An Overview of the s390x platform

Fei Li

fli@suse.com

openSUSE

Z System family (Mainframe)

- s390x architecture
- Big endian
- Each has its own machine type, like z13 has 2964, zEC12 has 2827. Each MT has several models distinguished by its PU, like z13 has N30/N96/NE1
- Hypervisor: PR/SM (manage LPAR which includes LPUs & memory & I/O devices, implement in firmware in 1980s) or z/VM (in software)
- z14 is released in 2017, but later will take z13 as an example.
- `lscpu` can learn more
- OS?

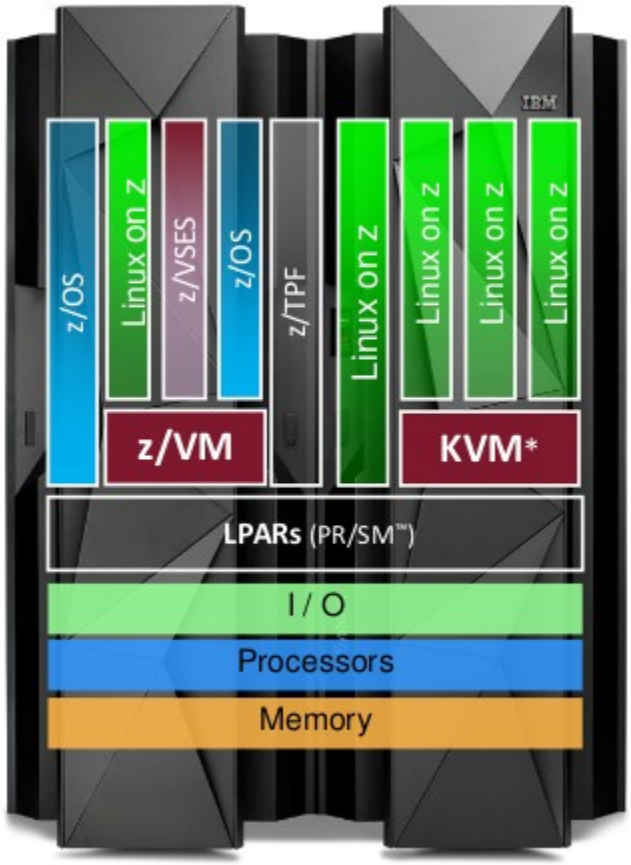
Introduce Linux on z Systems

What offerings can be installed on z Systems processors?

- z/OS
- z/VM (IBM's virt9n, 1967)
- Linux on z Systems (KVM* product)

  (2015-18: k/v/m/z => 2017-: SUSE etc)
- z/VSE
- z/TPF

FYI: PowerVM starts from 1997



Introduce Linux on z Systems (cont.)

Why z/VM => KVM?

- Simplifies configuration and operation of server virtualization
- Use common Linux skills to administer virtualization
- Embrace the Open Source virtualization community
- Easily integration into Cloud/OpenStack environments

To expand customers for z Systems.

