BRINGING THE WORLD TOGETHER

VOLTHA OPENOMCI

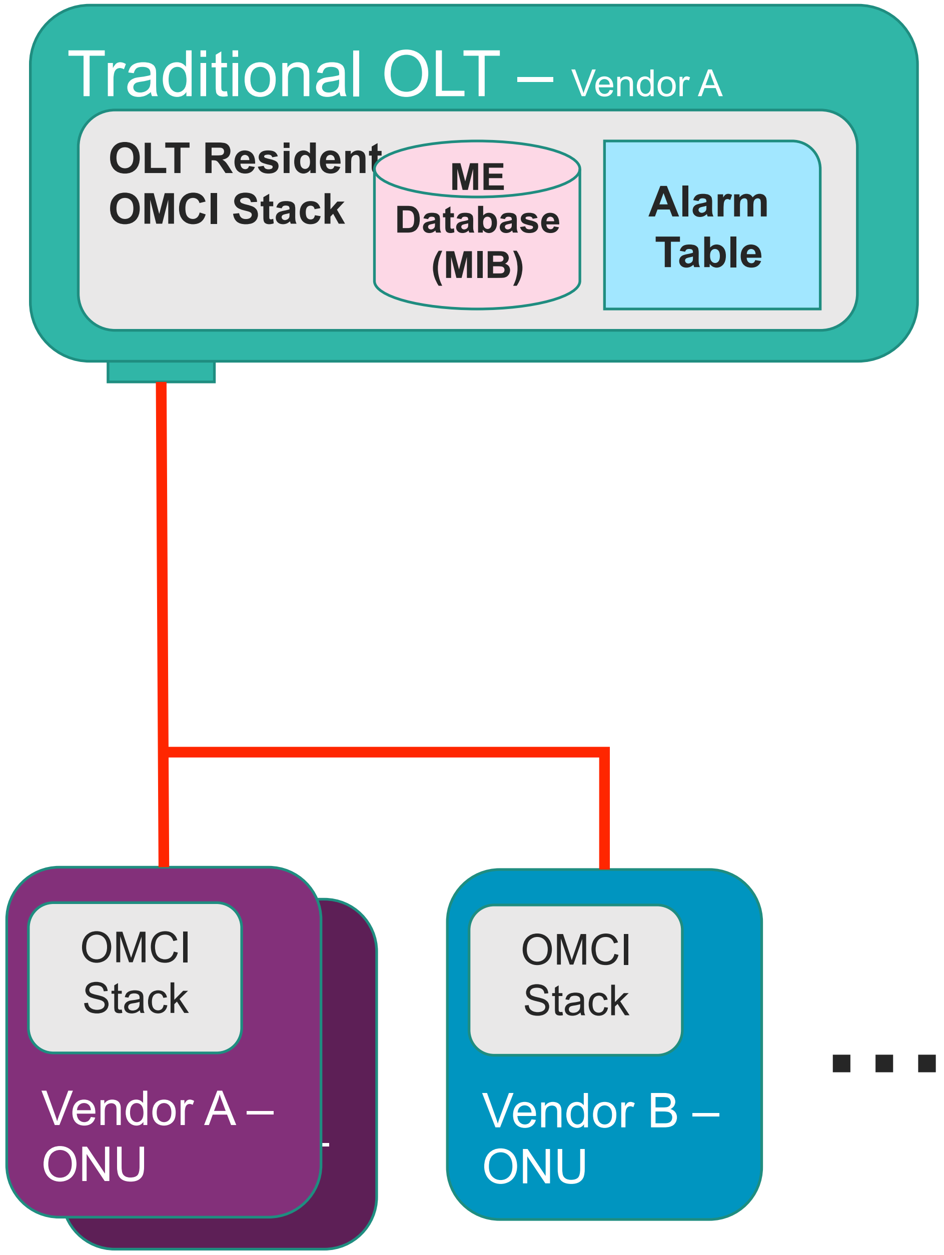
An Abstraction of the Typical OLT-Resident OMCI Stack

