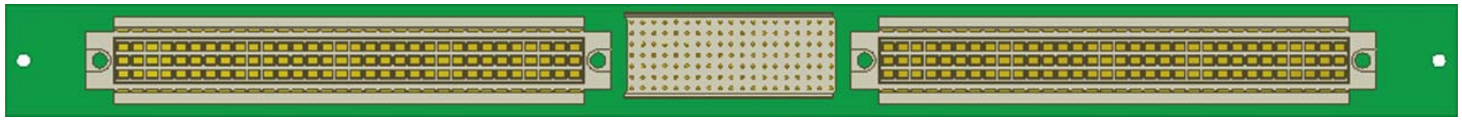
Architectures for High-Performance Embedded Computing

Robert Cooper — Mercury Computer Systems
Mark Littlefield — Curtiss-Wright Controls

What is OpenVPX?

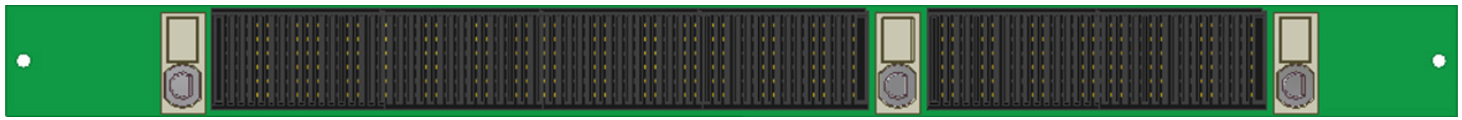
- Promotes standard components, interoperability, accelerated development and deployment
  - Defines a set of system specifications
- VITA 46 / VPX – a board form-factor standard intended as a VME/CPCI follow-on
  - Dense, compact, rugged form factor
  - Abundant backplane I/O
  - Highly scalable, highly flexible
  - Introduces 2-level maintenance through VITA 48/VPX-REDI
- Broad industry participation
  - Vendors, integrators, customers
- Wide applicability in military, aerospace and commercial
  - Multi-INT, radar data exploitation, information dissemination
  - Avionics
  - Homeland security
  - Telecom and transport

VPX Upgrades All Slot Connectors



VME64

- VPX — replaces all VME connectors with multi-gig RT2 7-row

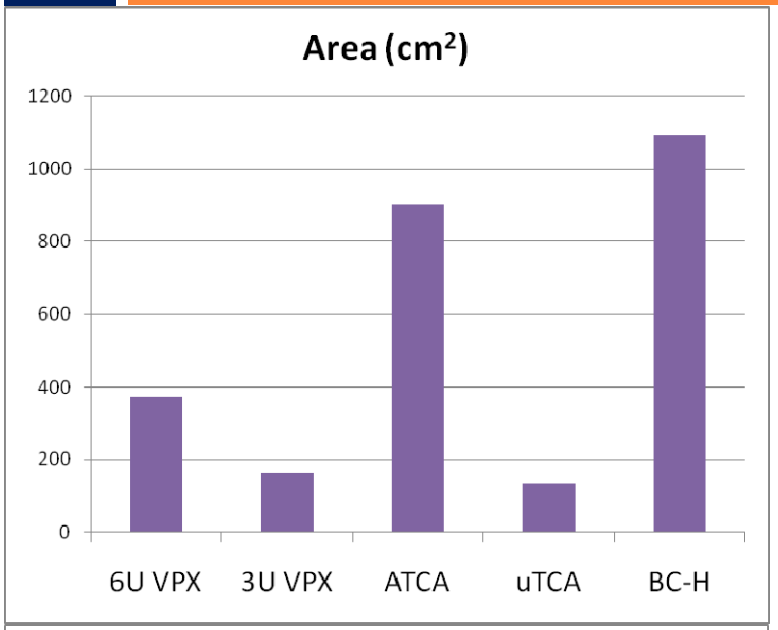


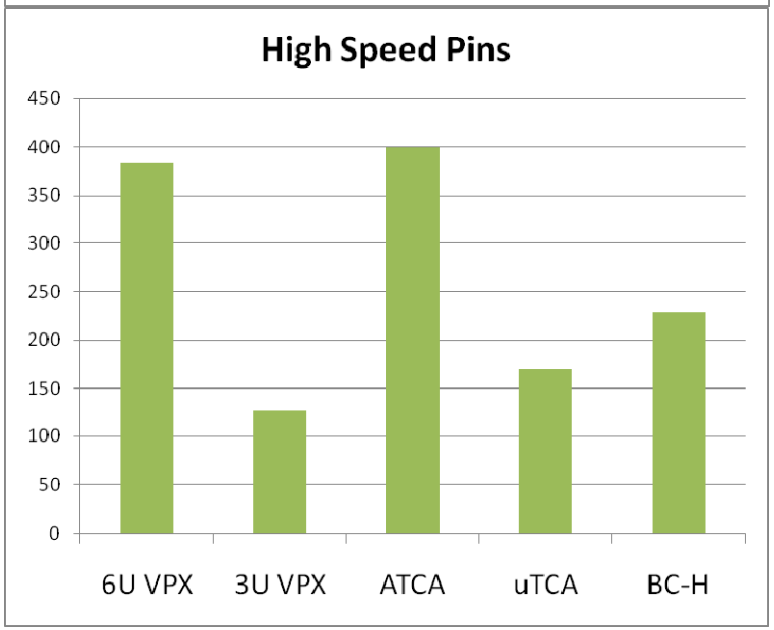
6U VPX

3U VPX

- Advantages
  - Enough high-speed pins (192 pairs) for switched fabric, Ethernet, & I/O
  - Allows huge amounts of rear I/O from the carrier and/or attached mezzanine cards when needed

VPX: Dense, Rugged, High Bandwidth



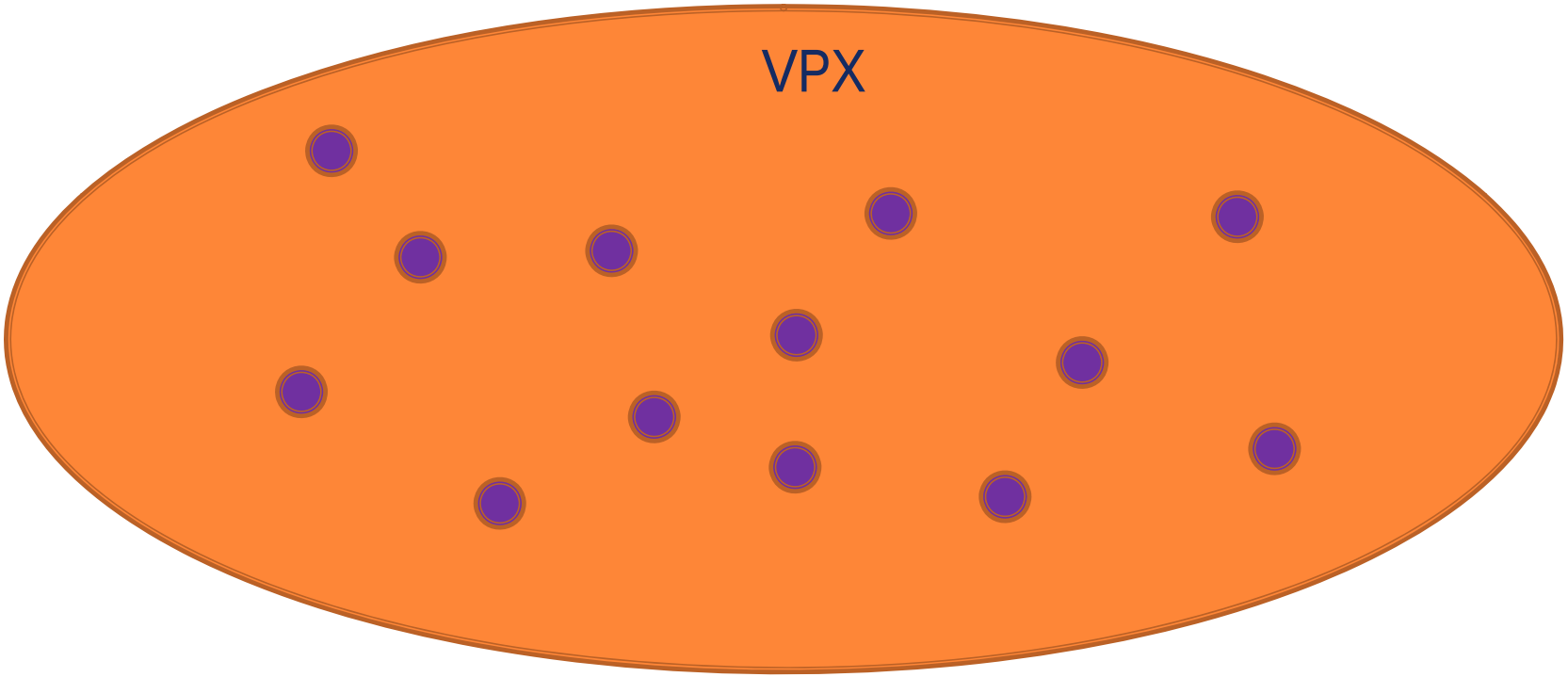


- Higher bandwidth density than ATCA™, Micro-TCA™ and BladeCenter™
  - Measured as # of high speed lanes* per board area
- Supports tougher environmental requirements
  - Temperature, shock and vibe more stringent than telecom standards (NEBS and GR-63-CORE)
- Supports module replacement in harsh environments
  - Two level maintenance

*Ignores ATCA Zone 3 (user I/O)
uTCA is Full Size Single Module (B+ connector)