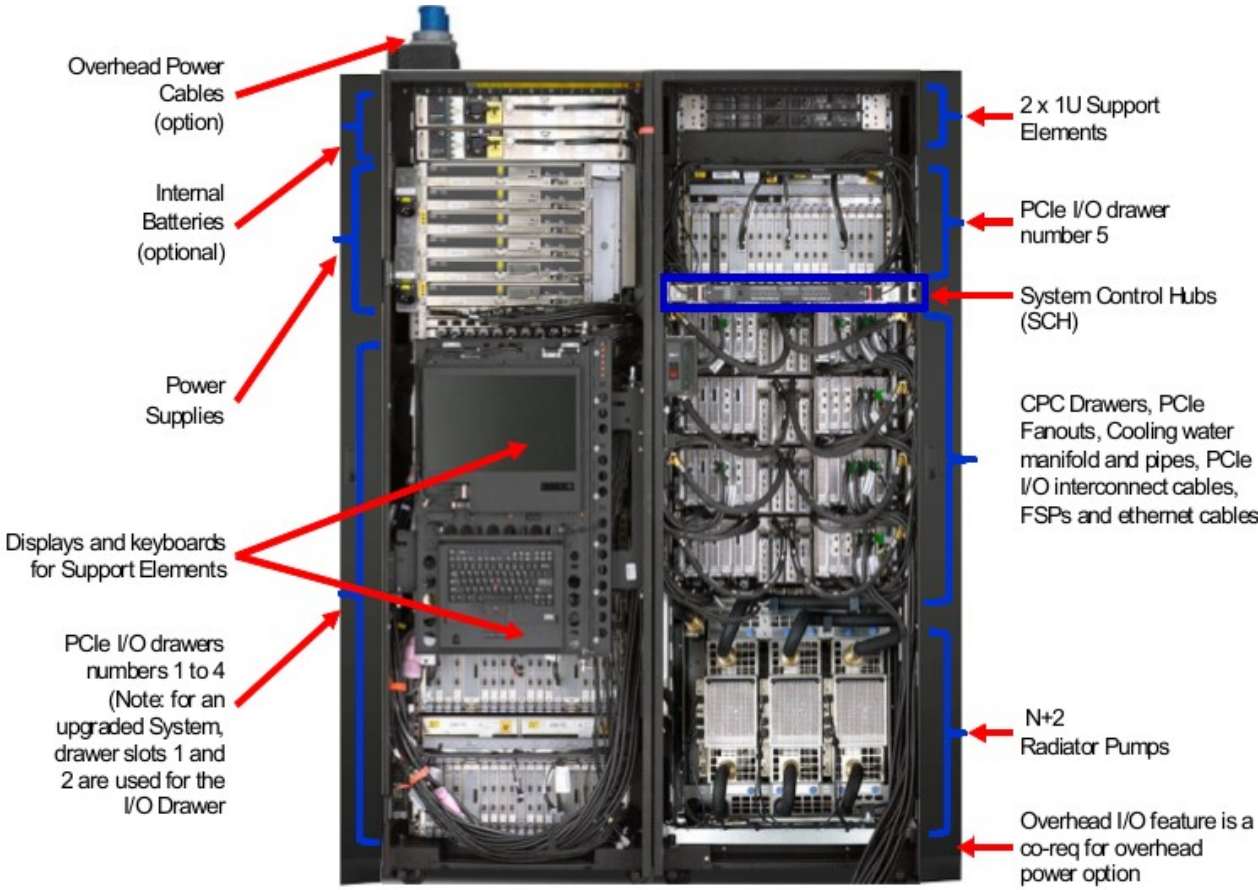
Z13 Hardware Overview

- Up to 141 characterizable PU chips
- Up to 10 TB of addressable real memory per system
- Per CPC drawer: 6 eight-core 5.0 GHz PUs with SMT, SIMD, SMP
- L1 cache: 96KB for instructions & 128KB for data