Chip Boling, Principal Software Design Engineer

December 6, 2019

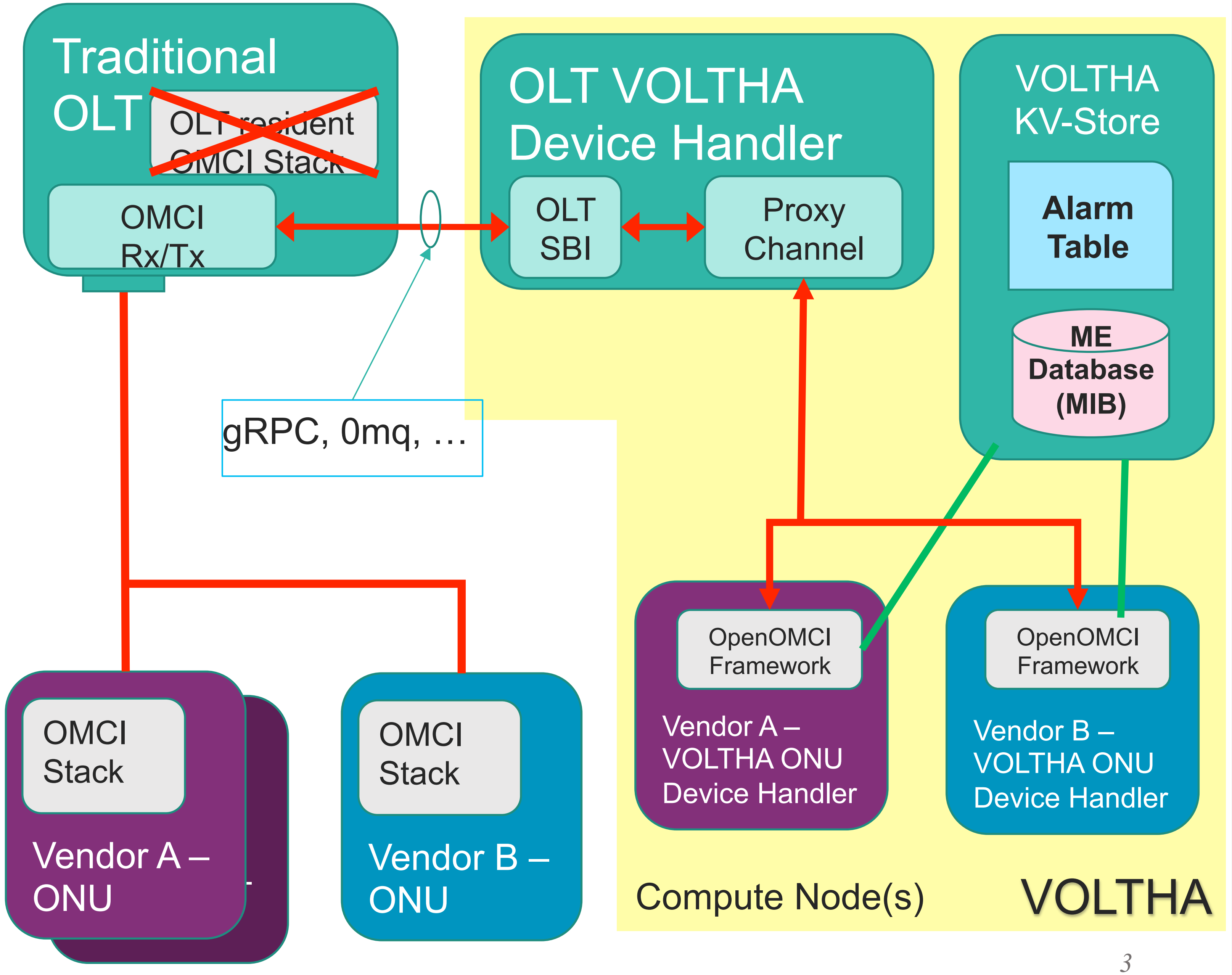
Traditional OLT & ONU

- OLT PNF responsible for:
  - running all applicable OMCI State Machines
  - Maintaining ME and Alarm Databases
  - Supporting any variations or ‘work-arounds needed for different vendors ONU’s OMCI stack implementations



VOLTHA OLT & ONU

- OLT PNF and VOLTHA OLT Device Handler responsible for:
  - OMCI Message Proxy (no modifications/monitoring of OMCI frame)
- ONU Device Handler responsible for:
  - Starting OpenOMCI
  - Provide per state-machine/OMCI Task modifications specific to Vendor's ONU



Basic Framework Concept

Various ITU documents (G.988, G.984, …) define the standards on how to implement OMCI on both an OLT and an ONU as well as services built upon the OMCI Managed Entities.

