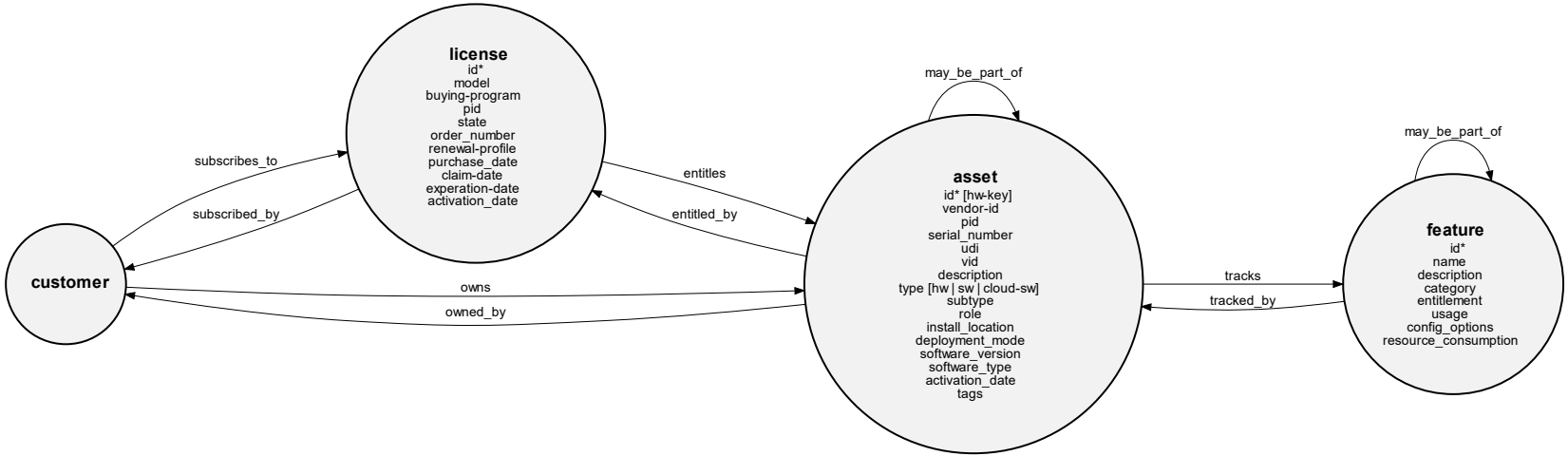
	Data Source Schema (Serialization)	Dataset “licensePurchased”	Reference
 Cisco DNA Center	AVRO	licenseItem "maxUsageCount” ”usageCountRemaining”	Sample AVRO file
 Webex Meetings	Json	list-licenses “totalUnits” “consumedUnits”	https://developer.webex.com/docs/api/v1/licenses/list-licenses
Meraki	Json	licenses “totalDurationInDays” “durationInDays”	https://developer.cisco.com/meraki/api/#!get-organization-licenses

