OpenBMC: End User Features and Function

Andrew Geissler
Senior Software Engineer
IBM
Ready for Prime Time

In production by multiple companies
  – Deployed within multiple datacenters throughout the world

Passed an outrageous amount of manual and automated tests
  – Manufacturing, Functional Test, System Test

Yocto/Bitbake based so continuous kernel and security updates available
System Management

- **IPMI**
  - IPMI 2.0 Compliance
  - Full DCMI support

- **REST**
  - All D-Bus interfaces defined in `phosphor-dbus-interfaces` repo in GitHub
  - All REST api’s have a direct mapping to all defined D-Bus interfaces

- **Web Interface**
  - Chassis control, Event and Sensor views, Firmware Updates, Server configuration

- **Event Notification**
  - Create dynamic web sockets to get call backs on critical events

- **Coming Soon**
  - Redfish
Firmware Update

- Two Implementations to choose from
  - Direct write JFFS2 implementation
  - UBI Based Firmware Update
    - Dynamic partitioning
    - Wear leveling
    - Host NOR presented as directories and files

- Multiple interface options
  - TFTP, SCP, PUT
  - Multi image support (UBI only)

- Brick Protection (UBI only)
  - U-Boot
  - Side switch

- Signed Image
  - Signed during build
  - Verified before writing to the flash
Configuration & Control

- Network Configuration
  - VLAN

- Time
  - NTP
  - Host/BMC Control management (Split/Both/Host/BMC)

- Remote Host Console

- Zeroconf / Avahi

- QEMU

- Event Logs

- Power and Cooling Management
Coming Soon

- VGA Mirroring
- Redfish
- Advanced User Management and Authentication
- KVM over IP
- Remote Media
- SNMP / Telemetry
- In-band Firmware Update