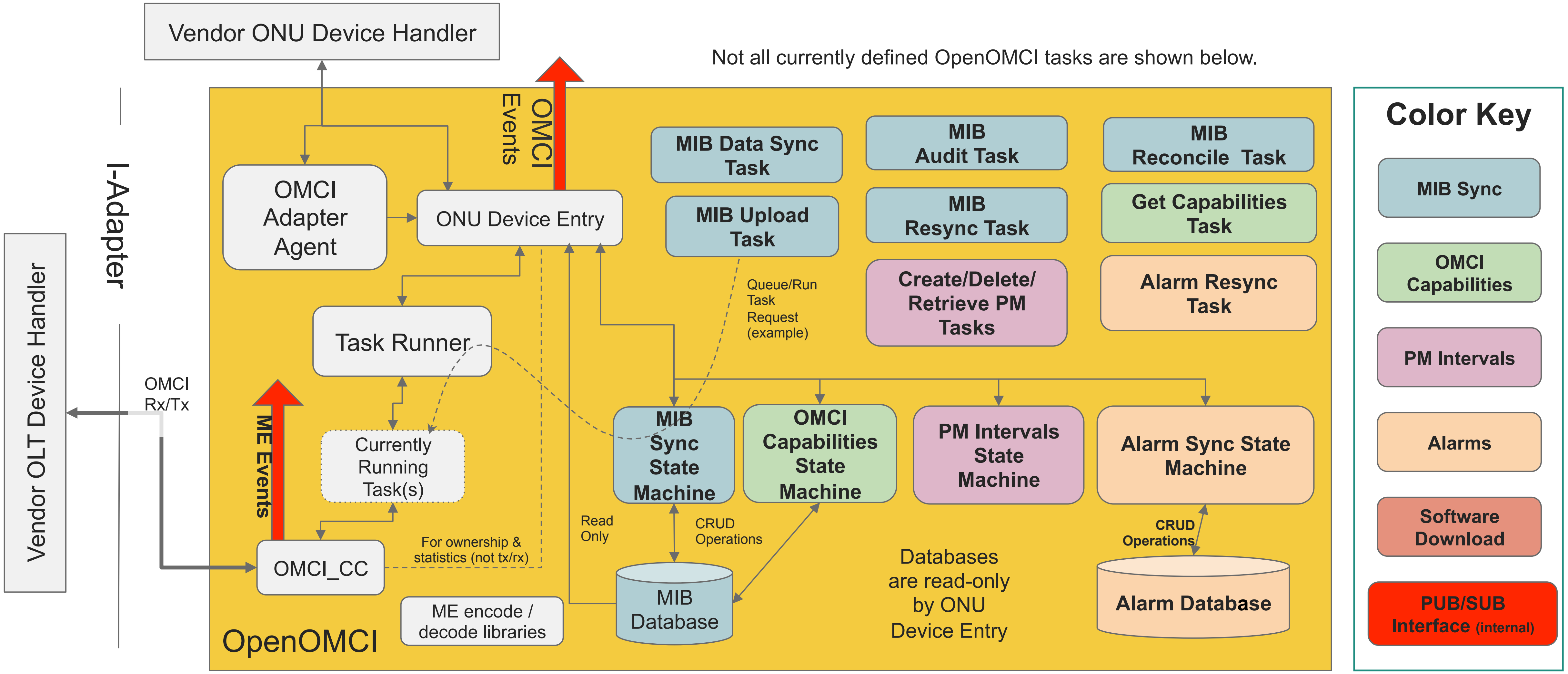
These Standards and Implementation Notes can be easily broken down into a set of specific State Machines and a series of steps (Tasks) that are ran at various points in time to provide some desired service.

Since vendors may interpret the standards differently, implement them with unintentional bugs, or provide more efficient ways to perform the same tasks, it would be advantageous to have a base set of classes that implement these State Machines and Tasks so that per-vendor ONU customizations can be taken advantage of.

OpenOMCI Framework (v2.0)

BRINGING THE WORLD TOGETHER

Not all currently defined OpenOMCI tasks are shown below.



General Business

Ways to customize OpenOMCI

A vendor’s ONU device handler can customize the OpenOMCI Framework in the following ways;

- Provide vendor-specific Managed Entity definitions to allow for OMCI frame decode and serialization
- Complete or Partial State Machine Customization: By providing either a new State Machine or creating a derived State Machine that overrides specific methods and/or states within the base class definition.
- Complete or Partial OMCI Task Customization: By providing either a new OMCI Task or creating a derived Task that overrides specific methods and/or states within the base class definition.