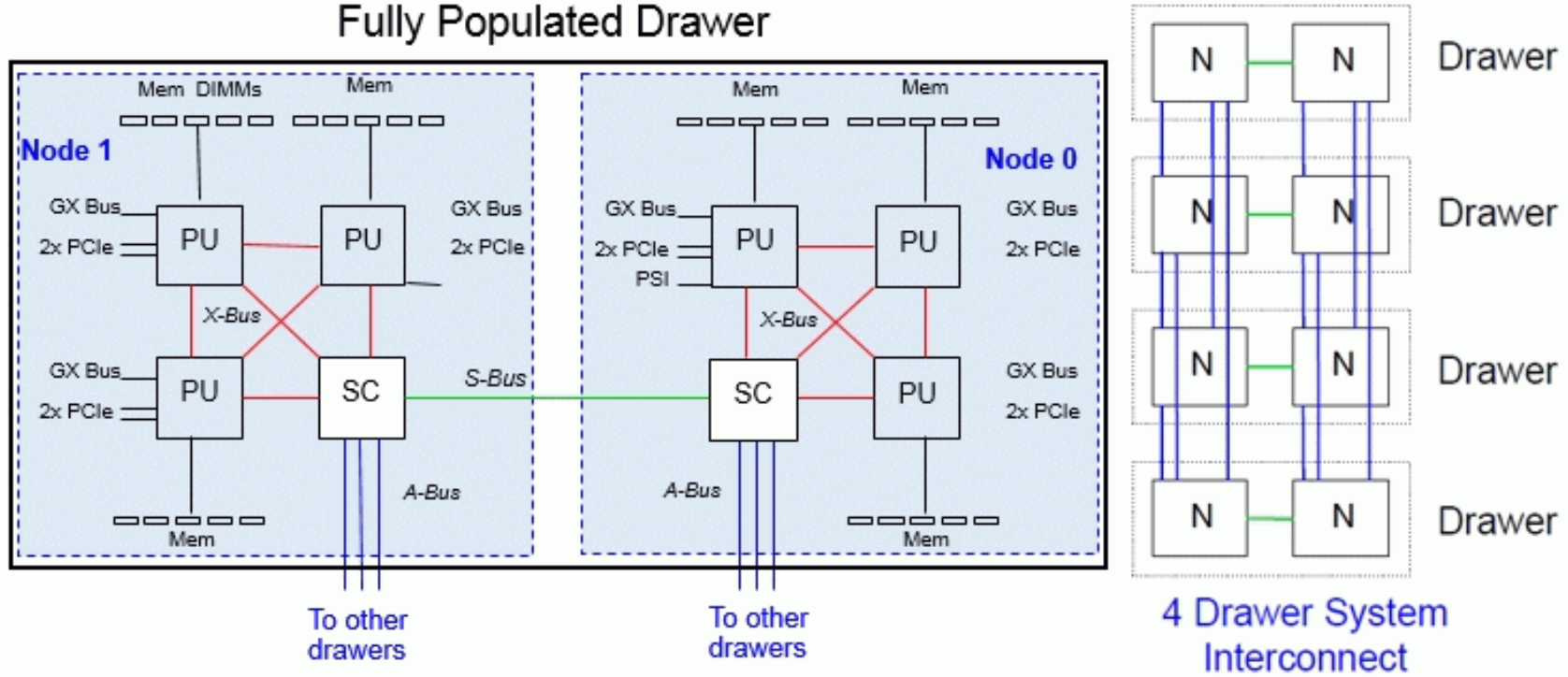
  L2: 2MB, L3 cache: 64 MB, L4: 480MB
- At most 6 LCSSs, 85 LPARs, 32K I/O devices by FICON channel
- Learn more from http://www.redbooks.ibm.com/redpieces/abstracts/sg248250.html?open

Z13 Hardware Overview (cont.)

The CPC (central processing complex)
- Two-frames: A & Z
- 5 PCIe I/O drawers, with each:
  = 32 slots + 4 switch cards
  = 4 domain * 8 features (FICON-2, OSA-2, RoCE, Crypto, Flash, zEDC)
- 4 CPC drawers, with each:
  = 6 PU SCMs (single chip modules) & 2 storage controller SCMs
  = 4 InfiniBand channel adapter & 10 PCIe Gen3 fanouts
  = [256GB, 2.5TB] memory && 20 or 25 DIMMs plugged
- CPC draw communicates via L4 shared caches
- hardware platform management: 2 integrated SE & standalone HMC
- air cooling & water cooling



Z13 Hardware Overview (cont.)



Z13 Hardware Overview (HMC && SE)