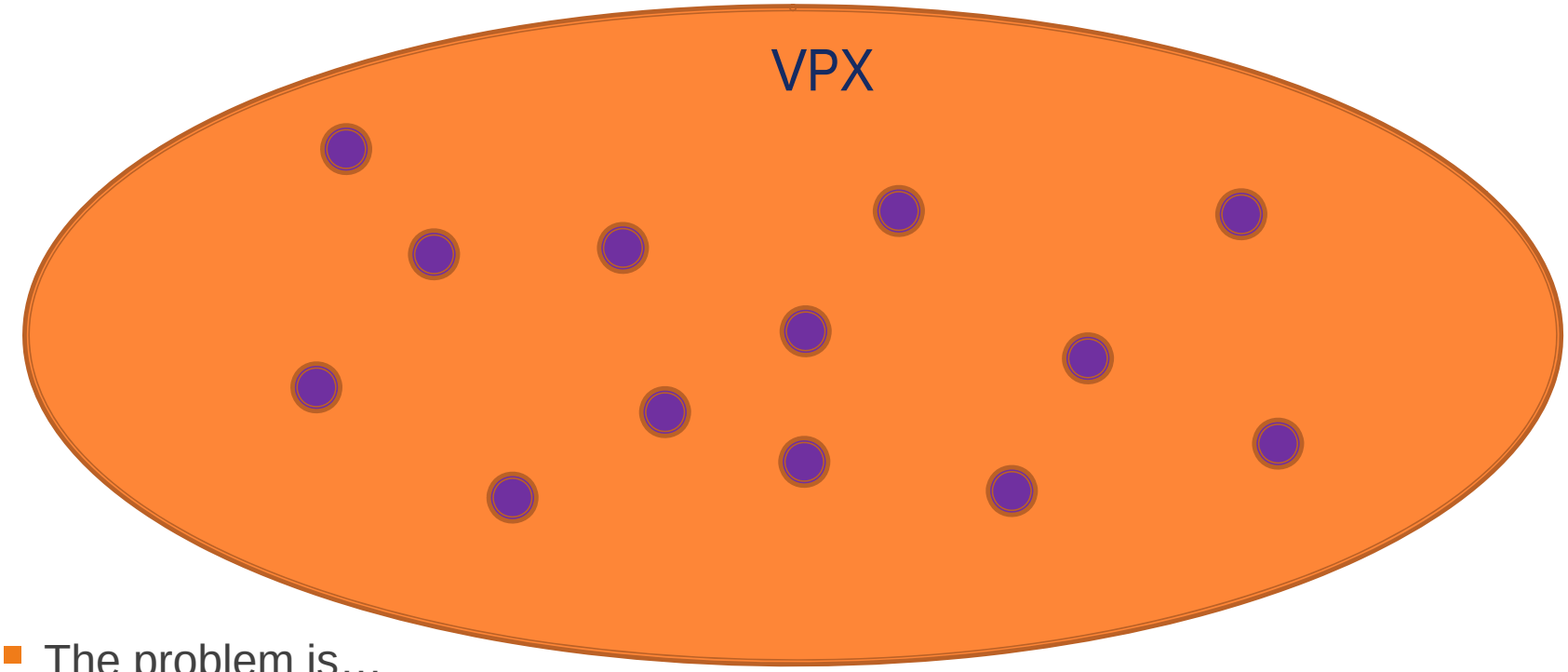
From VPX to OpenVPX

- VPX is a very large, flexible specification
  - It was designed that way to address many industry needs



From VPX to OpenVPX

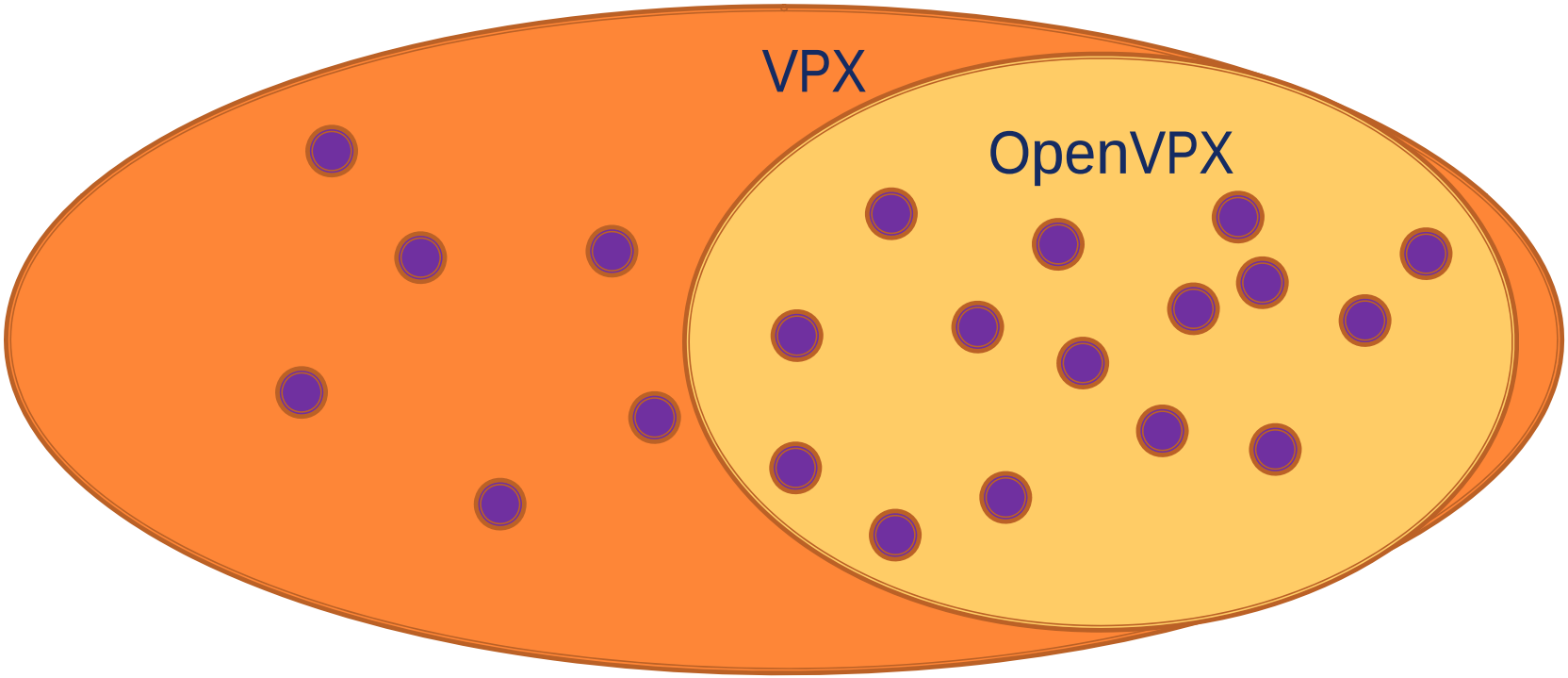
- VPX is a very large, flexible specification
  - It was designed that way to address many industry needs



- The problem is…
  - There are many possible implementations possible within the base and dot specifications
  - This leads to interoperability issues

From VPX to OpenVPX

- OpenVPX is a defined set of system implementations within VPX
  - Provides a framework for interoperability between modules and backplanes



- It is intended to be extensible
  - Includes existing implementation definitions
  - New profiles can be added over time as the industry evolves

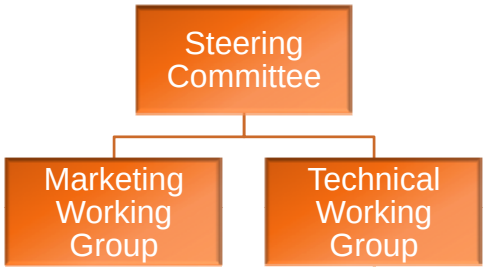
OpenVPX Scope and Priorities

- Specifies a set of *system* architectures
  - Not just a collection of pinout and protocol specifications
  - Guides system developers to choose one of a set of standard backplane and slot profiles
- Uses existing standards and drafts with minimal possible changes:
  - VPX (VITA-46)
  - REDI (VITA-48)
  - PMC / XMC (VITA-42)
- Rapidly delivers results into VITA Standards Organization
  - Urgency driven by critical programs needing system level VPX today
  - On target to contribute 1.0 Specification to VITA 65 by October 2009
  - VITA 65 to follow VSO process with goal to ratify as VITA / ANSI standard
  - Expect additional system profiles may be added over time as needed

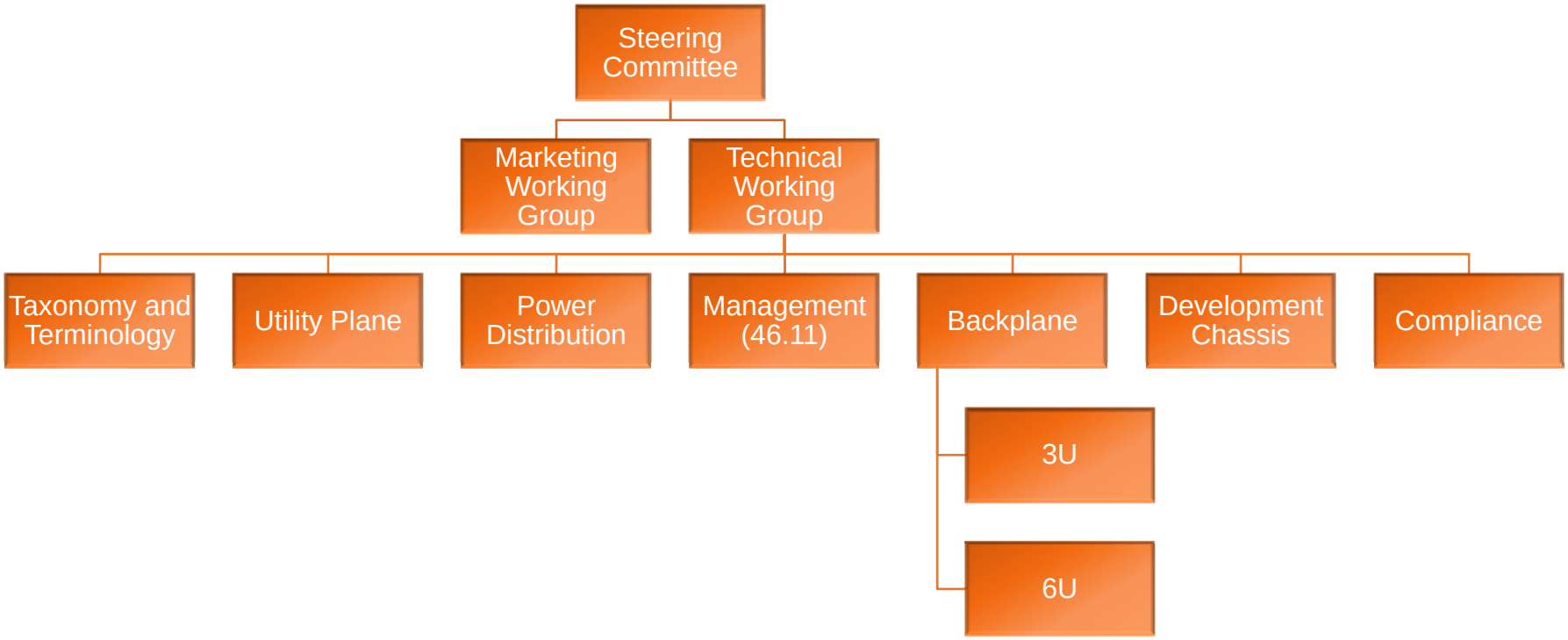
OpenVPX Members

- Aitech Defense Systems, Inc.
- Agilent Technologies Inc.
- BittWare, Inc.
- The Boeing Company
- Concurrent Technologies
- CSP Inc.
- Curtiss-Wright Controls, Inc.
- Diversified Technology, Inc.
- DRS Signal Solutions, Inc.
- Elma Electronic, Inc.
- Extreme Engineering Solutions (X-ES)
- Foxconn Electronics, Inc.
- GE Fanuc Intelligent Platforms
- General Dynamics Advanced Information Systems
- General Dynamics Canada
- Hybricon Corp.
- Kontron Modular Systems S.A.S.
- Lockheed Martin Corporation
- Mercury Computer Systems, Inc.
- Molex, Inc.
- Northrop Grumman Electronic Systems
- Pentair Electronic Packaging / Schroff
- Pentek, Inc.
- Pigeon Point Systems
- SIE Computing Solutions
- TEK Microsystems, Inc.
- Tracewell Systems
- Tyco Electronics Corporation