Our Modeling Journey



Information Model

Customer & Partner Telemetry



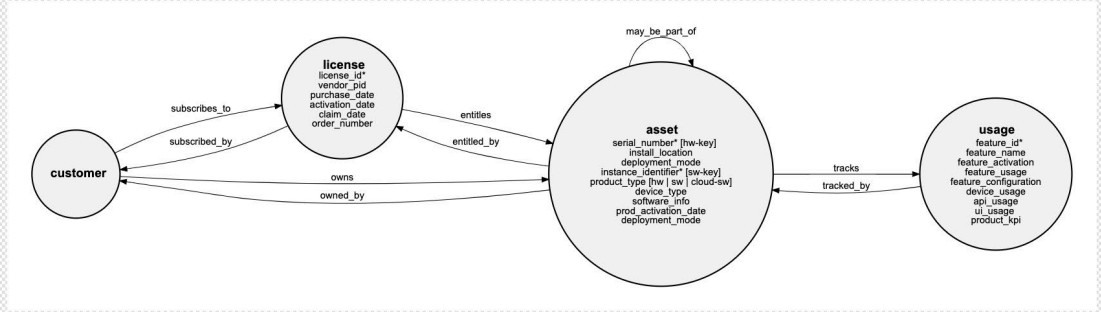
Information Model

GraphViz for Information Model Representation

graph-information-model.dot

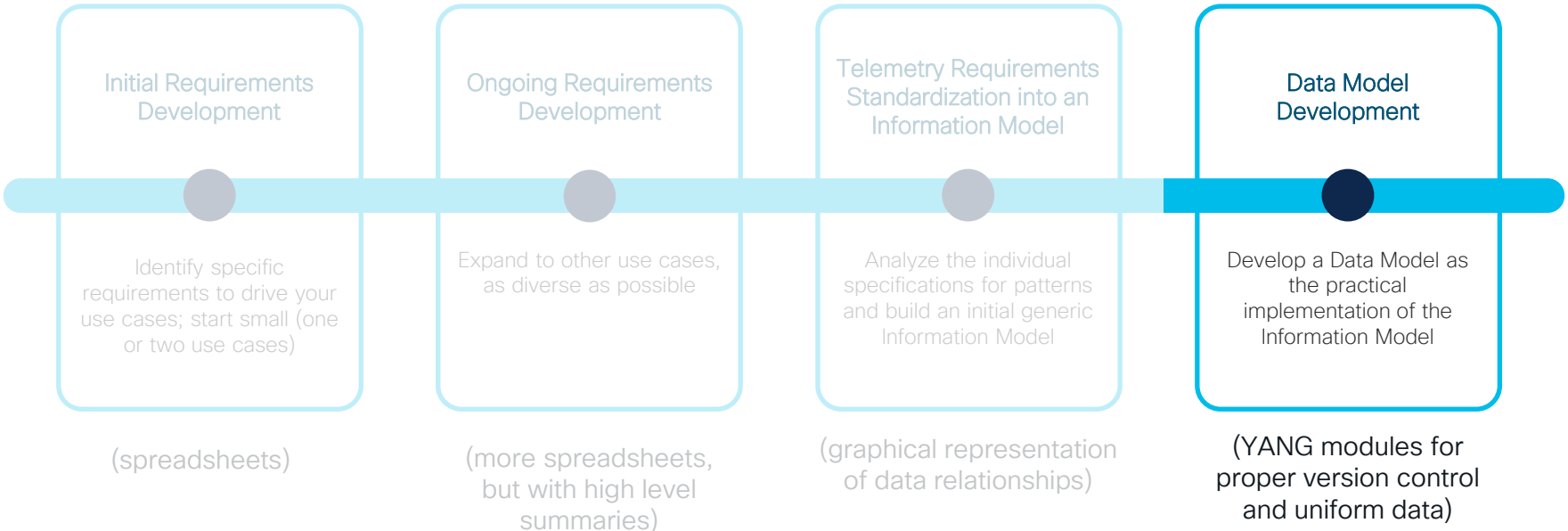
```
digraph {
    graph [pad="0.5", nodesep="0.5", ranksep="2", fontname="a
    node [shape = circle, style="filled,bold", color=black,
      fillcolor="#F2F2F2", fontname=arial, fontsize=18]
    edge [fontname=Arial]
    layout=dot
    rankdir=LR;

customer [label=<<b>customer</b>>]

customer -> license [label=subscribes_to]
customer -> asset [label=owns]

license [label=<<b>license</b><BR />
<FONT POINT-SIZE="14">license_id*<BR />
  vendor_pid<BR />
  purchase_date<BR />
  activation_date<BR />
  claim_date<BR />
  order_number<BR />
  </FONT>>]

license -> asset [label=entitles]
license -> customer [label=subscribed_by]

asset [label=<<b>asset</b><BR />
  <FONT POINT-SIZE="14">serial_number* [hw-key]<BR />
  install_location<BR />
  deployment_mode<BR />
  instance_identifier* [sw-key]<BR />
  product_type [hw | sw | cloud-sw]<BR />
  device_type<BR />
  software_info<BR />
  prod_activation_date<BR />
  deployment_mode<BR /></FONT>>]
usage [label=<<b>usage</b><BR />
  <FONT POINT-SIZE="14">feature_id*<BR />
  feature_name<BR />
  feature_activation<BR />
  feature_usage<BR />
  feature_configuration<BR />
  device_usage<BR />
```

graph-information-model.dot preview+ (d...