The HMC

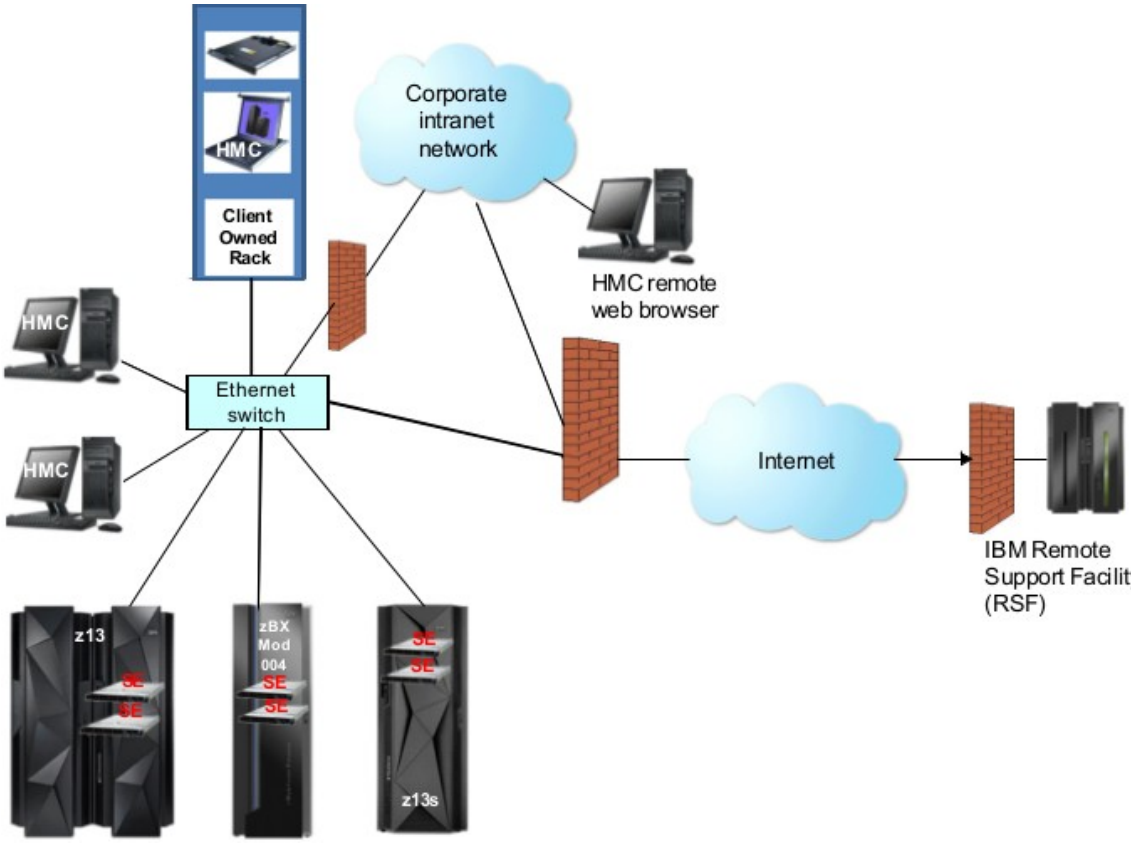
- is a stand-alone desktop computer
  or an optional rack-mounted computer

- communicates with 1 or more z Systems
  (also with POWER/x86 Systems)

The HMC && SE

- are closed systems, no other apps
  can be installed on them

- when tasks are performs on the HMC,
  the commands are sent to one or more
  SEs, which then issue commands to
  their CPCs



Channel in IBM System

- Channel I/O is a separate, simple, self-contained processor to share complex I/O tasks off CPU. Even (Initial Program Load) IPL is carried out by Channel.
- A Channel subsystem (CSS) manages the flow of data and I/O commands to an appropriate control unit which, in turn, controls I/O devices through the channel path.
- A channel program is a sequence of channel command words (CCWs) which are executed by the I/O channel subsystem.
- A channel command word (CCW) is an instruction to a specialized I/O channel processor. It is used to initiate an I/O operation, such as "read", "write" or "sense", on a channel-attached device.
- E.g. a CCW device: 0xfe.0.0001 (cssid.ssid.devno)

DASD (Direct Access Storage Device)

- set it online:

```
# chccwdev –e 0.0.7500

# lsdasd
Bus-ID    Status     Name     Device  Type  BlkSz  Size     Blocks
==============================================================
0.0.7500  active     dasde    94:0    ECKD  4096   7043MB   1803060
```

- The udev-created by-path device node for it:

```
# ls /dev/disk/by-path -l
total 0
lrwxrwxrwx 1 root root 11 Mar 11  2014 ccw-0.0.7500 -> ../../dasde
```

- set it offline

```
# chccwdev –d 0.0.7500
```

- Format the dasd

```
# dasdfmt -b 4096 /dev/disk/by-path/ccw-0.0.7500 -p
```

Virtualization on the s390x platform

- SIE
- Nested virtualization
- CPU model
- Qemu emulated main-system-bus
- PCI passthrough

Note: s390x only needs one ‘kvm’ module, not like x86 which has the ‘kvm’ module and another ‘kvm_intel/amd’ module .