OpenVPX Organization



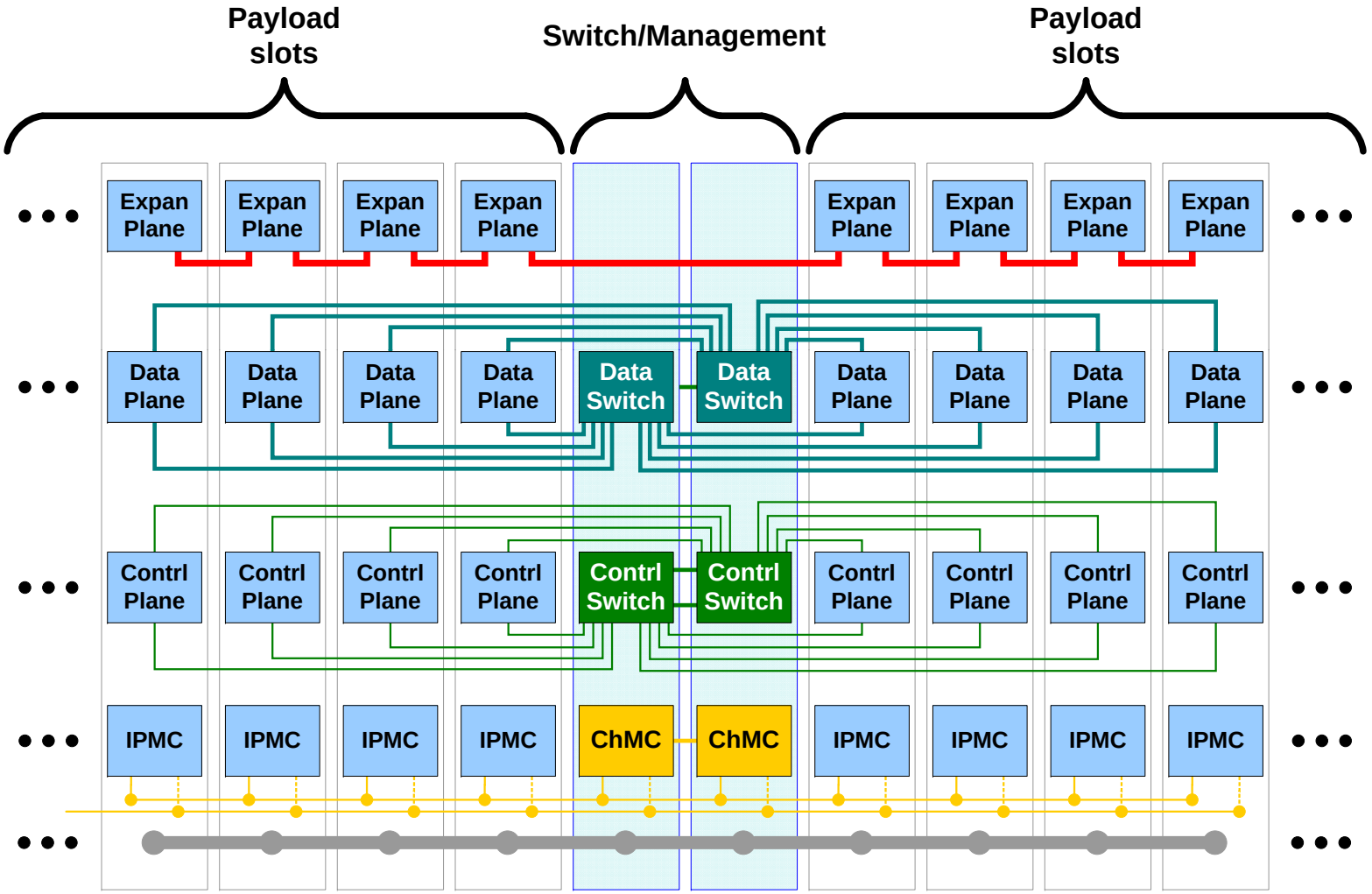
OpenVPX Organization



OpenVPX Specification

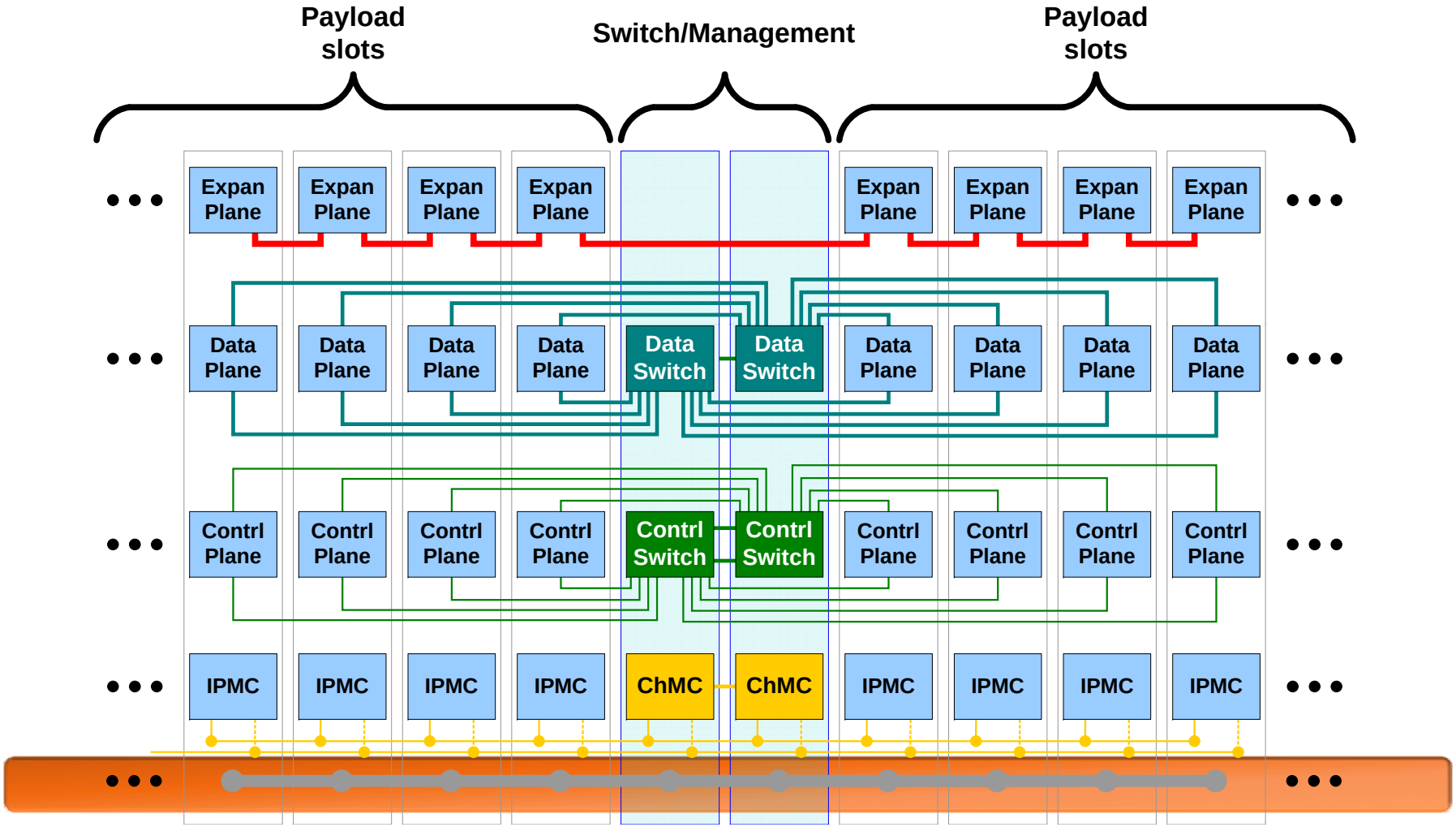
- Planes
- Pipes
- Profiles

Multiple Planes



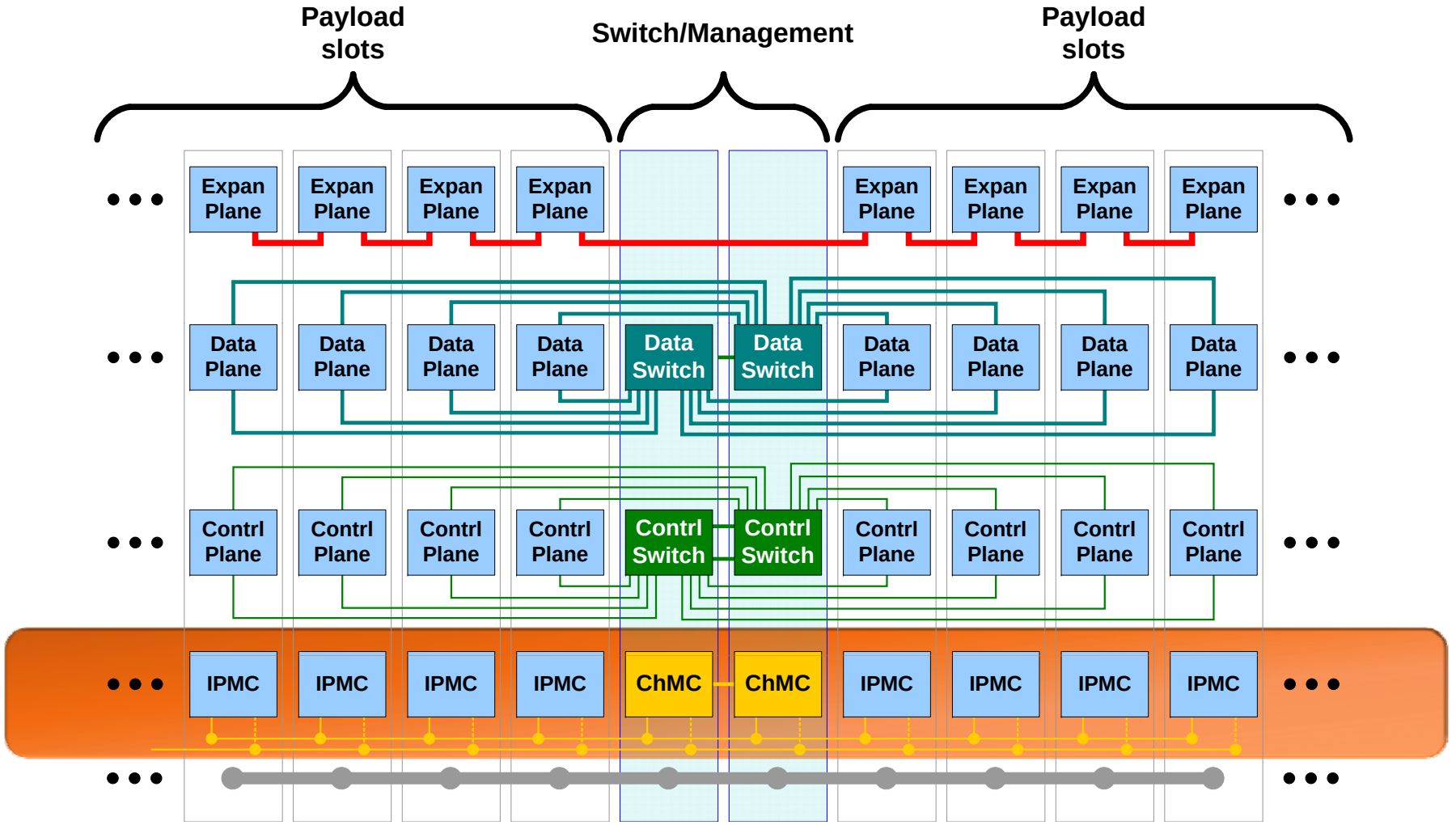
- Some OpenVPX system architectures utilize multiple planes to isolate traffic with different characteristics and requirements

