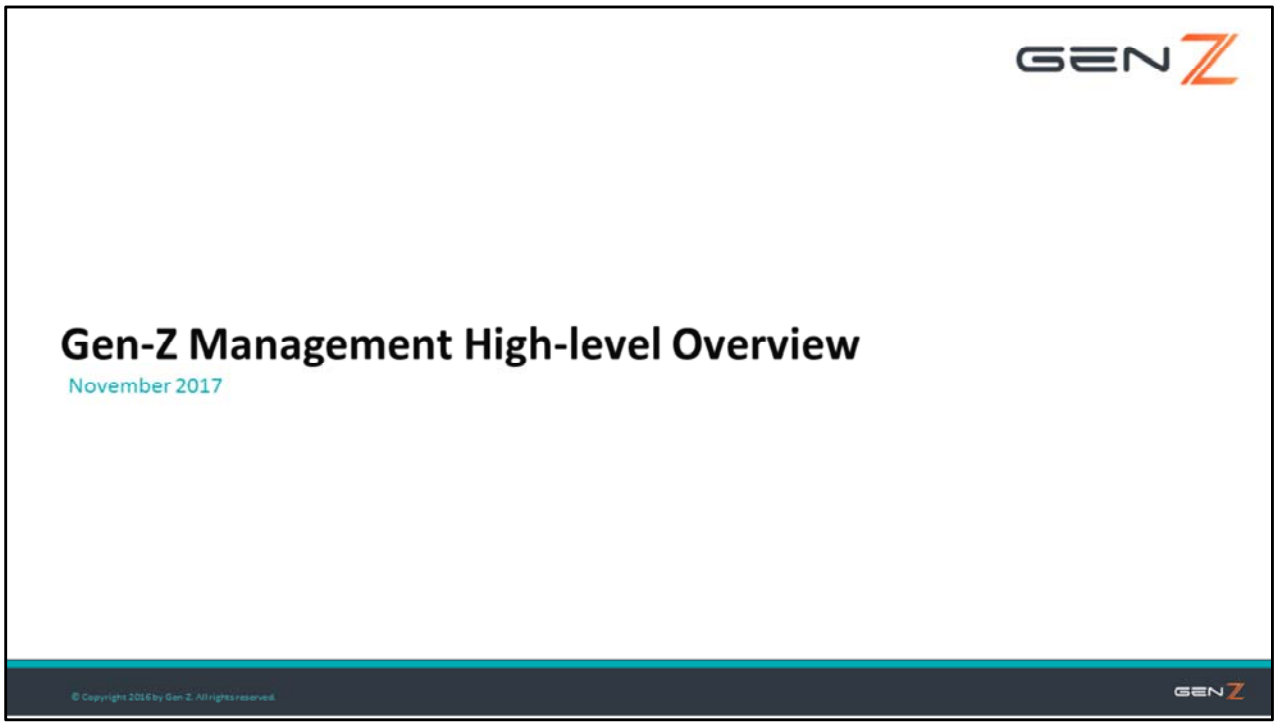



This presentation provides a high-level overview of Gen-Z Management.

# Disclaimer

This document is provided 'as is' with no warranties whatsoever, including any warranty of merchantability, noninfringement, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Gen-Z Consortium disclaims all liability for infringement of proprietary rights, relating to use of information in this document. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

