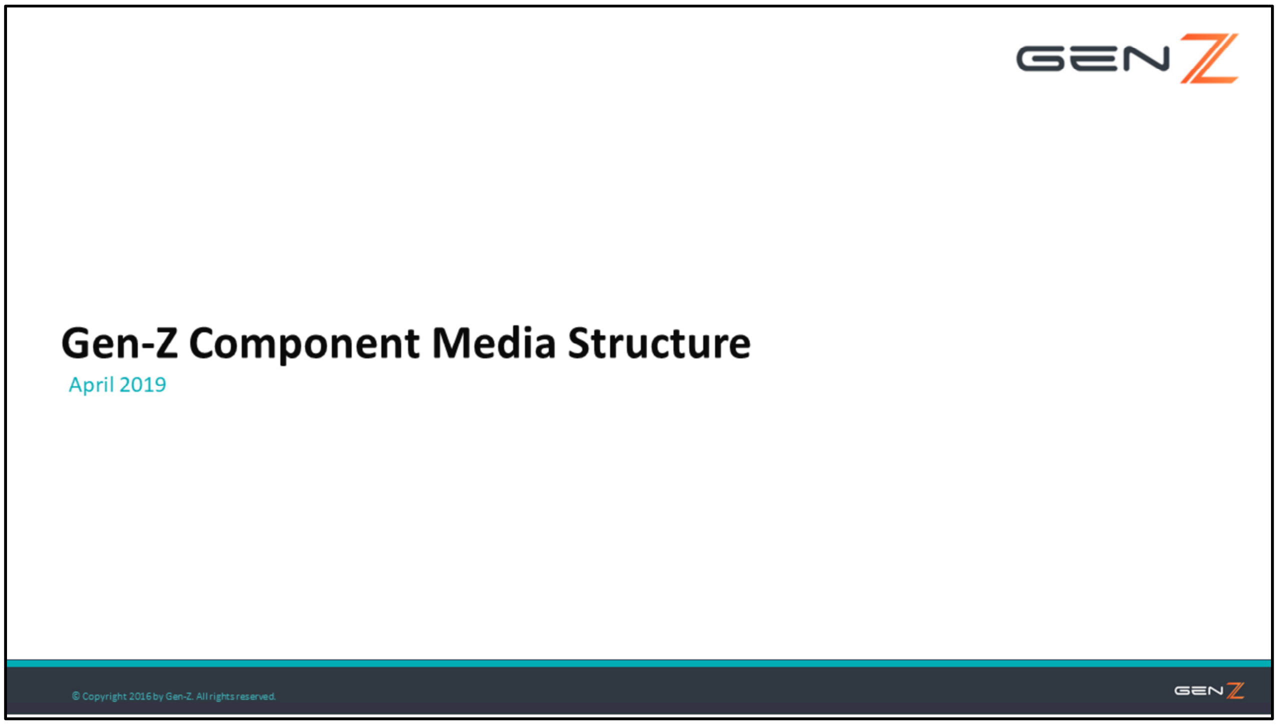
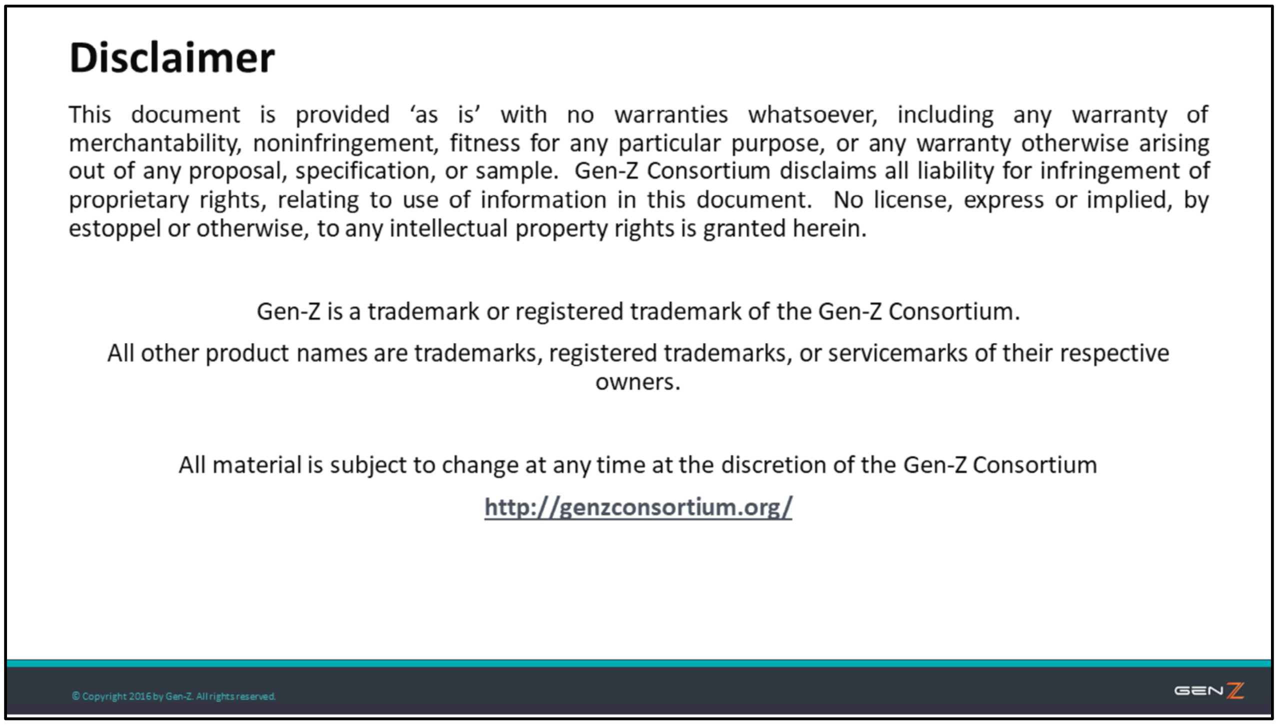