Utility Plane



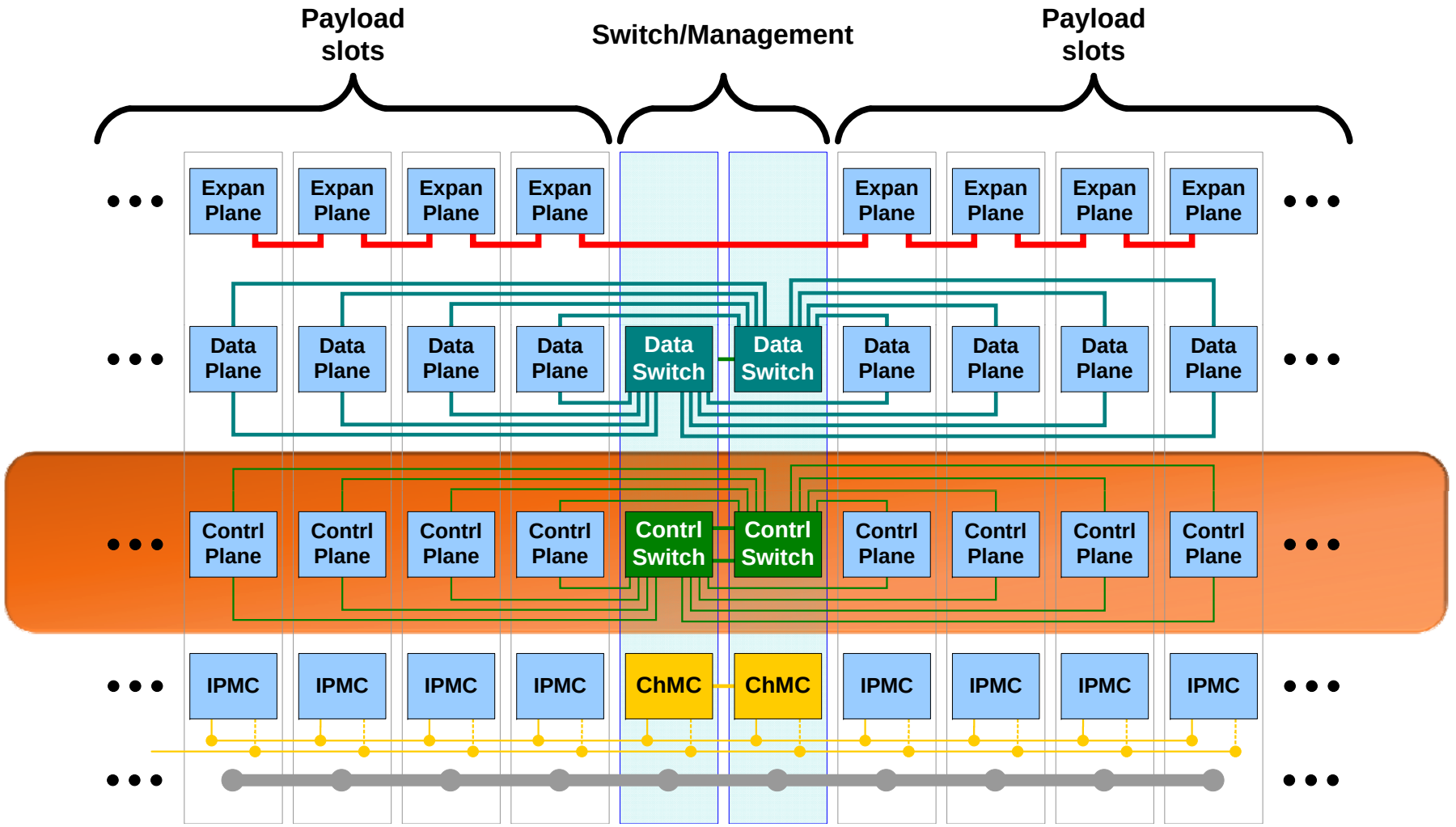
- Power pins and various utility signals
  - NVMRO, SYS_CLK (MBSC), REF_CLK & AUX_CLK (new), resets (including “maskable reset”)

Management Plane



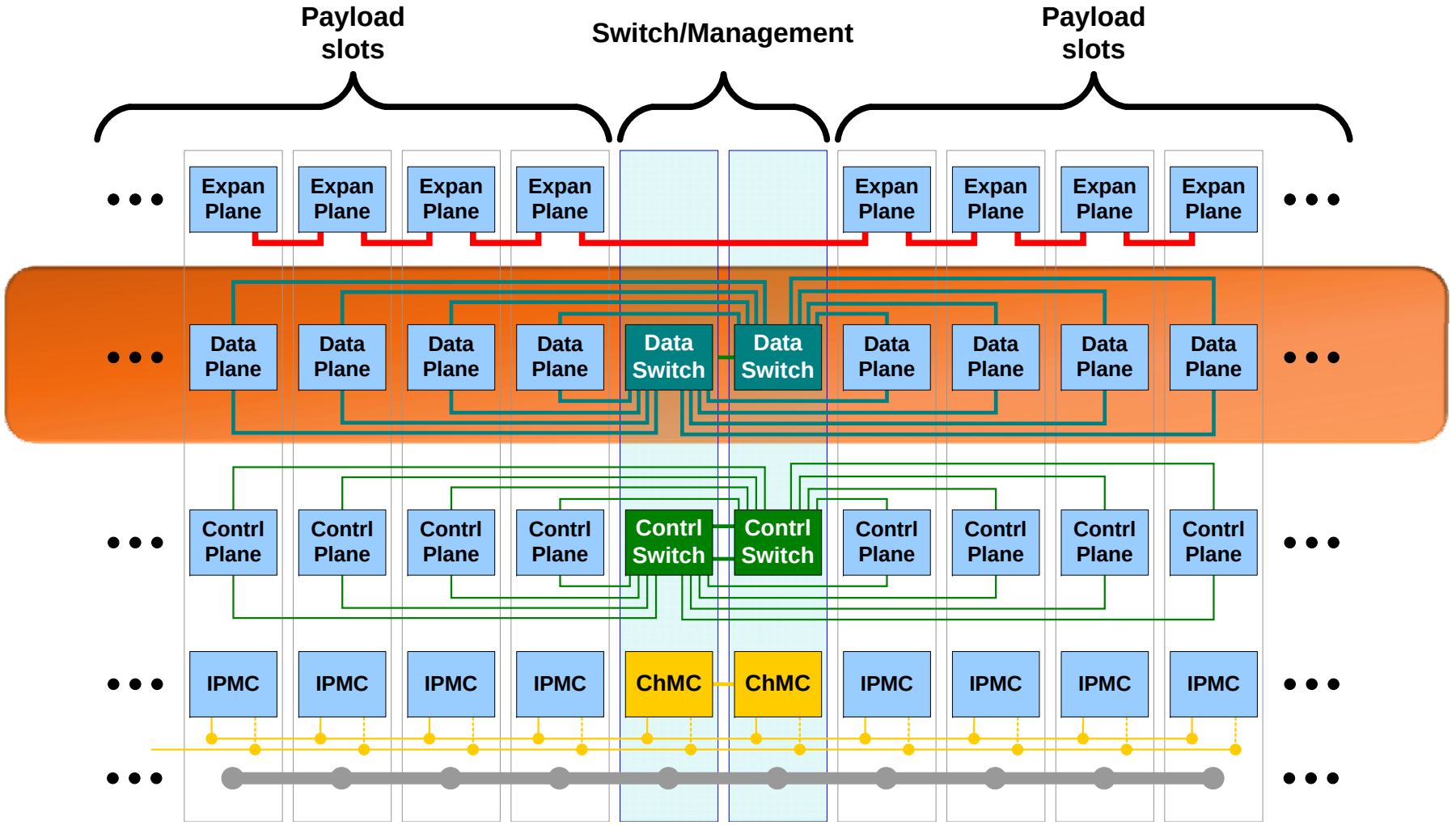
- Low-power
- Defined by VITA 46.0 and 46.11
- Prognosticates/diagnoses problems
- Can control module power