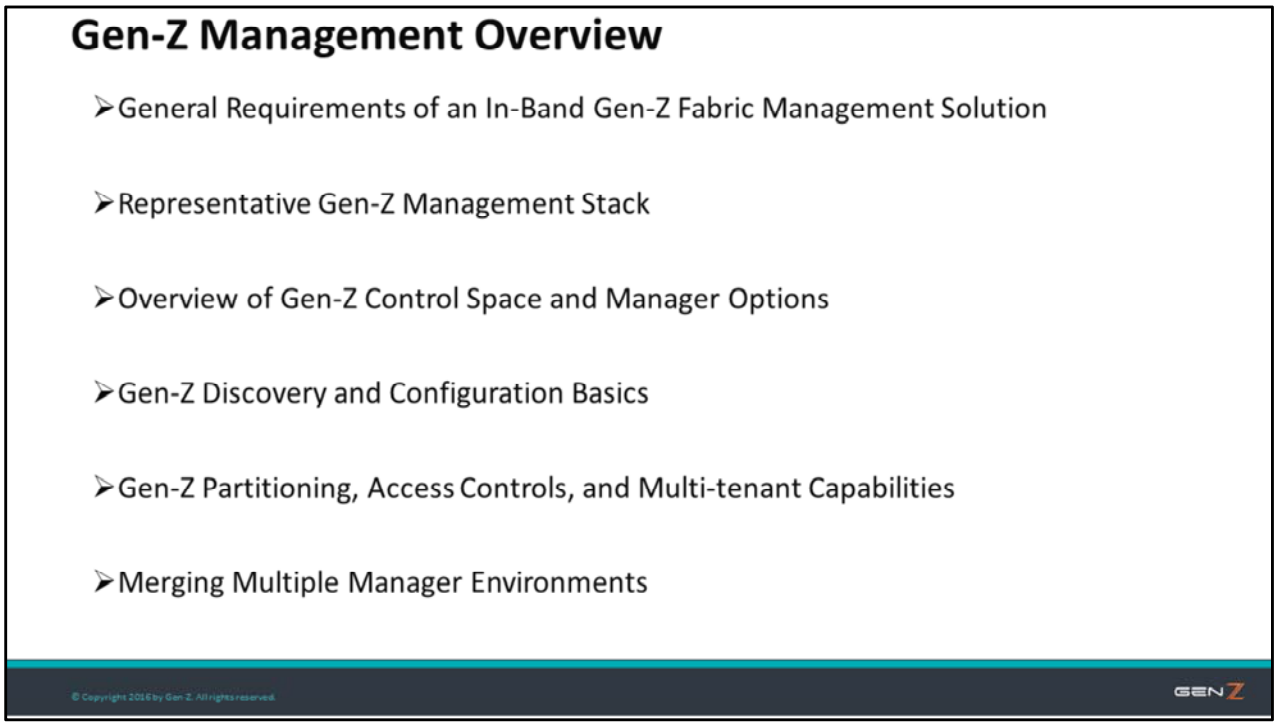
Gen-Z is a trademark or registered trademark of the Gen-Z Consortium.

All other product names are trademarks, registered trademarks, or servicemarks of their respective owners.

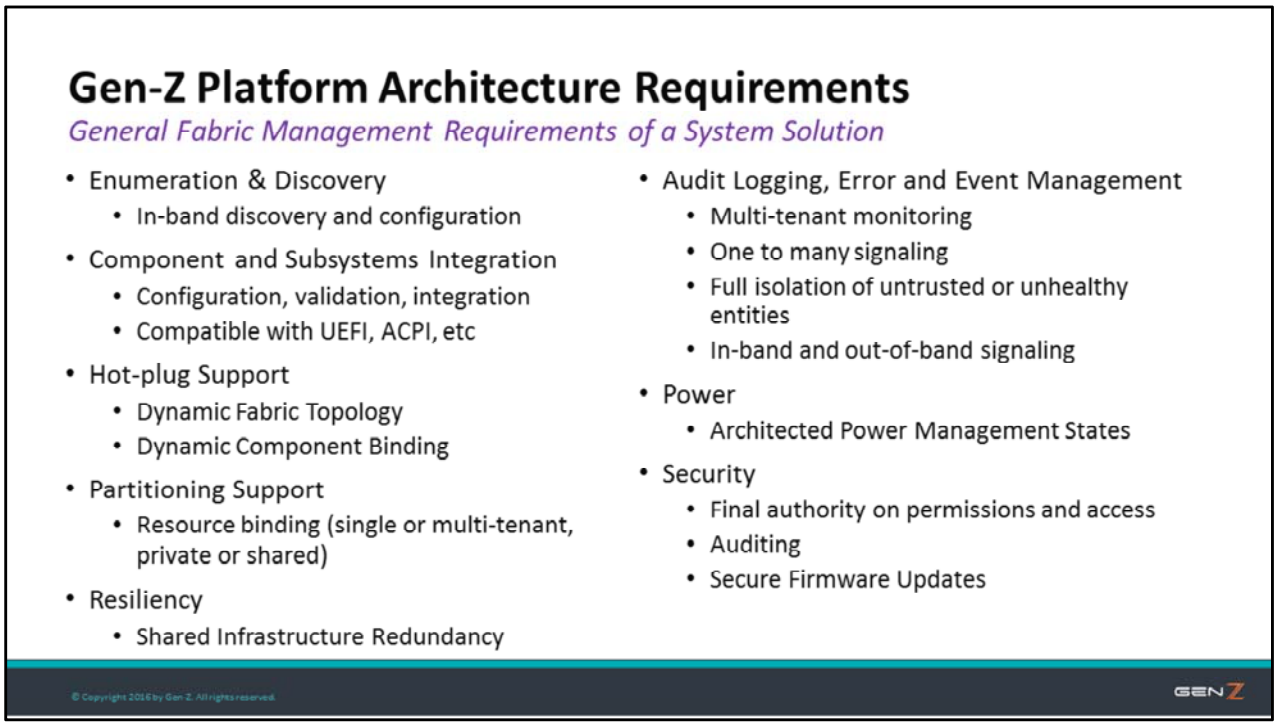
All material is subject to change at any time at the discretion of the Gen-Z Consortium

http://genzconsortium.org/

© Copyright 2016 by Gen-Z. All rights reserved.