Examples of each can be found in both the Adtran and BroadCom ONU Device Adapters. (link to each provide at final slide of this presentation)

What is currently available

ADTRAN®
BRINGING THE WORLD TOGETHER

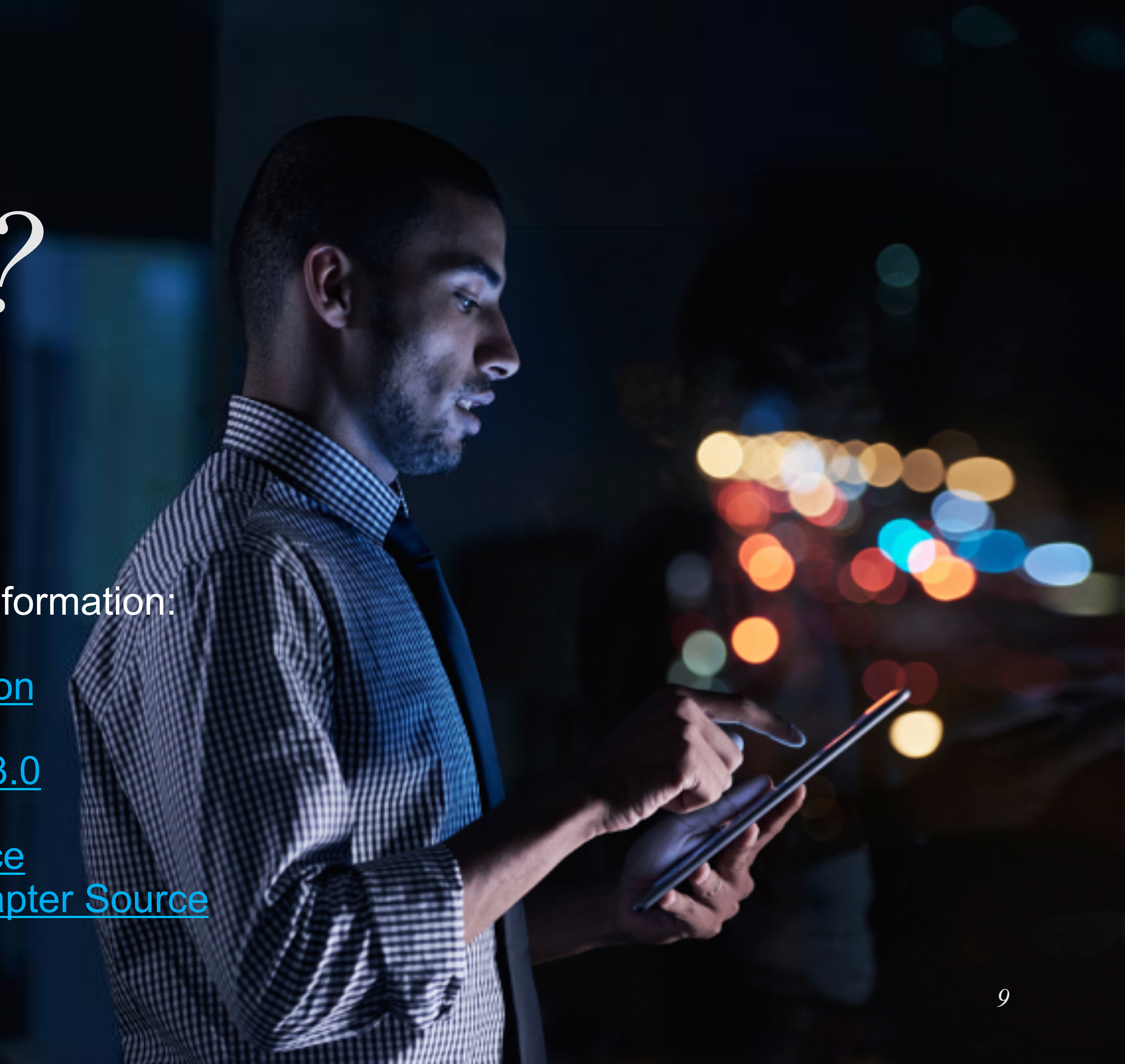
- OMCI State Machines
  - MIB Synchronization
  - Alarm Synchronization
  - OMCI Capabilities Discovery
  - Performance Monitoring/15-minute intervals'
  - Software Download and Image Activation
- 15 Pre-defined OMCI Tasks used by the above state machines
- OMCI-CC (Communications Channel)
- OMCI Task Runner (w/ watchdog support)
- VOLTHA CLI Commands to display Managed Entity Information
- Support on the following XGS-PON ONUs as of today's conference
  - ADTRAN
  - Alpha
  - Arcadyn
  - BroadCom
  - CIG
  - Iskratel
  - Nokia
  - T & W
  - TellLabs

Missing OMCI Capabilities in V2.0

ADTRAN®
BRINGING THE WORLD TOGETHER

The following features common to many OLT-resident OMCI stacks are not yet supported:

- Extended Message Format (currently only Baseline format supported)
- Test Results Support
- Power Shedding Support



Questions ?

Some useful links to further OpenOMCI information:

- VOLTHA OpenOMCI Documentation
- AT&T OpenOMCI Specification, V3.0
- Adtran ONU Device Adapter Source
- BroadCom OpenOMCI Device Adapter Source