Control Plane



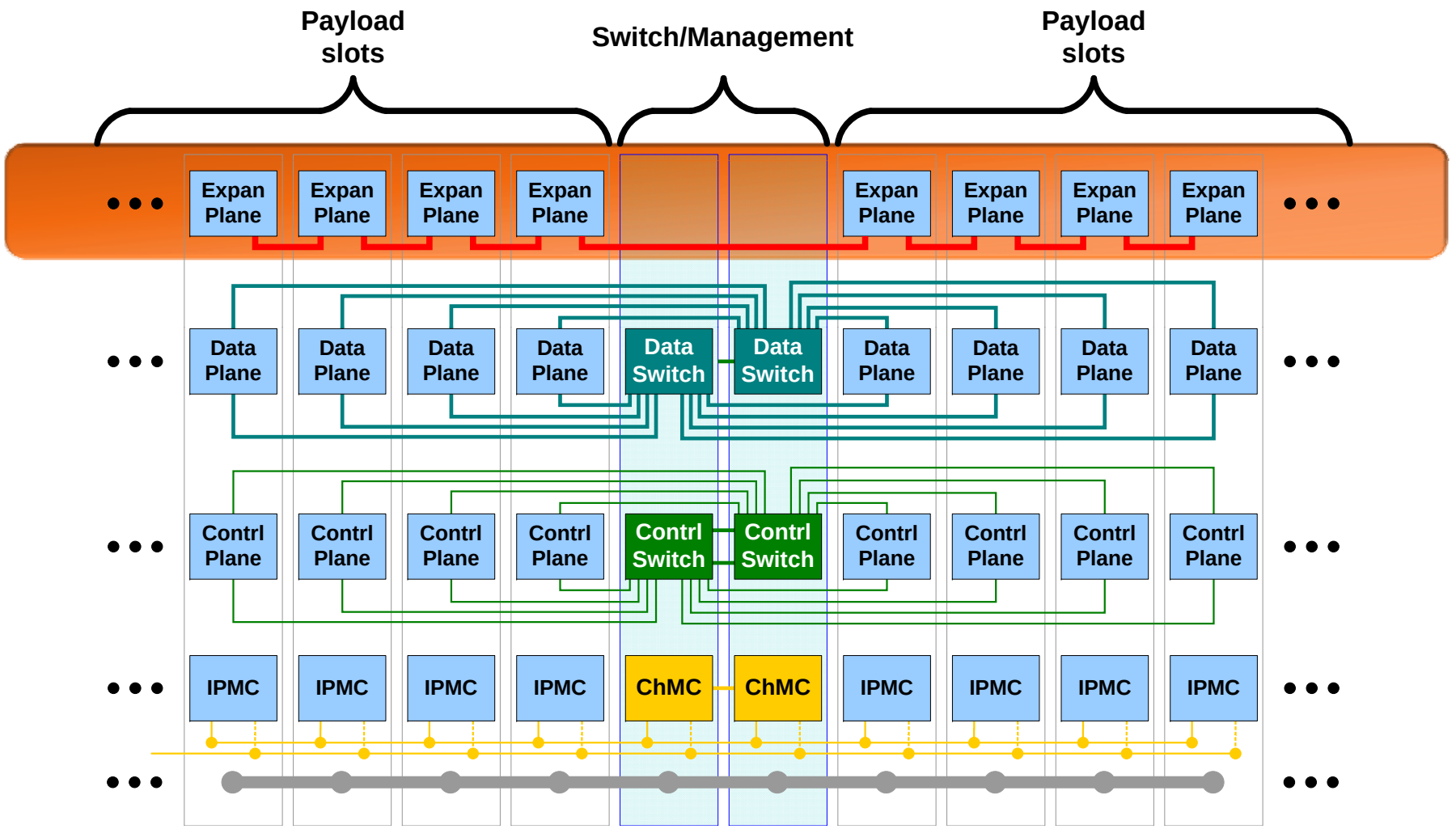
- Reliable, packet-based communication for application control, exploitation data
- Typically Gigabit Ethernet

Data Plane



- High-throughput, predictable data movement without interfering with other traffic
- Examples: Serial RapidIO or PCI Express

Expansion Plane



- Tightly coupled groups of boards and I/O
- Typically VME bridging or PCI Express

Pipes

- Pipe: A collection of differential pairs assigned to a plane or other functions
  - Used by slot profiles
  - Does not specify what protocol is used on it (module profiles do that)

	Differential Pairs	Example Protocols
Fat Pipe (FP)	8	4x sRIO x4 PCIe 10GBase-BX4 10GBase-KX4
Thin Pipe (TP)	4	2x sRIO x2 PCIe 1000Base-T
Ultra Thin Pipe (UTP)	2	1x sRIO x1 PCIe 1000Base-BX

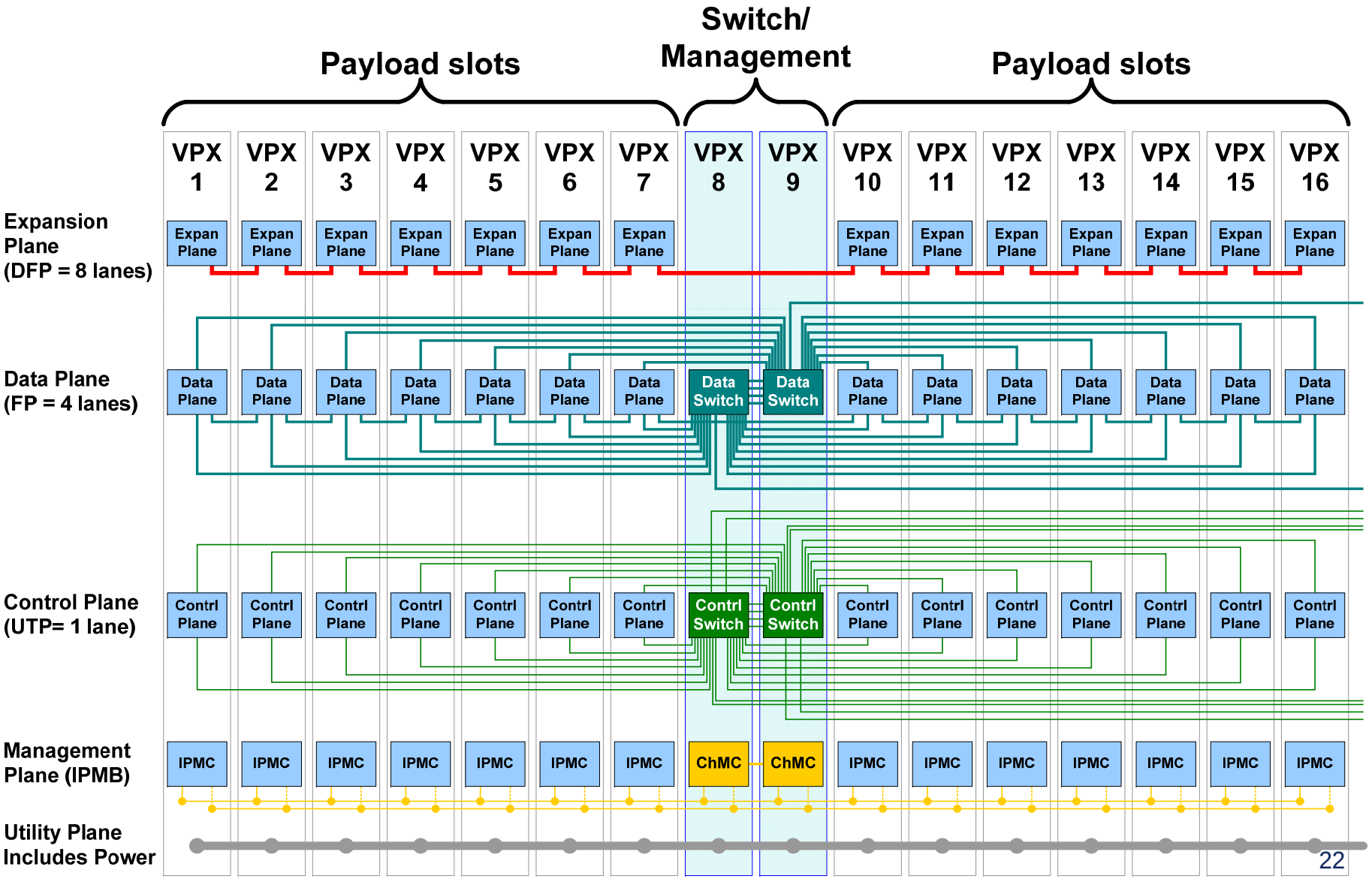
Profiles

- The specification uses *profiles* for structure and hierarchy in the specification
- Slot Profile
  - A physical mapping of ports onto a slot's backplane connectors
  - Uses notions of *pipes* and *planes*
  - Does not specify actual protocols conveyed over the backplane
- Backplane Profile
  - A physical specification of a backplane
  - Specifies the number and type of slot profiles
  - Defines the topology of channels and buses that interconnect the slots
- Module Profile
  - Extends a slot profile by mapping protocols to a module's ports
  - Includes thermal, power and mechanical requirements
  - Provides a first order check of compatibility between modules