Though various aspects of this presentation are applicable to out-of-band management and to point-to-point topologies that support in-band management, the presentation is primarily focused on switch topologies. This presentation covers single and multi-subnet switch topologies. The presentation will also touch upon multiple manager solutions.

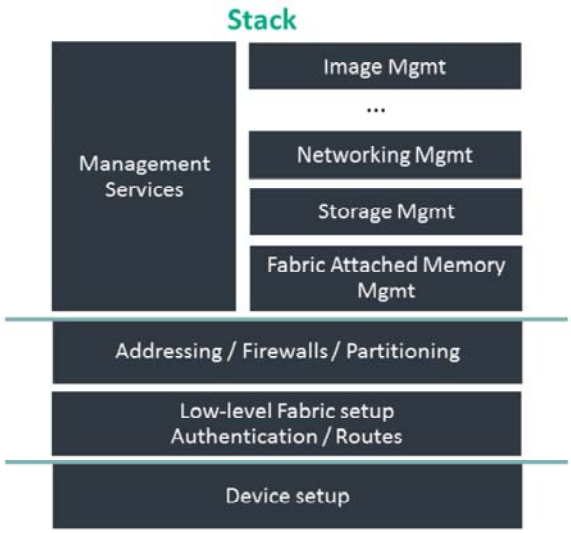



This slide enumerates numerous fabric management requirements (it is not an exhaustive list).

In general, Gen-Z uses a software defined management model to keep the hardware simple and efficient.

## Representative Gen-Z Management Stack

- **Device Setup**
  - Basic device power, clocks, resets
  - Hardware defaults
- **Low-level Fabric setup, authentication and routing**
  - Secure the Basic HW control surfaces
  - Establish local link connections, establish local state
  - Make changes to low level setup at request of higher layers
- **Addressing, Firewalls, and Partitioning**
  - Authentication of components—allowing / disallowing their participation on the fabric
  - Validating proper operation of components and unlocking fabric resources based on component IDs
  - Logical composition of memory, compute, storage and networking resources
- **High level management services**
  - Managing resources and services to meet user needs
  - Translating user intent into composition and configuration requests
  - Functions, policies and associated APIs are beyond scope of the Gen-Z Core Specification




© Copyright 2016 by Gen-Z. All rights reserved.

GEN-Z

This is a representative management stack that can be matched to the features and capabilities defined in the Core Spec.

Device set up and configuration is focused on initial component bring up (including the physical layer). Once a component is initialized, fabric managers provide additional configuration based on solution-specific needs.

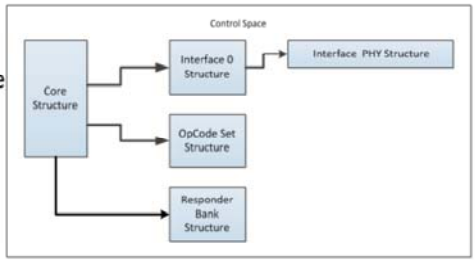
Low-level management is focused on establishing component ownership and configuration, configuration paths and relay tables, etc.

Addressing / firewalls / partitioning is focused on establishing peer component access, component and interface Access Key configuration, packet authentication and filtering configuration, etc.

Higher level management services are focused on resource provisioning, fine-grain access control, component sharing, etc.

# Gen-Z Component Control Space Overview

- Control structures are configuration and management structures that are provisioned only in Control Space.
- Control structures are self-describing, allowing a variety of structure mix and organization tailored to solution-specific needs.
- Core structure is a mandatory structure
  - Core structure is located at byte 0 in Control Space.
  - Core structure is used to locate all other structures
  - Core structure contains numerous component-wide configuration fields, e.g., component identifiers, timers, UUIDs, etc.
  - Core structure contains control and status fields
  - Core structure contains component identifiers associated with the Primary Manager or a Primary or Secondary Fabric Manager
    - These managers have complete access to Control Space