Gen-Z Component Media Structure

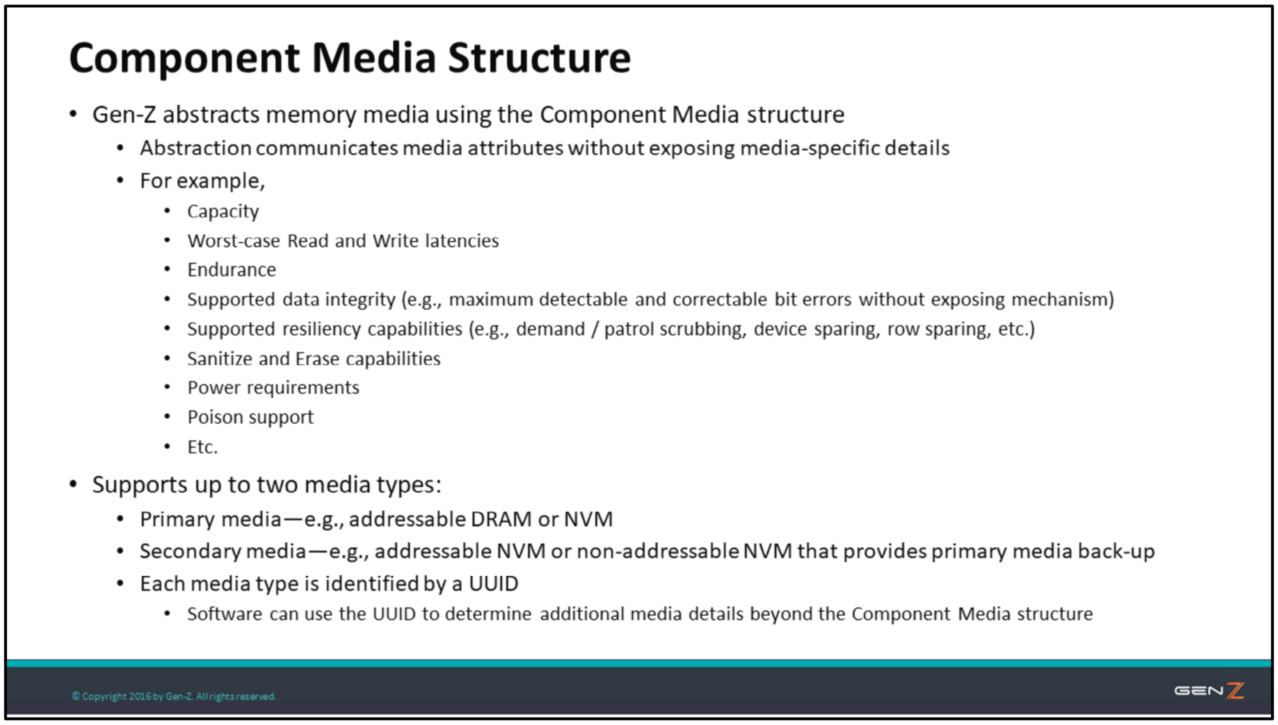
April 2019

This presentation covers the Component Media structure. The Component Media structure may be used by any component that supports addressable media.