Virt -> SIE (Start Interpretive Execution)

- The SIE instruction is used to run a virtual machine in emulation mode.
- vm is in the sie operation mode, like the vmx in x86
- ENTRY(sie64a) in arch/s390/kernel/entry.S
- SIE is exited either by interception or interruption. An intercept is caused by any condition that requires CP interaction such as I/O or an instruction that has to be simulated by CP.
- `lscpu | grep sie` to check if a linux instance is a hypervisor

Virt → nested virtualization

- Supported machine types: from ‘s390-ccw-virtio-2.8’
- Enable nested
  = when load kvm kernel module: `modprobe kvm nested=1`
  = append ‘kvm.nested=1’ to kernel command line
- Check if nested is enabled in the host terminal:
  `cat /sys/module/kvm/parameters/nested`, ‘Y’ is enabled.
- Check if the guest can be a hypervisor in the guest terminal:
  `cat /proc/cpuinfo | grep sie

Virt → CPU Models

- Supported since kernel 4.8/qemu 2.8

- How to use:

  = in qemu: -cpu host,+/-SomeFeature

  = in libvirt: <cpu mode='host-passthrough'/>

- Use QMP to query host's cpu model information:

  = in qemu:

    1. enable qmp in qemu command line: -qmp tcp:localhost:4444,server,nowait

    2. in another console, run `telnet localhost 4444`, and input:

       {"execute":"qmp_capabilities"}

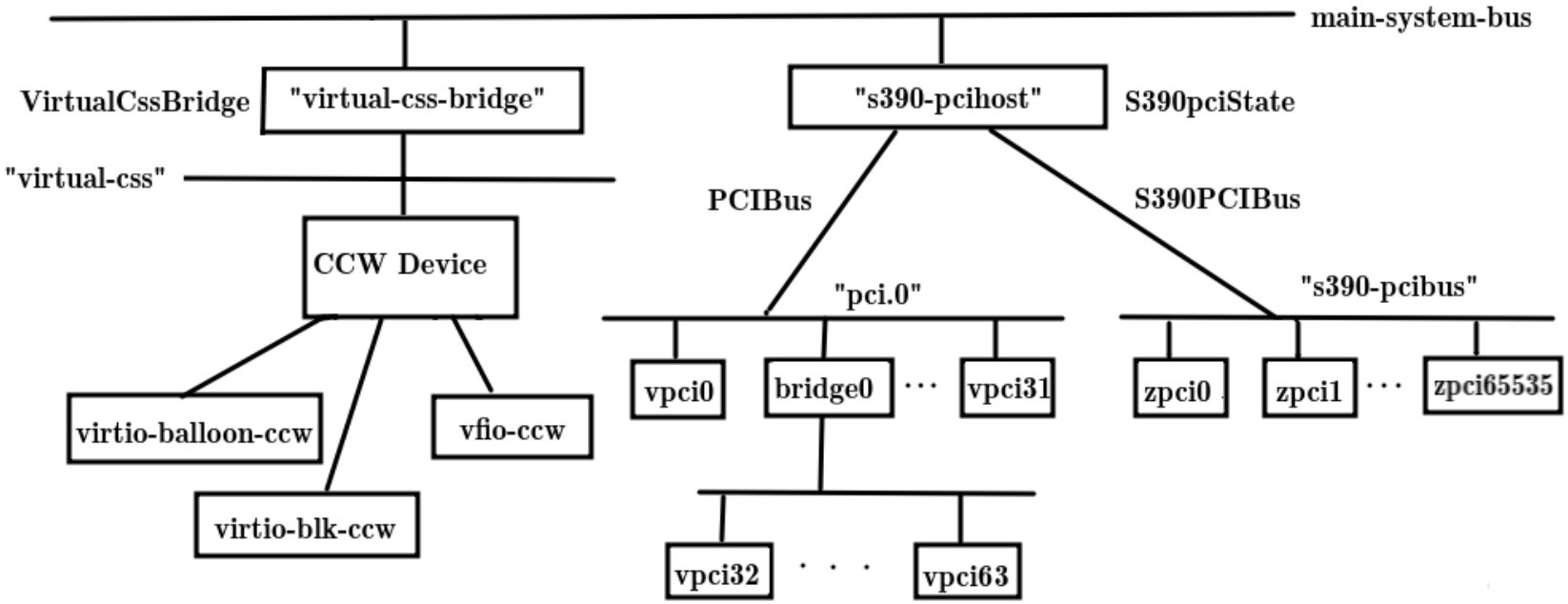
       {"execute": "query-cpu-model-expansion", "arguments":{"model" {"name":"host"},"type":"static"}}

  = in libvirt:

    virsh # qemu-monitor-command vm-name '{"execute": "qmp_capabilities"}'

    virsh # qemu-monitor-command vm-name '{"execute": "query-cpu-model-expansion", "arguments": {"model": {"name": "host"}, "type": "full"}}'

Virt -> QEMU emulated main-system-bus



Virt → PCI passthrough via vfio

- Background: PCI is supported quite recently on the z Systems

- Special:

  = An add on facility, can not be usable as boot device.