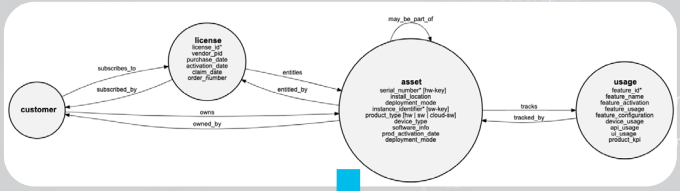


https://graphviz.org

Our Modeling Journey



Modeling in YANG



Project

- cx-telemetry-specification
  - .circleci
  - .git
  - common
  - design-doc
  - dist
  - rfc
  - yang-mod
    - .cache
    - .gitignore
    - cx-telemetry-assets-inventory.yang
    - cx-telemetry-common.yang
    - cx-telemetry-licenses.yang
    - cx-telemetry-product-usage.yang
    - ietf-palmero-telemetry-assets-inventory.y
    - ietf-palmero-telemetry-assets-usage.yan
    - ietf-palmero-telemetry-common.yang
    - ietf-palmero-telemetry-licenses.yang
    - Makefile
    - README.md
  - .gitattributes
  - .gitignore
  - Makefile
  - README.md

cx-telemetry-assets-inventory.yang

```
// FIXME: update contact as needed

description
  "Cisco";

revision 2020-10-27 {
  description
    "Update
  reference
    "https://www.cisco.com/c/en/us/products/unig
}

container capture-info {
  config false;
  description
    "Capture information for this data";
  leaf collected-on {
    type yang:date-and-time;
    description
      "Time at which this data was collected";
  }
  leaf collected-from {
    type string;
    description
      "Identifier for original source of this da
  }
}
container assets {
  config false;
  description
    "Assets Container";
  list asset {
    key "id";
    description
      "Asset ID";
    leaf id {
      type cx-telemetry-common:asset-id;
      description
        "Unique identifier for the hardware or s
    }
    /// FIXME: will this id be separate from SN?
    leaf vendor-id {
      type string;
      description
        "Vendor / Manufacturer name or identifie
      /// FIXME: define via typedef in cx-teleme
    }
    leaf pid {
      type string;
      description
        "Part or Product Identifier";
```

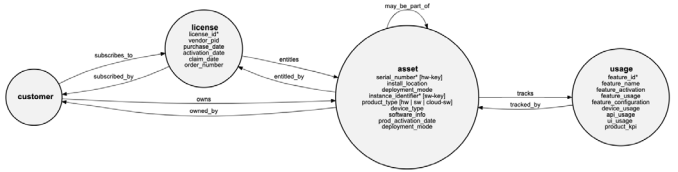
cx-telemetry-licenses.yang

```
// FIXME: update contact as needed

description

container capture-info {
  config false;
  description
    "Capture information for this data";
  leaf collected-on {
    type yang:date-and-time;
    description
      "Time at which this data was collected";
  }
  leaf collected-from {
    type string;
    description
      "Identifier for original source of this data
  }
}
container licenses {
  config false;
  description
    "Licenses";
  list license {
    key "id";
    description
      "License ID";
    leaf id {
      type cx-telemetry-common:license-id-t;
      description
        "Universal identifier for a license or bun
      //def
    }
    leaf model {
      type cx-telemetry-common:license-model-t;
      mandatory true;
      description
        "License Model or Type";
    }
    leaf buying-program {
      type cx-telemetry-common:license-buying-prog
      description
```

cx-telemetry-product-usage.yang | cx-telemetry-common.yang

```
// FIXME: update contact as needed

description

rate changes from IETF module";

 and Partner Telemetry Program";

container capture-info {
  config false;
  description
    "Capture information for this data";
  leaf collected-on {
    type yang:date-and-time;
    description
      "Time at which this data was collected";
  }
  leaf collected-from {
    type string;
    description
      "Identifier for original source of this data";
  }
}
container features {
  config false;
  description
    "Features";
  list feature {
    key "id";
    description
      "Feature List";
    leaf id {
      type string;
      description
        "Identify BE/Product/Feature uniquely across all products";
      /// Identify naming convention via typedef in cx-telemetry-comm
    }
    leaf name {
      type string;
      description
        "Friendly name of the feature";
    }
    leaf description {
      type string;
      description
        "Brief description of the feature";
```