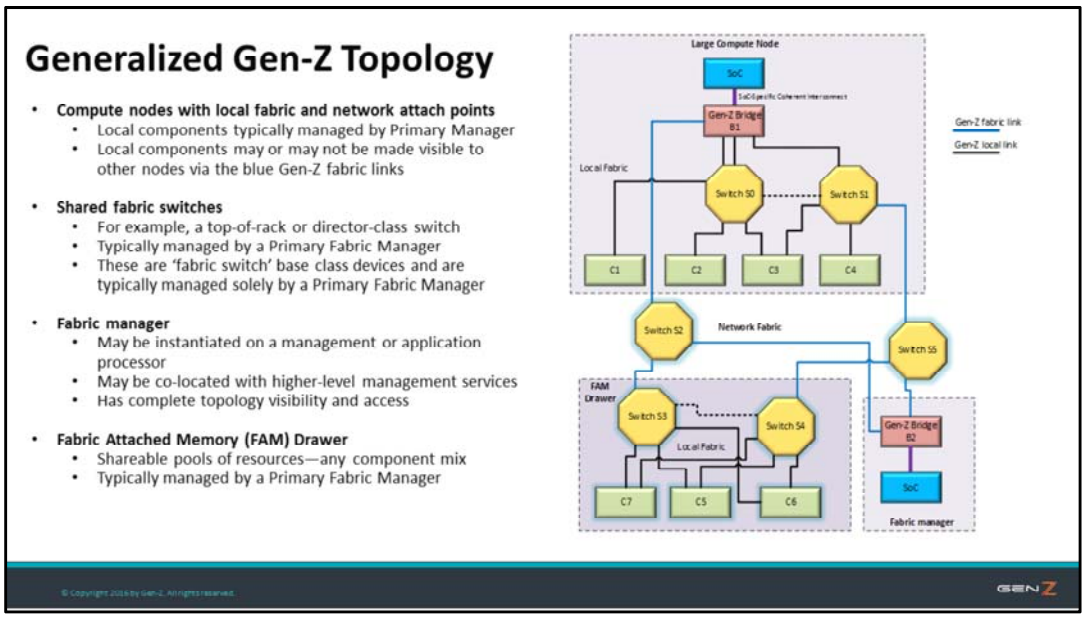



© Copyright 2016 by Gen-Z. All rights reserved.

# Gen-Z Managers

- **Primary Manager (aka 'local' manager)**
  - Primary Manager typically associated with single-enclosure solutions, e.g., a server, laptop, etc.
  - Primary Manager typically implemented using system firmware
  - Configures components using in-band Control Read and Control Write request packets
    - May also configure components through an out-of-band management interconnect
  - Component captures Source CID in first Control Write packet and designates the source component as the Primary Manager
    - Core structure PMCID identifies Primary Manager—PMCID updates transition Primary Manager to another component
- **Primary Fabric Manager (PFM)**
  - Primary Fabric Manager typically associated with multi-enclosure solutions, e.g., a server or storage rack
  - Primary Fabric Manager typically implemented as an application or agent executing on a management processor / system
  - Configures components using in-Band Control Read and Control Write request packets
  - Primary Fabric Manager may be initially viewed as the component's Primary Manager
    - Core structure PFMCID / PFMSID identifies the Primary Manager
- **Secondary Fabric Manager (SFM)**
  - Secondary Fabric Manager can act as a part of a federated management system or as a PFM back-up
  - Has same functional capabilities and uses the same operational semantics as PFM

© Copyright 2016 by Gen-Z. All rights reserved.




Standalone compute and resource enclosures may contain a Primary Manager to provide power-up initialization, access control, local topology configuration, etc.

Standalone switches do not require a Primary Manager, but use Primary and Secondary Fabric Managers to perform all configuration, exception handling, etc. These switches need to be configured in order to configure leaf compute and resource enclosures.

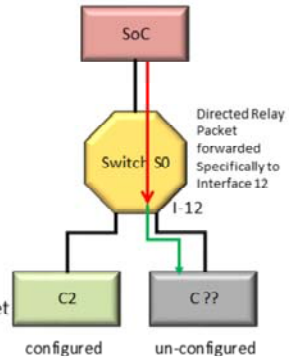
The fabric attached memory drawer, as shown, also has no local manager, but a simple one could be installed to provide self-test, initialization, and other services.

Thus, we have both local (primary) managers and at least one Fabric Manager.

Now, we need to talk about how these managers discover and obtain control of the components they are meant to control.

## Example In-band Discovery and Configuration Process

- Gen-Z uses an iterative process to discover and configure a fabric. In this example,
  - Management on the SoC detects the directly-attached switch
  - Management issues a series of Control Read and Control Write request packets
    - Assigns a CID to the switch
    - Configures all component interfaces including packet relay tables
    - Configures Unsolicited Event packet generation for a subset of events, e.g., new component
    - Enables the switch
  - Switch detects two new components and transmits an Unsolicited Event packet
  - Management uses Directed Packet Relay to communicate with the uninitialized components
    - Upon receipt, the switch uses the Directed Relay Interface Identifier to relay the request packet to the directly-attached component
    - Directed Relay enables management to communicate with and configured uninitialized components
    - Management stops using Directed Relay once it has configured the component's CID and the switch's packet relay tables
  - Management configures and enables the components




© Copyright 2016 by Gen-Z. All rights reserved.