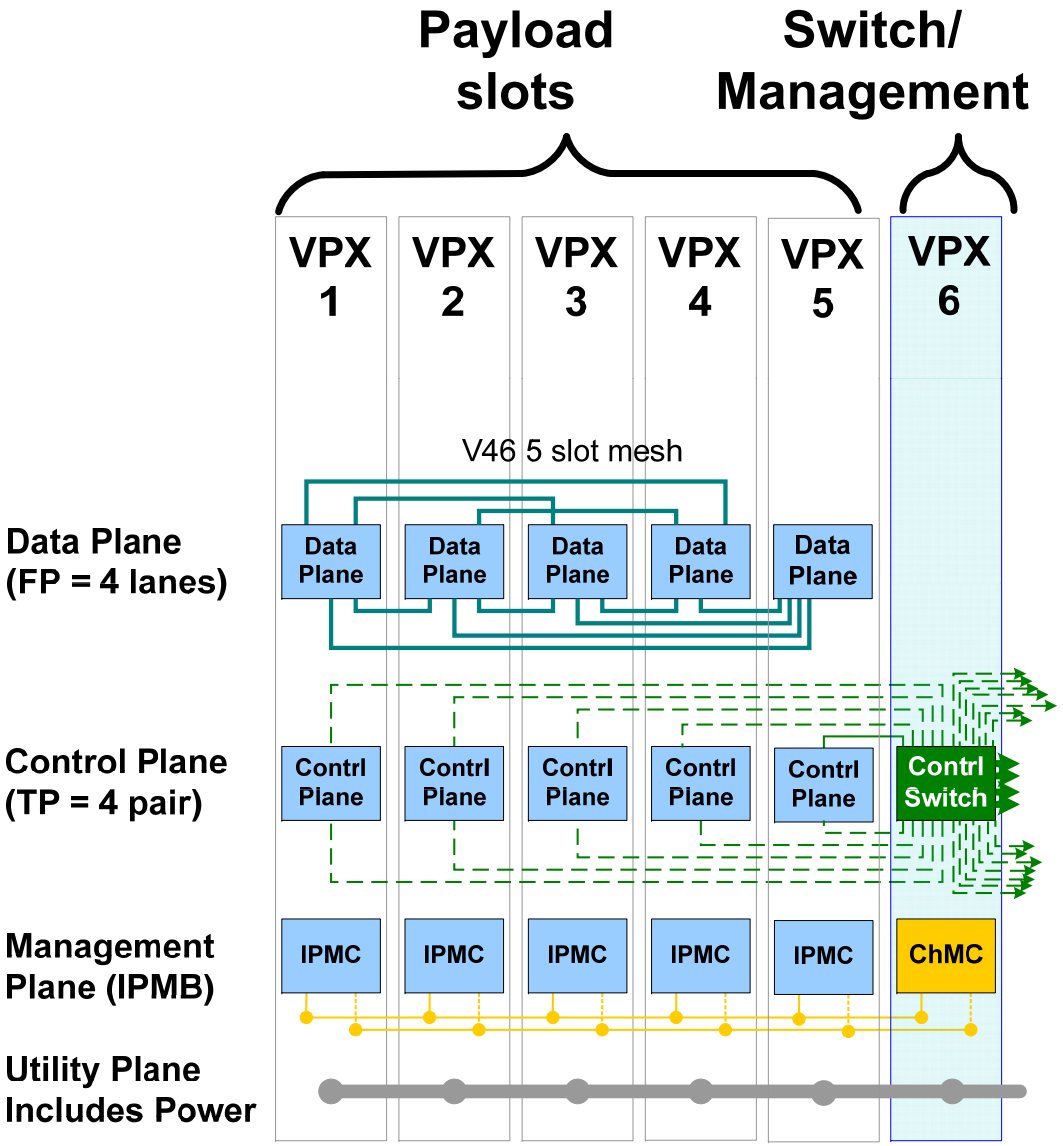
Backplane Topology Types

- Centralized switching
  - A set of peer payload boards connected by a switch fabric boards
  - Single or dual star topology for multiple path routing and potential redundancy
  - Also provides system management function
- Distributed switching
  - A set of peer payload cards connected in a full or partial mesh
  - Useful for small slot count systems as it avoids dedicated switch slots
  - Larger slot count systems require switching logic on each payload card
- Host / slave
  - Typically comprise a master host board with several slave boards linked by PCIe
  - Allows an SBC to have greatly expanded capabilities without complexity of a general switching fabric
- Some examples on the next few slides

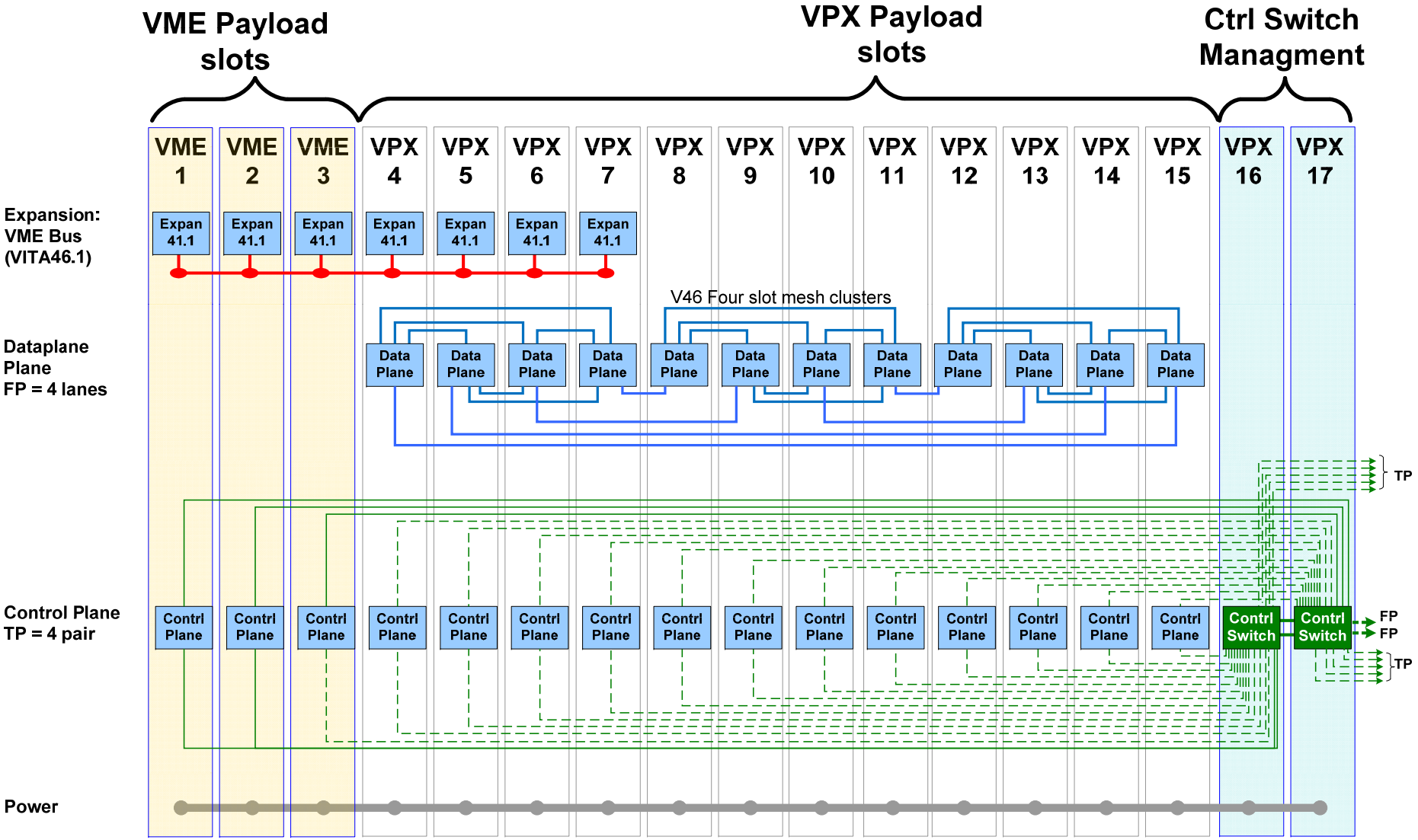
Centralized Switching Example (6U)



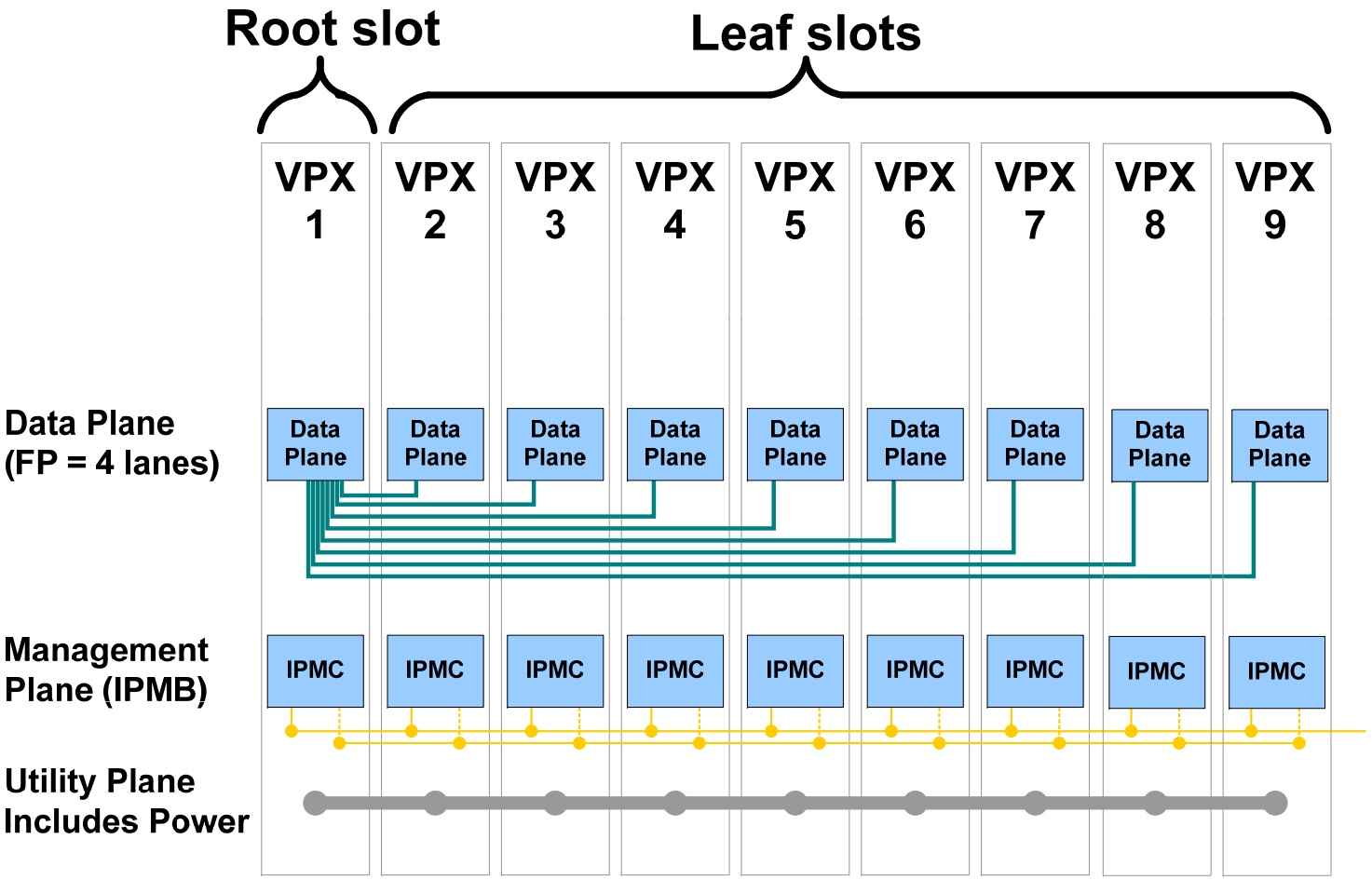
Distributed Switching Example (6U)