Disclaimer

http://genzconsortium.org/



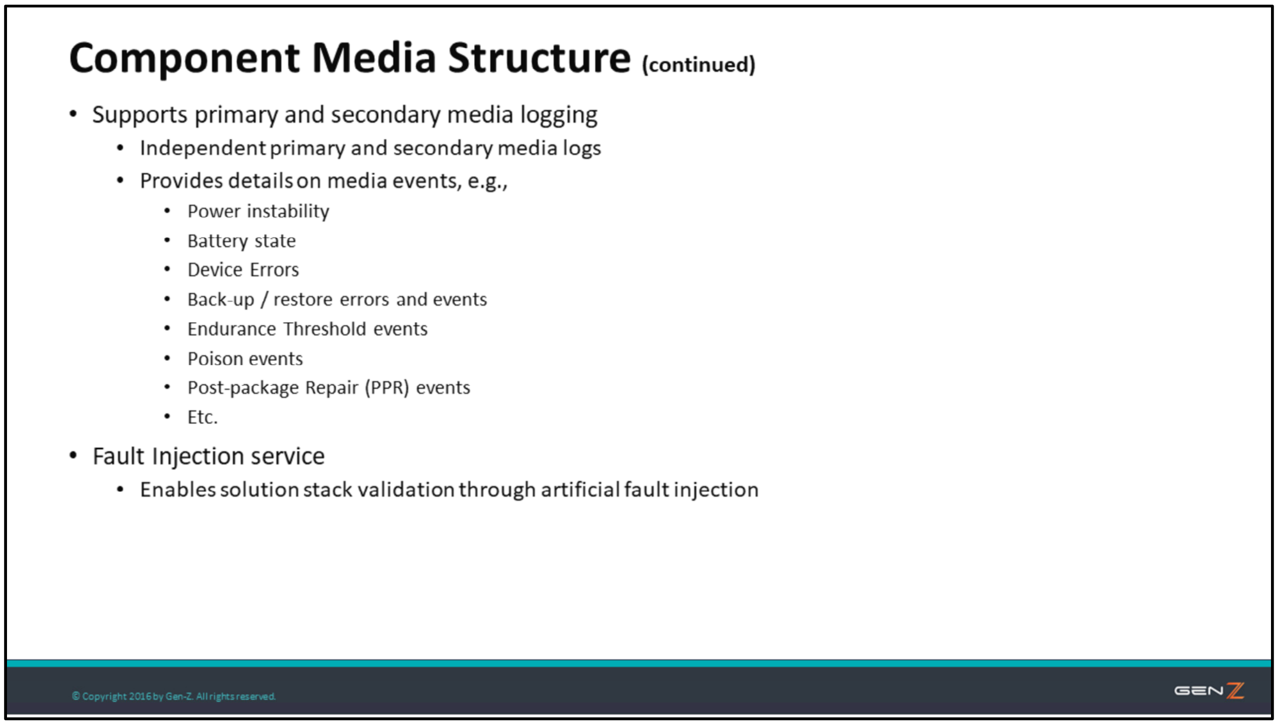
Component Media Structure

- Gen-Z abstracts memory media using the Component Media structure
  - Abstraction communicates media attributes without exposing media-specific details
  - For example,
    - Capacity
    - Worst-case Read and Write latencies
    - Endurance
    - Supported data integrity (e.g., maximum detectable and correctable bit errors without exposing mechanism)
    - Supported resiliency capabilities (e.g., demand / patrol scrubbing, device sparing, row sparing, etc.)
    - Sanitize and Erase capabilities
    - Power requirements
    - Poison support
    - Etc.
- Supports up to two media types:
  - Primary media—e.g., addressable DRAM or NVM
  - Secondary media—e.g., addressable NVM or non-addressable NVM that provides primary media back-up
  - Each media type is identified by a UUID
    - Software can use the UUID to determine additional media details beyond the Component Media structure

The Component Media structure can be used by any component that contains addressable memory media (volatile or non-volatile) or storage media. Its primary purpose is to abstract the media to enable access by any Requester without having to be media-specific aware. For example, a Requester such as a processor that supports the Component Media structure could interoperate with multiple media types—DRAM, MRAM, PCM, STTRAM, etc. using the same Gen-Z operations to access the media. Further, the abstraction enables a media controller to provide numerous services, e.g., wear leveling, resiliency (row sparing, device sparing, etc.), demand and patrol scrubbing, etc. without involving the memory controller. This can improve performance and reduce power consumption by eliminating unnecessary data movement.