GEN-Z

To communicate with uninitialized components within a switch topology, management sets the Directed Relay Interface Identifier field in Control Read and Control Write request packets to inform the switch which egress interface to relay the packet through to the component. This enables the switch to proxy request and response packets on behalf of the uninitialized component.

Directed Relay should be used only until management has configured the component's CID / SID and the switch's packet relay tables.

Upon receipt of the first Control Write request packet, the component captures the management component's CID / SID, and the manager now has control of the component. To uniquely identify the manager, management may configure the Core structure MGR-UUID field. Once configured, all subsequent Control Read and Control Write request packets shall contain the same MGR-UUID value, or the component will reject all new control request packets whose MGR-UUID field does not match.

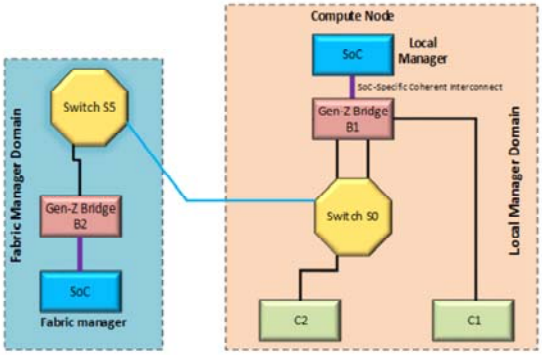
# Peer Component Detection and Attributes

- **Managers detect components in one of two ways**
  1. Uninitialized components launch 'Ready for Configuration' link packets when the link transitions to L-Up (post physical layer training). If the uninitialized component is attached to a configured component, the configured component will generate an Unsolicited Event packet to notify a manager
  2. Managers probe for uninitialized components by querying the link interfaces of their currently configured components and determining the state of the 'peer component' at the other end of each link.
- **Managers examine the Interface Structure Peer State fields connected to the uninitialized components to determine:**
  - If the peer component is already configured
  - The peer component's base class component type
  - The peer component's state of operation
  - The peer component's component ID value, and whether it is valid
  - Etc.
- **Managers use the Peer State information to determine which actions to take**
  - For example, if the peer component is uninitialized, then the manager may take control of the component
  - For example, if the peer component is configured and managed, then the manager uses the Core structure MGR-UUID to identify the current manager to determine next steps.

© Copyright 2016 by Gen Z. All rights reserved.

# Example: Merging Multiple Manager Environments

- Example depicts a standalone compute node that is dynamically attached post power-up initialization to an existing Gen-Z fabric
  - Primary Fabric Manager "owns" the existing multi-enclosure switch topology and leaf components
  - Primary manager configures the compute node
- Upon the blue link transitioning to L-Up, the two managers probe and respectively discover configured components
  - Managers need to reconcile which will take ownership of the expanded topology and components
  - Determine which compute node components are visible to the external switch topology
    - Assign new CID / SID to visible components
    - Reconfigure switch packet relay tables to enable access
  - Configure access control and permission to visible components
- See the Core specification for additional details on how the managers communicate with one another




© Copyright 2016 by Gen Z. All rights reserved.

# Tools available to Enable Merging Multiple Managers

- **Control Write Messages that include Directed Relay Interface Identifiers**
  - Create an indirect messaging channel between the managers on both ends of a link
- **MGID 0 Multicast protocols**
  - Enable local managers to broadcast messages to the adjacent fabric manager(s)
- **MGR-UUID**
  - Uniquely identifies a software instance of a distributed management entity
  - Used to validate compatibility among multiple managers
- **Software defined 'sticky bits'** in component's control structure
  - Used by current component manager to alter the manager's subsequent behaviour, e.g., to signal a Primary Manager to not take ownership and instead let a Primary Fabric Manager when coming out of component reset
- **Non-Control OpClass Packet Filtering**
  - Silently discard non-Control OpClass packets to isolate a component during to configuration from transmitting application packets, e.g., to ensure it cannot interfere with or cause harm to other components
- **Unsolicited Event Packets**
  - Enables components to communicate a wide range of events to management components.
  - Different management components can be configured to handle specific events, e.g., mechanical events to a resource manager and fabric exceptions to a Primary Fabric Manager

© Copyright 2016 by Gen Z. All rights reserved.

GEN Z

# Thank you