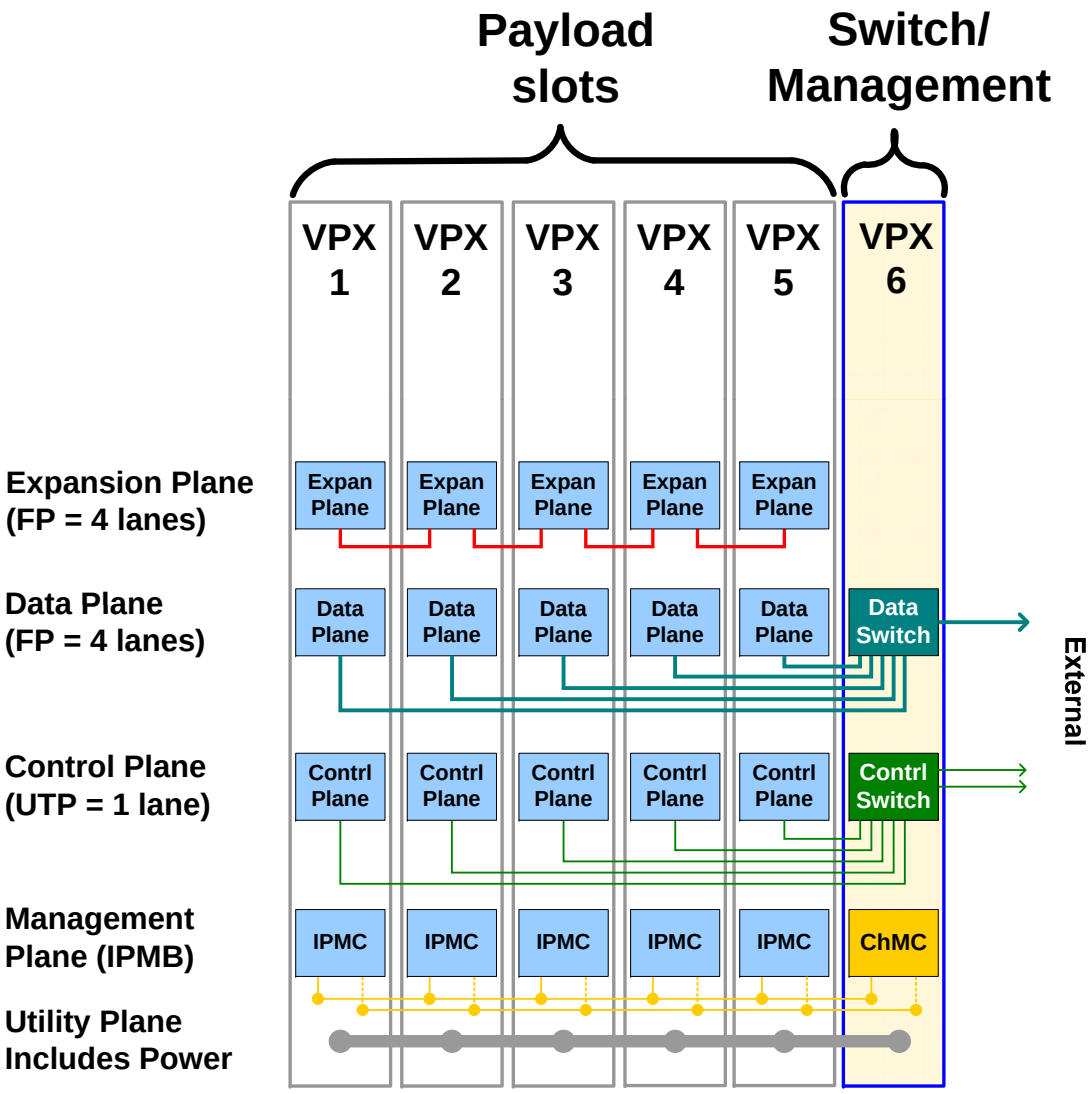
Hybrid VME / VPX Example (6U)



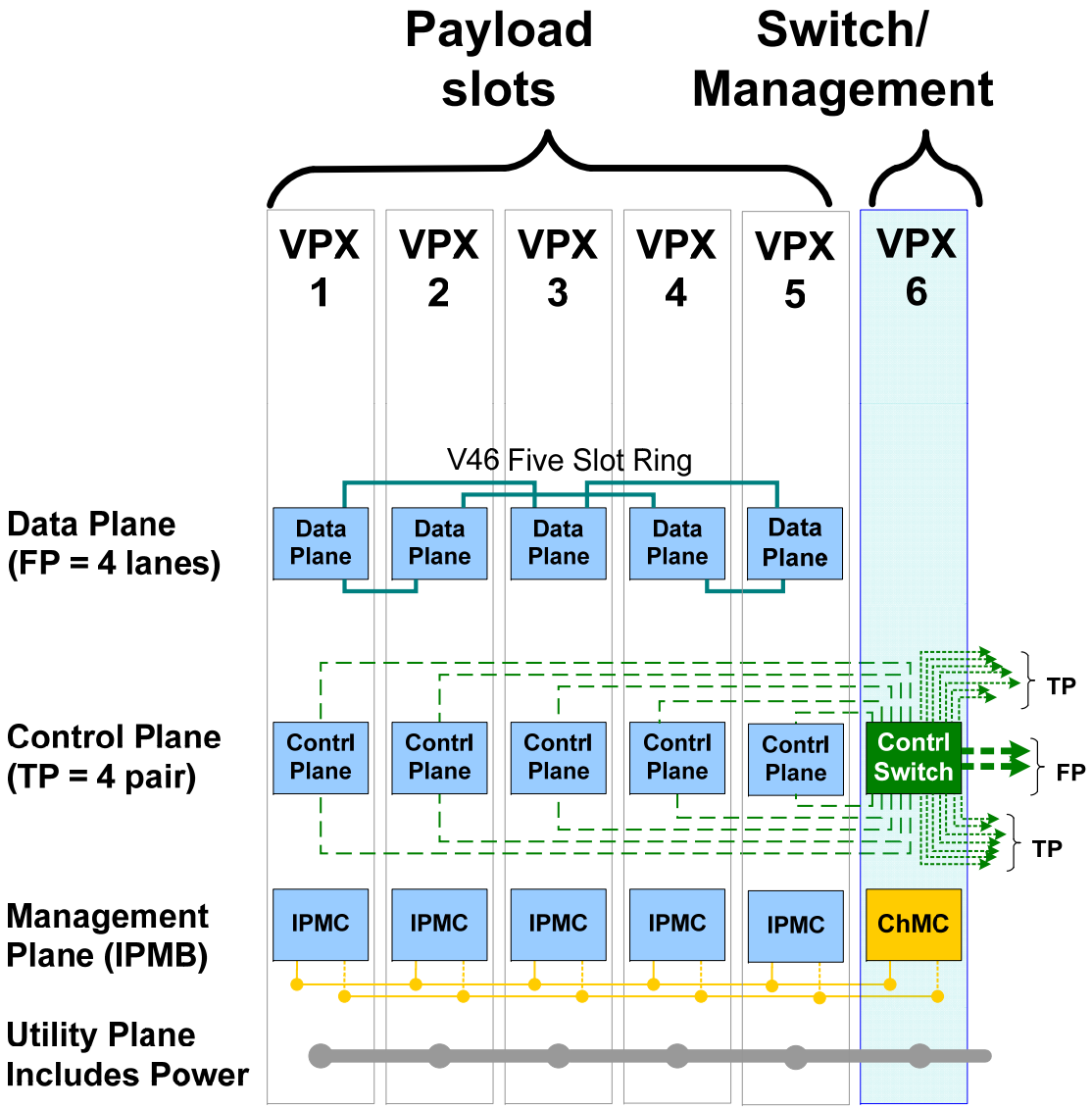
Host / Slave Example (6U)



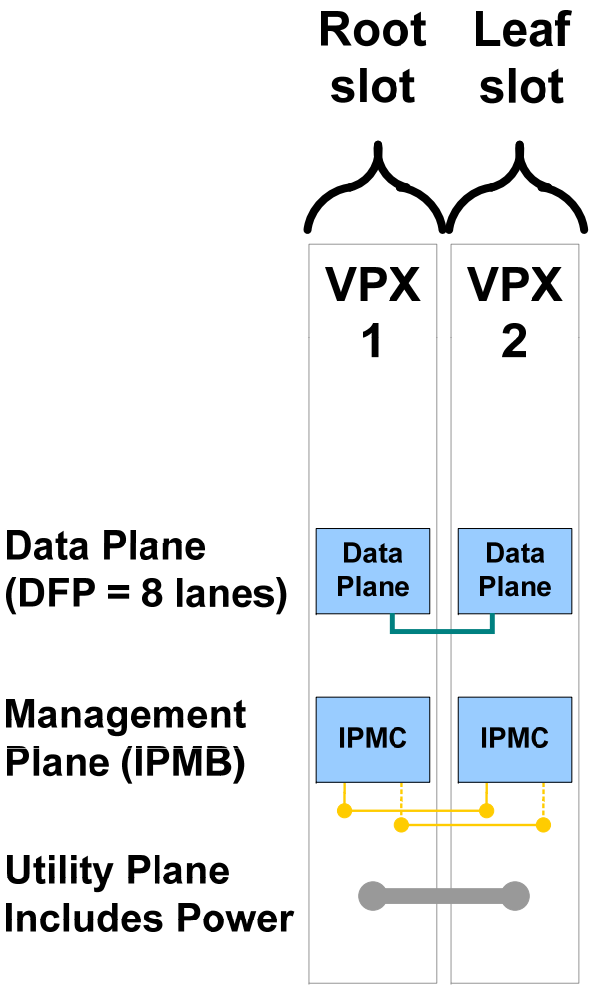
Centralized Switching Example (3U)

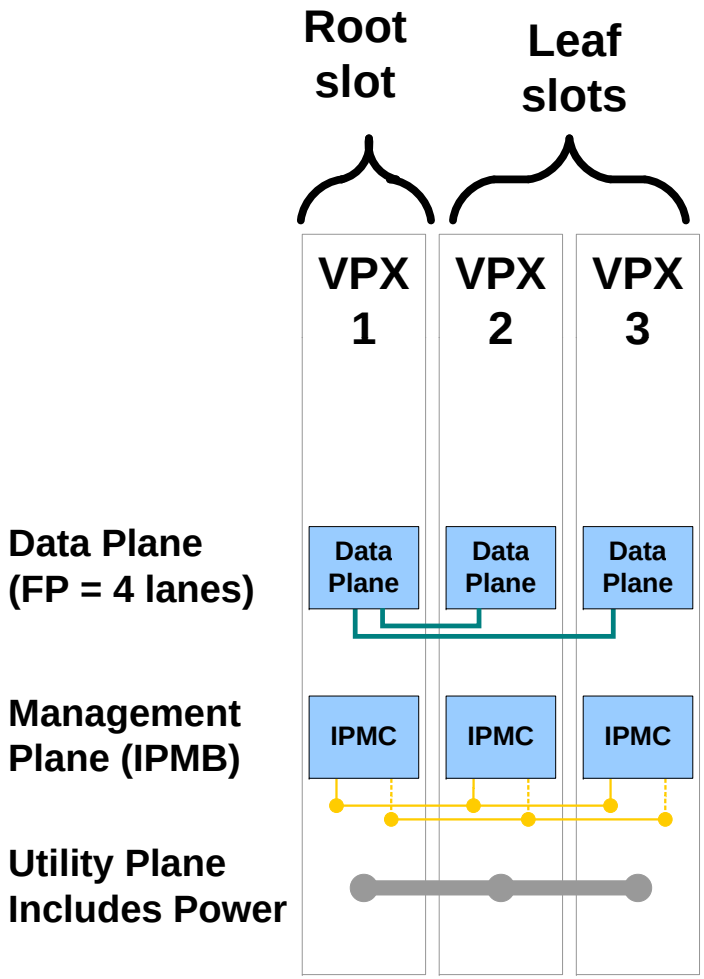


Distributed Switching Example (3U)



Host / Slave Examples(3U)