Modeling in YANG

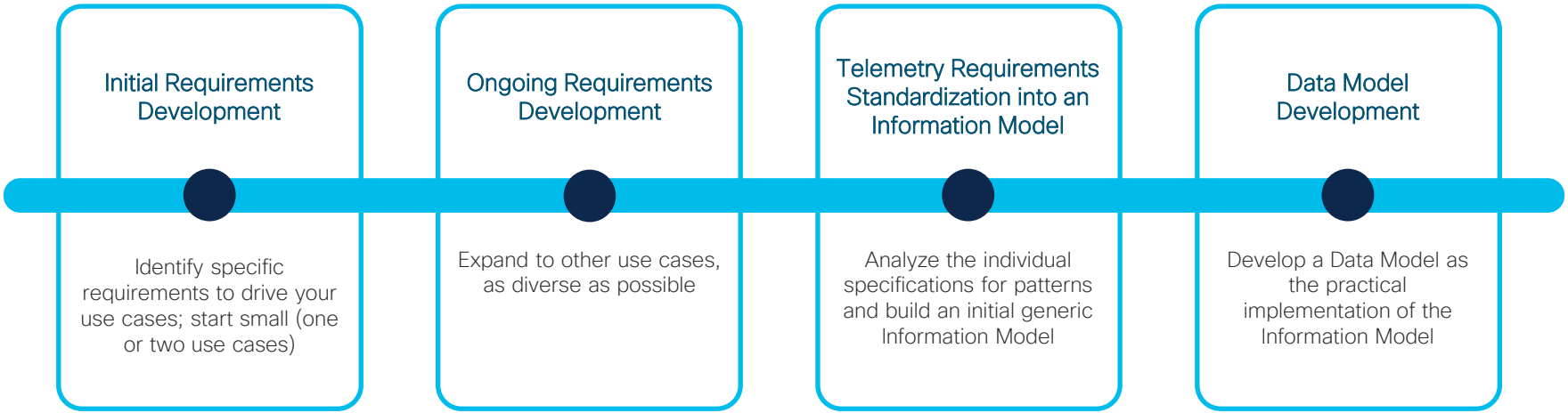



```
}
container assets {
  config false;
  description
    "Assets Container";
  list asset {
    key "id";
    description
      "Asset ID";
    leaf id {
      type cx-telemetry-common:asset-id;
      description
        "Unique identifier for the hardware or s
    }
    /// FIXME: will this id be separate from SN?
    leaf vendor-id {
      type string;
      description
        "Vendor / Manufacturer name or identifie
      /// FIXME: define via typedef in cx-teleme
    }
    leaf pid {
      type string;
      description
        "Part or Product Identifier";
    }
    leaf serial-number {
```

```
  }
}
container licenses {
  config false;
  description
    "Licenses";
  list license {
    key "id";
    description
      "License ID";
    leaf id {
      type cx-telemetry-common:license-id-t;
      description
        "Universal identifier for a license or bun
      //def
    }
    leaf model {
      type cx-telemetry-common:license-model-t;
      mandatory true;
      description
        "License Model or Type";
    }
    leaf buying-program {
      type cx-telemetry-common:license-buying-prog
      description
        "License buying program, if applicable";
    }
    leaf offer-type {
```

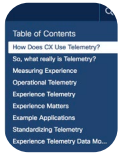
```
  }
}
container features {
  config false;
  description
    "Features";
  list feature {
    key "id";
    description
      "Feature List";
    leaf id {
      type string;
      description
        "Identify BE/Product/Feature uniquely
      /// Identify naming convention via type
    }
    leaf name {
      type string;
      description
        "Friendly name of the feature";
    }
    leaf description {
      type string;
      description
        "Brief description of the feature";
    }
    leaf category {
```

Our Modeling Journey

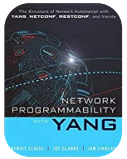


References & Key Takeaways

References



[Experience Telemetry by CX Whitepaper]



[Network Progammability with YANG]

Just getting started with YANG? Whether for network programmability or other needs like business telemetry, this book is a great place to start.

[IETF RFC 7950: The YANG 1.1 Data Modeling Language]

[Getting Started with Data Modeling]

[Cisco CX Cloud]

Learn more about the use case we highlighted here – the Cisco CX Cloud portal driven by collaborative intelligence (and lots of telemetry!)

Key Takeaways

Telemetry is not just for operations – it can provide insights to many other challenging problems

Standardization is important – building an Information Model helps you visualize your data sets, and building a Data Model from that Information Model makes it "real" – regardless of the telemetry type

YANG can be a great option for modeling your data – it provides structure and rigor to standardize your data, but is also readable and easily translated to other languages and for other uses

Think outside the box! How do you use operational or business telemetry in your day-to-day job, and how might it be improved through standardization?

The bridge to possible

Thank you


cisco Live!
TURN
IT
UP
#CiscoLive