  = PCI device configuration space and memory spaces can not be accessed by memory operations, but by z specific special instructions.

  = No I/O MMU driver support, instead implement it in kernel

  = Intercept not handled in kernl, goto qemu

- Enable: `modprobe vfio-pci disable_idle_d3=1 ids="0x1111:0x2222"`

- Supported since qemu 2.7, currently libvirt does not support it

- How to use:

  = in qemu: -device zpci,uid=23,fid=45,target=vpci0,id=zpci0 \

  -device vfio-pci,host=0001:00:00.0,id=vpci0 \

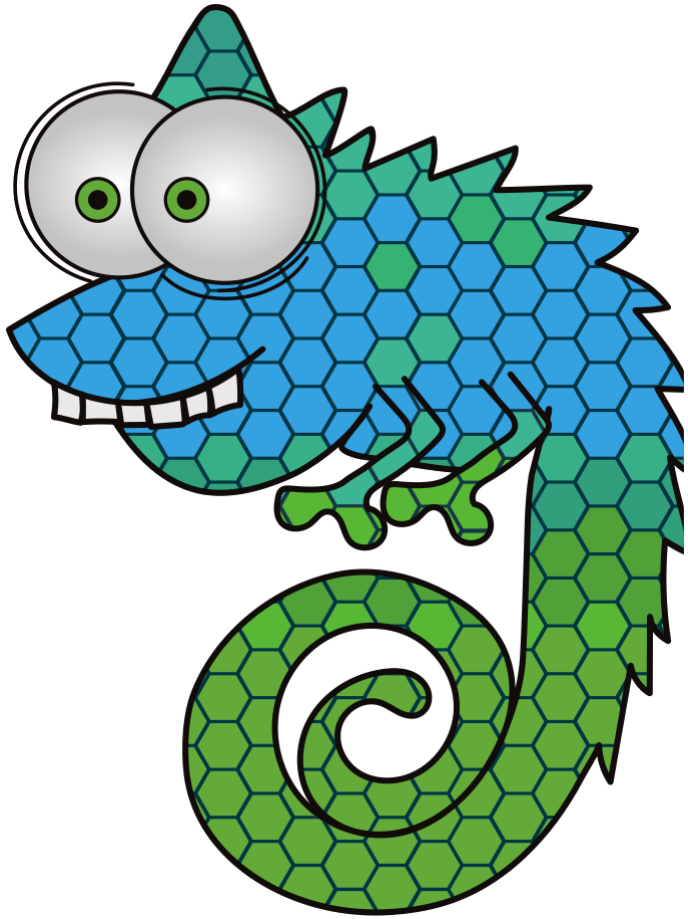
Reference

- IBM Knowledge Center
- IBM RedBooks
- kvmonz.blogspot.hk

Questions?

Join the conversation,
contribute & have a lot of fun!
www.opensuse.org

Thank you.



Have a Lot of Fun, and Join Us At:

www.opensuse.org

License

This slide deck is licensed under the Creative Commons Attribution-ShareAlike 4.0 International license. It can be shared and adapted for any purpose (even commercially) as long as Attribution is given and any derivative work is distributed under the same license.

Details can be found at https://creativecommons.org/licenses/by-sa/4.0/

General Disclaimer

This document is not to be construed as a promise by any participating organisation to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. openSUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for openSUSE products remains at the sole discretion of openSUSE. Further, openSUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All openSUSE marks referenced in this presentation are trademarks or registered trademarks of SUSE LLC, in the United States and other countries. All third-party trademarks are the property of their respective owners.

Credits

Template
Richard Brown
rbrown@opensuse.org

Design & Inspiration
openSUSE Design Team
http://opensuse.github.io/branding-guidelines/