OpenVPX Is Not Specifying Everything

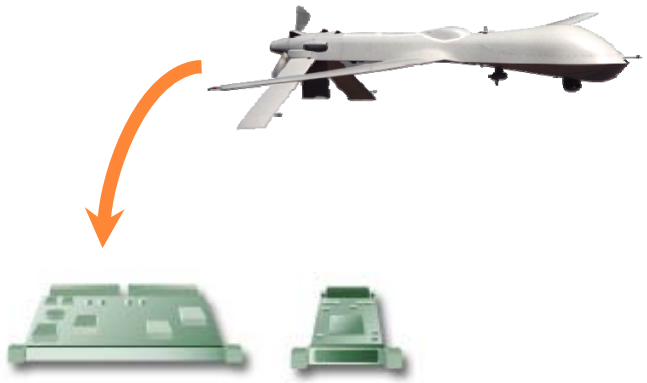
- User defined pins reserved in every slot profile
  - Provides for flexibility in handling I/O and custom board-to-board links
    - Historically, 6U VME provided lots of user I/O pins on P0 and P2
  - Limits full interoperability and interchangeability of OpenVPX compliant modules
    - Full plug-and-play is considered less critical than customer and vendor differentiation to meet critical application functional and SWaP requirements
- Module profiles do not fully specify interoperability above layers 1 and 2
  - E.g. fabric discovery, enumeration and routing choices not fully specified
  - These may be specified via later standards work
- Only development chassis are standardized
  - I/O provided via rear transition modules (RTMs)
  - Deployment scenarios typically use a custom backplane to deal with I/O in conduction cooled and other rugged packages

Typical OpenVPX Development Flow

- Determine application requirements
  - Size, weight and power
  - Processing, fabric and I/O requirements

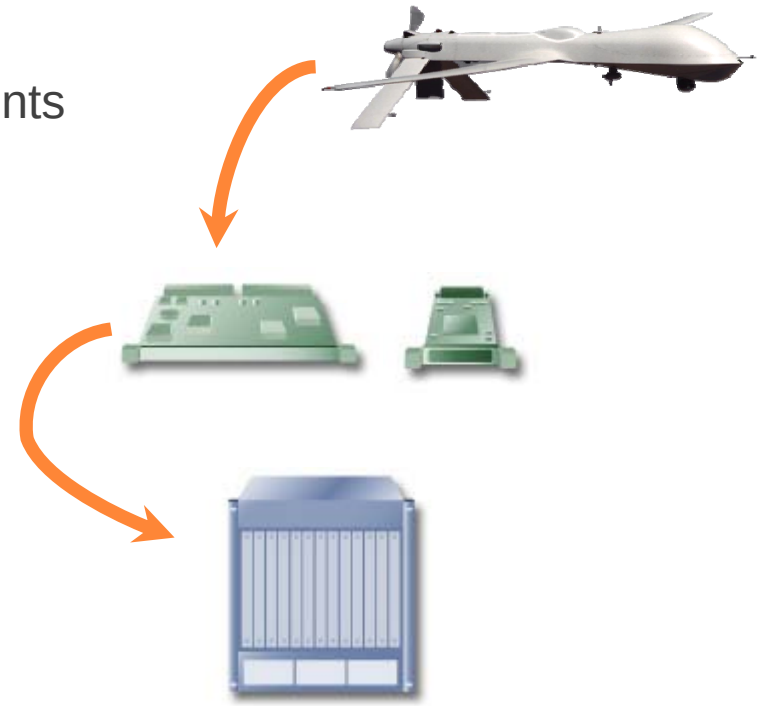
Typical OpenVPX Development Flow

- Determine application requirements
  - Size, weight and power
  - Processing, fabric and I/O requirements
- Select overall system parameters
  - 3U or 6U?
  - Switching topology?
  - Number and type of slots?



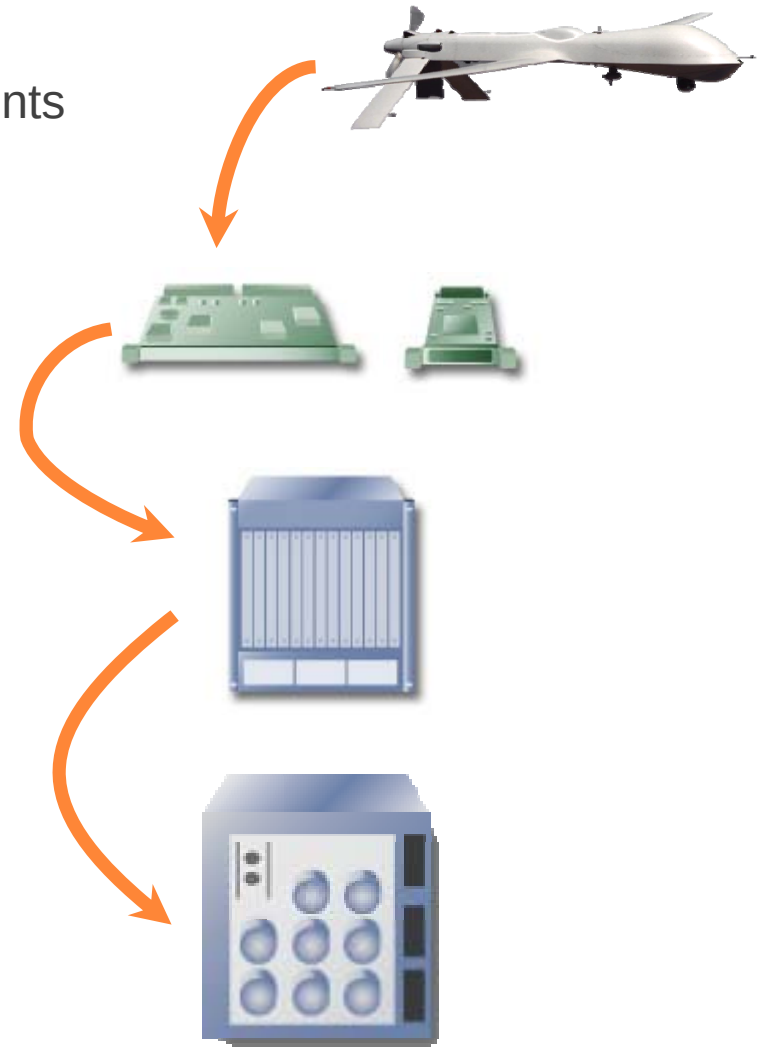
Typical OpenVPX Development Flow

- Determine application requirements
  - Size, weight and power
  - Processing, fabric and I/O requirements
- Select overall system parameters
  - 3U or 6U?
  - Switching topology?
  - Number and type of slots?
- Assemble development vehicle
  - COTS development chassis
  - COTS boards
  - COTS or custom RTMs



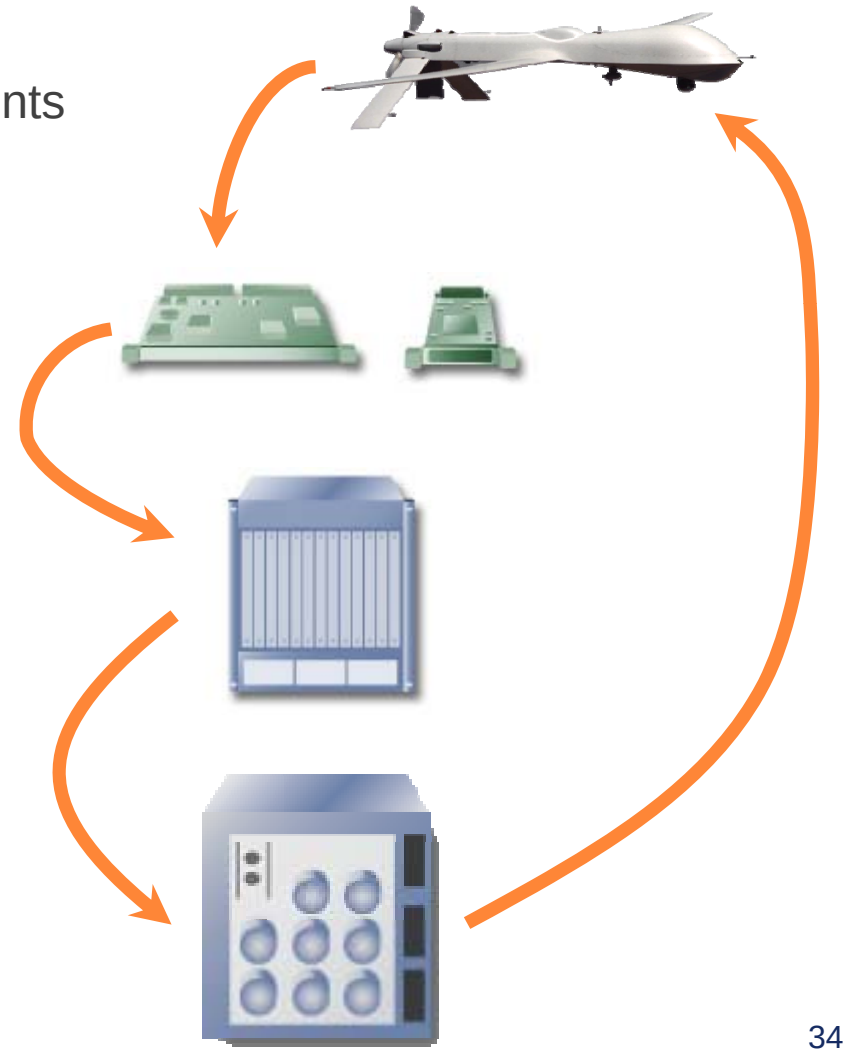
Typical OpenVPX Development Flow

- Determine application requirements
  - Size, weight and power
  - Processing, fabric and I/O requirements
- Select overall system parameters
  - 3U or 6U?
  - Switching topology?
  - Number and type of slots?
- Assemble development vehicle
  - COTS development chassis
  - COTS boards
  - COTS or custom RTMs
- Design deployment system
  - Typically custom backplane
  - Typically route I/O signals to custom I/O slot or bulkhead connector



Typical OpenVPX Development Flow

- Determine application requirements
  - Size, weight and power
  - Processing, fabric and I/O requirements
- Select overall system parameters
  - 3U or 6U?
  - Switching topology?
  - Number and type of slots?
- Assemble development vehicle
  - COTS development chassis
  - COTS boards
  - COTS or custom RTMs
- Design deployment system
  - Typically custom backplane
  - Typically route I/O signals to custom I/O slot or bulkhead connector



OpenVPX Benefits

- Promotes interoperability and vendor choice
- Provides specific design profiles that vendors can design to and integrators can specify as requirements
- Reduces integration issues resulting in faster development & deployment time
- Higher board volumes ➔ Economies of scale
- Industry leading bandwidth and density
- Higher velocity of technology upgrades
- Will support higher backplane signaling speeds as technology matures

Questions?