The Component Media structure communicates multiple media attributes including capacity, latency, endurance, etc. It also communicates additional media services and capabilities, e.g., scrubbing, device and row sparing, sanitize and erase, etc. capabilities.

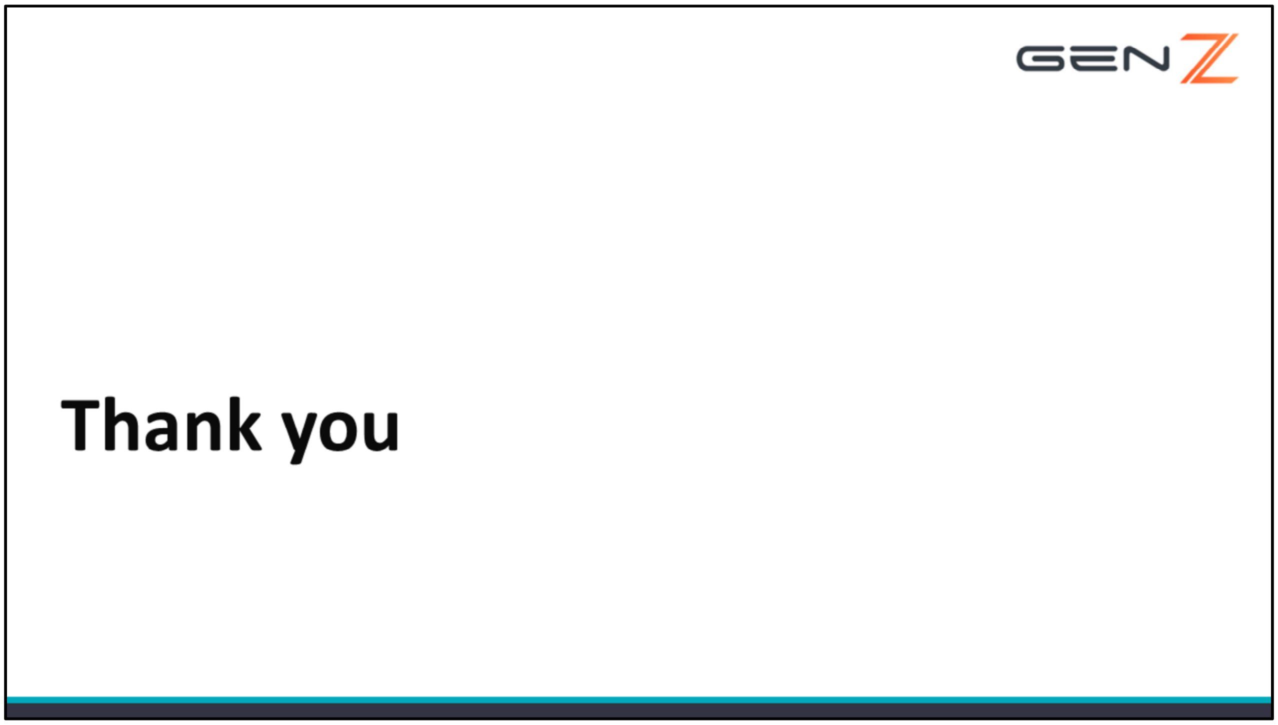
A component can support multiple media types. Each Component Media structure specifies up to two media types—a primary media and a secondary media. If a component supports more than two media types, then it provisions additional Component Media structures. Each media type is identified by a UUID. Not only does a UUID enable software to identify the media type, the UUID can identify numerous media attributes, e.g., media-specific attributes not abstracted by the Component Media structure such as a specific generation of a given media type.



Component Media Structure (continued)

- Supports primary and secondary media logging
  - Independent primary and secondary media logs
  - Provides details on media events, e.g.,
    - Power instability
    - Battery state
    - Device Errors
    - Back-up / restore errors and events
    - Endurance Threshold events
    - Poison events
    - Post-package Repair (PPR) events
    - Etc.
- Fault Injection service
  - Enables solution stack validation through artificial fault injection

In addition to media abstraction, the Component Media structure is used to manage media logs and to provide a fault injection service to ensure correct solution stack operation (the ability to test all fault conditions using standard controls is extremely important to customers not just for quality assurance, but also to help debug problems in the field).



Thank you

This concludes this presentation. Thank you.